Spider Data Collector

Benutzerhandbuch

Produktversion 1.2202

Ausgabe: 24-02-2022

Dokumententitel	Spider Data Collector - Benutzerhandbuch
Produktversion	1.2202
Herstellung und Druck	Flexera Software LLC 300 Park Boulevard Itasca IL 60143, USA www.flexera.com
Veröffentlichungsdatum	25.02.2022

Inhaltsverzeichnis

1 General 1

1.1 Typografische Konventionen ... 1
1.2 Hilfestellungen ... 1
1.3 Abkürzungen ... 1

2 Neuigkeiten 2

3 Installation und Einrichtung des Data Collectors 5

3.1 Systemvoraussetzungen ... 5
3.1.1 Software ... 5
3.1.2 Hardware ... 6
3.2 Installation ... 6
3.3 Konfiguration ... 17
3.3.1 Data Collector ... 17
3.3.2 SFTP Server ... 19
3.4 Zurücksetzen des letzten Verarbeitungsdatums ... 21
3.5 Hersteller zu Vertrauenswürdige Hersteller hinzufügen ... 22
3.6 Deinstallation ... 28
3.7 Behebung von Verbindungs-/Authentifizierungsproblemen ... 30
3.8 Dienstkonto des Data Collectors ändern ... 32

4 Konfiguration der Konnektoren 34

4.1 Passwortverschlüsselung bei Konnektoren ... 37
4.2 Datenbankbasierende Konnektoren ... 37
4.2.1 Discovery Systems Data Connector (DSDC.exe) ... 37
4.2.2 Ausgabe von MAC und IP Informationen unterdrücken (DSDC.exe.config) ... 38
4.2.3 Microsoft Endpoint Configuration Manager (MECM) ehemals System Center Configuration Manager (SCCM) ... 39
Metering ... 40
SQL Server Editionserkennung ... 40
4.2.4 Spider Data Center Inventory ... 42
4.2.5 Heat Discovery ... 42
4.2.6 Frontrange Discovery ... 43
4.2.7 Landesk ... 44
4.2.8 Lansweeper ... 44
4.2.9 Altiris 7 ... 44
4.2.10 Generischer Konnektor ... 45
4.2.11 Microsoft Assessment and Planning Toolkit (MAP) ... 46
4.2.12 Matrix42 (Beta) ... 48
4.2.13 Empirum Workplace Management (Beta) ... 48
4.2.14 Baramundi (Beta) ... 49
4.2.15 Snow (Beta) ... 49
4.2.16 Übersicht der ermittelten Daten ... 50
4.3 API basierende Konnektoren ... 55
4.3.1 Einführung ... 55
Verwendung eines Proxy Servers ... 55
Fehlersuche bei der PowerShell Konnektor Ausführung ... 56
PSRemoting - Befehle auf entfernten Computern ausführen ... 56
PowerShell Execution Policy ... 57
4.3.2 VMware vCenter / ESX Server ... 58
Systemvoraussetzungen ... 58
Einstellungen ... 58

Konnektor für Datacenter-Modul ... 61
4.3.3 Adobe Online ... 63
4.3.4 Microsoft Azure (Microsoft Online) ... 67
Microsoft AzureAD - Nutzerbasierende Lizenzinformationen ... 68
Microsoft AzureAD - Benutzer Export ... 69
Microsoft AzureAD - Gruppen Export ... 69
Microsoft AzureAD - Einrichtung API Zugriff über eine Applikation ... 70
4.3.5 Microsoft Intune ... 76
Microsoft Intune - Einrichten der Applikation im Azure Portal ... 76
4.3.6 Microsoft Active Directory ... 80
Benutzerobjekte ... 80
Computerobjekte ... 82
Gruppen-Objekte ... 83
4.3.7 Microsoft Application Virtualization (App-V) Konnektor ... 85
App-V Paketdaten, PowerShell basierend ... 86
App-V Paketdaten, SQL basierend ... 86
App-V Nutzungsdaten, SQL basierend ... 87
4.3.8 Hyper-V ... 88
4.3.9 Hyper-V via Virtual Machine Manager ... 90
4.3.10 Microsoft Exchange Connector (Beta) ... 91
4.3.11 LDAP (Beta) ... 92
4.3.12 XEN Server (Beta) ... 93

5 Spider/Columbus Inventory (Windows / Mac OS) 94
5.1 Windows ... 94
5.1.1 Systemvoraussetzungen Columbus Inventory ... 94
5.1.2 DSGVO / GDPR Verhalten ... 94
5.1.3 Columbus Inventory Agent ... 95
Columbus Inventory Agent Quelldateien ... 95
Columbus Inventory Agent Konfiguration ... 95
Columbus Inventory Agent Installation ... 97
Columbus Inventory Agent Aktualisierung ... 97
Columbus Inventory Agent Scanzeitpunkt zurücksetzen ... 98
Columbus Inventory Agent Metering ... 98
5.1.4 Columbus Inventory Agent MSI ... 98
Columbus Inventory Agent MSI Installation ... 98
Columbus Inventory Agent MSI Quelldatei ... 100
Verteilung per GPO (Schritt für Schritt) ... 100
5.1.5 Columbus Inventory Scanner ... 106
Columbus Inventory Scanner Quelldateien ... 106
Columbus Inventory Scanner Konfiguration ... 107
Columbus Inventory Scanner Ausführung ... 108
5.1.6 Columbus Inventory Scanner Scanzeitpunkt zurücksetzen ... 110
5.1.7 Erhobene Hardware Informationen ... 110
5.1.8 Unterschiede Scanner / Agent ... 112
5.1.9 Erweiterte Inventarisierung mit den Scanner Add-on DLLs ... 112
5.1.10 Zusätzliche Werte aus der Registry erfassen ... 112
5.1.11 SSL verschlüsselte Übertragung ... 113
5.2 Mac OS ... 114
5.2.1 Columbus Inventory Scanner Quelldateien ... 114
5.2.2 Columbus Inventory Scanner Konfiguration ... 114
5.2.3 Columbus Inventory Scanner Installation ... 116
5.2.4 Columbus Inventory Scanner Ausführung ... 117

6 Data Center Inventory (Linux / Unix) 118
6.1 Voraussetzungen ... 118
6.1.1 Begriffsdefinition ... 118
6.1.2 Netzwerk-Ports ... 119

6.1.3 Serversysteme ... 119
6.1.4 UUID-Generator ... 120
6.2 Oracle Datenbanken ... 120
6.2.1 Ausführung der Grant-Skripte ... 120
6.2.2 Hinterlegung der Anmeldedaten ... 121
6.3 Installation der Agenten ... 122
6.3.1 Linux ... 122
RPM Pakete ... 122
DEB Pakete ... 122
6.3.2 Solaris, HPUX, AIX ... 122
6.3.3 Mac OS ... 123
6.3.4 Windows ... 123
6.4 VMware vCenter ... 123
6.5 Agenten in der Spider Data Center Appliance einrichten ... 124
6.5.1 Anlegen mit dem Editor ... 124
6.5.2 Import von größeren Mengen von Systemen ... 126
6.6 Deinstallation der Agenten ... 129
6.6.1 Deinstallation auf Linux, HPUX, AIX, MacOS ... 129
6.6.2 Deinstallation der RPM Pakete ... 130
6.6.3 Deinstallation der DEB Pakete ... 130
6.6.4 Deinstallation auf Windows ... 130

7 Erfüllung der DSGVO/GDPR Anforderungen in Spider Produkten 131
7.1 Konnektoren mit personenbezogenen Daten ... 131
7.1.1 API basierende Konnektoren ... 131
7.1.2 Datenbankbasierende Konnektoren ... 136
7.2 Inventory Komponenten ... 136
7.2.1 Windows ... 136
7.3 Ablageorte von personenbezogenen Daten ... 137
7.4 Sicherer Datentransport ... 137

8 FAQ 137
8.1 TCP/IP Socket basierende Kommunikation (OTB) ... 138
8.2 Ablageorte der Logdateien ... 138
8.3 Datenfluss ... 139

9 Anhang 139
9.1 PowerShell Modul - bwgTools ... 139
9.2 Stored Procedures des generischen Konnektors ... 140
9.2.1 dbo.swrGetWorkList ... 141
9.2.2 dbo.swrGetHardwareScan ... 141
9.2.3 dbo.swrGetFileScan ... 144
9.2.4 dbo.swrGetSoftwareScan ... 144
SQL Server Editionserkennung ... 146
9.2.5 dbo.swrGetDeviceRelationship ... 148
9.2.6 dbo.swrGetADUserObject ... 148
9.2.7 dbo.swrGetADGroupObject ... 149
9.2.8 dbo.swrGetADGroupMember ... 150
9.2.9 dbo.swrGetSwidScan ... 150
9.3 Inventarisierung mittels MAP Toolkit ... 151
9.3.1 Datenbank ... 151
9.3.2 Installation ... 153
9.3.3 Konfiguration ... 154
9.3.4 Inventardaten erheben ... 156
VMware Daten ... 160

General

In diesem Kapitel

Typografische Konventionen 1
Hilfestellungen..... 1
Abkürzungen 1

1.1 Typografische Konventionen

In diesem Handbuch werden verschiedene Formatierungen verwendet, um bestimmte Begriffe und Aktionen hervorzuheben. Spezielle Hinweise und Tipps werden je nach Gewichtung mit einer anderen Hintergrundfarbe dargestellt.

Formatierung	Beschreibung
Fette Schrift	Elemente in der Software oder im Betriebssystem, wie Menüpunkte, Buttons oder Elemente einer Auswahlliste
Kursivschrift	Hervorhebungen (wichtige Details) und Verweise auf andere Kapitel oder Dokumente
Dreieck Symbol "➢"	Schritt einer Handlungsanweisung
Spitze Klammer ">"	Befehlsmenüabläufe, z.B. **Datei** > **Öffnen**
`Systemschrift`	Verzeichnisse, Code- und Scriptbeispiele
GROSSBUCHSTABEN	Tastenbezeichnungen, z.B. SHIFT, STRG, oder ALT
TASTE+TASTE	Tastenkombinationen, bei welchen der Benutzer eine Taste gedrückt halten muss und eine weitere Taste drückt, z.B. STRG+P oder ALT+F4.

Hinweis	Wird für Hinweise oder Tipps verwendet, welche die Arbeit erleichtern oder für zusätzliche Informationen, die das Verständnis für das Produkt fördern.

Wichtig	Informationen, die der Benutzer beachten sollte, da sonst Probleme oder Mehraufwand im Betrieb entstehen können.

Achtung	Informationen, die der Benutzer beachten muss, um Fehlfunktionen des Systems (Abstürze, Datenverluste, Systemausfall) zu verhindern.

1.2 Hilfestellungen

Für zusätzliche Informationen und Unterstützung empfehlen wir die Flexera Community (https://community.flexera.com/). Hier finden Sie die Produktdokumentation, Download-Links und Zugang zum Support.

1.3 Abkürzungen

Zum besseren Verständnis sind hier die Details zu den im Dokument verwendeten Abkürzungen aufgeführt:

DC	Data Collector
DR	Data Receiver
OSE	Operating System Environment
WMI	Windows Management Instrumentation
PS	PowerShell

Neuigkeiten

1.2202.1

- Die Passwortverschlüsselung bei PowerShell Konnektoren ist nun geräteabhängig. Die erstellten Dateien können daher nur noch auf dem System verwendet werden auf dem sie generiert wurden.

1.2201.2

- Neue Version der Datacenter-Inventory-Komponenten (Version 12.4).

1.2111.1

- Der vSphere Konnektor erfasst nun die aktuelle Version der PowerShell und PowerCLI Komponenten.
- Das neue diagnostic tool kann nun die verwendete version des vCenters ermitteln.

1.2108.1

- Die Kommunikation zwischen dem Spider Data Collector und dem Recognition Modul wurde aktualisiert. Es wurden verbesserte Sicherheitsmechanismen implementiert. Es wird nun openSSL 1.1 und eine neue Autentifizierungsmethode verwendet. Dies bedeutet auch, dass alle Spider Data Collectoren auktualisiert werden müssen, damit sie weiterhin Daten an das Recognition Modul senden können.
- Es wird eine neue Version des Columbus/Spider Inventory ausgeliefert (Version: 7.6.5.21214). Der Inventory Agent kann, so konfiguriert, sich selbst aktualisieren, der Inventory Scanner muss manuell aktualisiert werden.
- Der Adobe Portal Konnektor wurde erweitert und unterstützt nun TLS. Um TLS zu verwenden muss der Aufruf des Konnektors um den Parameter "TLS = true" ergänzt werden.
- Fix: Es kommt nicht mehr zu Problemen, wenn der Benutzer-Import aus dem Adobe Portal für mehrere Benutzer die selbe EMail-Adresse enthält.

1.2107.1

- Folgender Fehler im vCenter Konnektor wurde behoben: Wurden Clusterdaten vom vCenter Konnektor und der Datacenter Appliance geliefert, wurde diese nicht als identisch erkannt, was zu Redundanzen im Recognition Modul führte und damit auch zu doppelten Assets in Spider Asset.

1.2104.1

- Ein Fehler im AD Konnektor, der im Zusammenhang mit GetADComputer auftreten konnte, wurde behoben.

1.2103.1

- Der vCenter Konnektor wurde weiter verbessert. Er wird automatisch mit dem Update des Spider Data Collectors (SDC) aktualisiert.
  - Es werden nur noch "Powered On" Guest für die Host-Guest-Beziehungen berücksichtigt um ungünstige Konstellationen zu vermeiden.

1.2102.1

- Ein Problem mit dem HEAT Konnektor konnte behoben werden
- Der vCenter Konnektor wurde weiter verbessert. Er wird automatisch mit dem Update des Spider Data Collectors (SDC) aktualisiert.
  - Der neue Parameter "OnlyWindows" bewirkt, dass nur virtuelle Instanzen mit einem Windows Betriebssystem detailliert erfasst werden. Andere virtuelle Instanzen (z. B. mit Linux Betriebbssystemen) werden nicht derartig erfasst. Die zusätzlichen Informationen erlauben es in Spider Assets zu diesen erfassten Instanzen zu erstellen.
  - Der vCenter Konnektor liefert, aufgrund eines erkannten Problems, nun keine IP-Adressen mehr.
  - In einer VDI-Infrasturktur liefert der vCenter Konnektor für Windowssysteme (Windows 7, Windows 8, or Windows 10) nur noch den Hostnamen für die Geräteidentifikation.

1.2012.1

- Der SCCM Konnektor wurde um die folgenden Punkte erweitert:
  - SQL Server 2019 Versionserkennung
  - Windows 10 build Versionserkennung
  - Dateiexport für Visual Studio
- Der VCenter-Konnektor wurde weiter verbessert. Nur verbundene Hosts werden bearbeitet, nicht verbundene Hosts werden ignoriert.
- Es konnte vorkommen, dass Guests, für die kein Hostname geliefert wurde, einen Fehler verursacht haben. Das wurde behoben.

1.2011.1

- Für den Columbus Inventory Agent und den Columbus Inventory Scanner wurden neue Versionen veröffentlicht. Zusätzlich zur Behebung eines sicherheitsrelevanten Problems, wurde die Leistungsfähigkeit des Inventory Scanners verbessert.
- Fix: Unter bestimmten Umständen lieferte der vCenter Konnektor keine Daten. Das zugrunde liegende Problem bei der Erstellung der SWRD-Datei wurde behoben.
- Seit der Septemberversion des vCenter Konnektors wurden Daten von laufenden virtuellen Systemen erfasst und an Spider geliefert. Das führte dazu, dass Assets erstellt wurden auch wenn für sie keine weiteren Inventardaten vorlagen. Dieses Verhalten kann nun abgeschaltet werden.

1.2009.1

- Der vCenter interface Konnektor wurde um weitere Guest-Informationen erweitert. Es werden nun Hostname, Domain, Betriebssystem, CPU Informationen und weitere Felder übertragen. Das führt dazu, dass Geräte, die von keinem anderen Konnektor geliefert werden, nun über diesen Weg in Spider angelegt werden.

1.2006.1

- Es wurden zusätzliche Filescan-Filter an den datenbankbasierten Konnektoren implementiert. Dateien, die nicht mehr benötigt werden, werden nicht mehr exportiert. Da die Änderungen bereits beim Erstellen der, nun kleineren, Exportdatei greifen, wirkt sich das am Import in die Datenbank in gesteigerter Leistung aus.
- Die datenbankgestützten Konnektoren unterstützen nun auch verschlüsselte Passwörter, wie bislang nur die API-basierten Konnektoren
- Der Hyper-V- Konnektor wurde erweitert. In einer Konfigurationsdatei können Hostnamen von weiteren windowsbasierten Hyper-V Hosts angegeben werden. Diese Geräte werden der Reihenfolge nach inventarisiert ohne den Konnektor erneut ausführen zu müssen.

1.2005.1

- SCCM-Konnektor: Aufgrund von Performance-Problemen wurde der mit dem März-Update ausgelieferte, zusätzliche Filter für Dateien entfernt.

- Die Lieferung für das Data Center Inventory für Linux und Unix wurde in den Multi-Plattform-Inventaragenten geändert. Diese übertragen die Inventarergebnisse direkt an die Spider Data Center Appliance. Der vorherige "cis"-Agent wurde aus der Lieferung entfernt.

1.2004.1

- Lieferungen des Adobe Online Conectors werten die "Single App" Einträge aus. Die Profilnamen werden verarbeitet und in Spider angezeigt

1.2003.1

- Der SCCM-Connector wurde verbessert, um die Größe der Ausgabedatei zu verringern, indem irrelevante Dateiscans vermieden werden.

1.2001.2

- Citrix Remote Desktop Services werden nun auch bei doppelt verschachtelten Zugriffen erkannt.

1.2001.1

- Fehler bei FileScan_Columbus bei großen Datenlieferungen behoben
- Möglicher Überlauf Fehler (Arithmetic overflow error) des Ivanti Connectors behoben

1.1912.1

- Priorisierung anliefernder Connectoren möglich

1.1911.1

- Exchange Connector wird nun mit ausgeliefert
- Fehler in Erkennungsregeln bzgl. Adobeprodukten behoben.
- Anpassen und Erweiterung der Anbindung von eRunbook
- Verbesserungen am Intune Connector

1.1910.1

- Verbesserung am Intune Connector: durchgehende Umstellung auf SerialNo-basierte URN
- Bessere Unterstützung zur Anonymisierung von Datenexporten

1.1909

- Ältere Versionen des Adobe Online Connector wurden durch die neue Connectorversion ersetzt.
- Änderung des Microsoft Intune Connector, die Deviceerkennung basiert nun auf dem Feld SerialNo.

1.1908

- Microsoft Intune Connector wurde verbessert und durch Kundenfeedback konnte beta-Kennzeichnung entfallen.

1.1907

- Für Altiris Connector konnte durch Kundenfeedback die beta-Kennzeichnung entfallen

1.1906

- Der Spider Data Center Inventory Connector bleibt bestehen

1.1905

- Performance des Altiris Connectors verbessert

1.1812

- Sektion zur Konfiguration des SFTP Server hinzugefügt.

1.1811

- Altiris Connector (Beta) hinzugefügt

1.1809

- XEN Connector (Beta) hinzugefügt

1.1806

- ESX/vCenter Konnektor für das Datacenter-Modul
- Linux und Unix zusätzlich zu MAC Inventory

1.1805

- Informationen zu DSGVO ergänzt
- Microsoft Exchange Connector (Beta)

1.1804

- Columbus Inventory Scanner für Mac OS
- Restrukturierung des Dokuments

1.1803

- Das Spider Data Collector User Manual ist jetzt auch auf Deutsch verfügbar
- Azure Verbindung mittels Applikation/Zertifikat (anstatt Benutzer/Password)
- Empirum Workplace Management (Beta)

1.1802

- Microsoft App-V Connector (Beta)
- Baramundi Connector (Beta)
- Export der Meteringdaten mit dem SCCM Konnektor

Installation und Einrichtung des Data Collectors

3.1 Systemvoraussetzungen

3.1.1 Software

- Betriebssystem: Windows 2008 Server oder höher
- Microsoft .NET 4
- Microsoft PowerShell 3.0 oder höher
- Darf nicht parallel auf einer Maschine installiert werden auf der Software Recognition (RC) installiert ist.
- Mehrfache Installationen auf der gleichen Maschine sind nicht unterstützt.

Achtung	Beginnend mit dem Release 1610 sind alle PowerShell (ps1) Skripte digital signiert. Sollte die PowerShell Execution Policy auf "AllSigned" eingestellt sein, ist es notwendig den Hersteller des Signaturzertifikates dem Speicher für Vertrauenswürdige Hersteller auf der lokalen Maschine hinzuzufügen. Das Setup bietet an den Hersteller dem Speicher für Vertrauenswürdige Hersteller hinzuzufügen, diese Aktion fügt den Hersteller aber nur dem Speicher für den aktuellen Benutzer hinzu. Um das Zertifikat dem Maschinenspeicher für Vertrauenswürdige Hersteller hinzuzufügen damit alle Konten der Maschine darauf zugreifen können, befolgen Sie bitte die Anleitung im Kapitel Hersteller zu Vertrauenswürdige Hersteller hinzufügen (siehe Seite 22).

3.1.2 Hardware

Es wird empfohlen eine physische oder virtuelle Maschine mit den folgenden Werten zu verwenden:

Prozessor: 2 oder mehr aktuelle CPUs/Cores

RAM: 4-8 GB

Festplatte

Zusätzlich zu dem durch das Betriebssystem verbrauchten Platz wird benötigt:

Abhängig von der Anzahl der Maschinen die exportiert werden, gelten die folgenden (groben) Richtwerte:

System	Empfehlung
Inventory Agent/Scanner	Zip Dateien haben in der Regel eine Größe von 10kb bis 800kb pro Maschine, nach der Übermittlung werden die Zip Dateien gelöscht.
SCCM	2000 Maschinen, 25MB 560 Maschinen, 3.5MB 32900 Maschinen, 450MB
MAP	1280 Maschinen, 1.2MB

Notiz	Zu beachten ist das die Dateigröße von der Menge der abgefragten Daten abhängt, speziell die Abfrage von Dateiinformationen kann eine größere Menge an Speicherplatz benötigen.

3.2 Installation

Herunterladen der "Spider Data Collector.exe" und Starten der Installation.

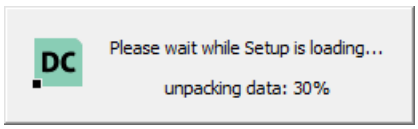


Abbildung - Setup extrahieren

Wahl der gewünschten Setup Sprache.

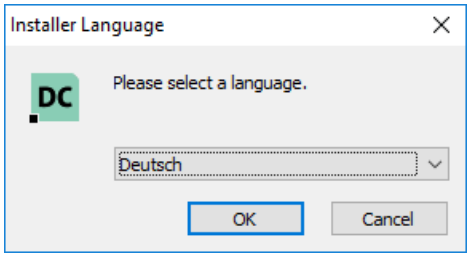


Abbildung - Sprache wählen

Next klicken.

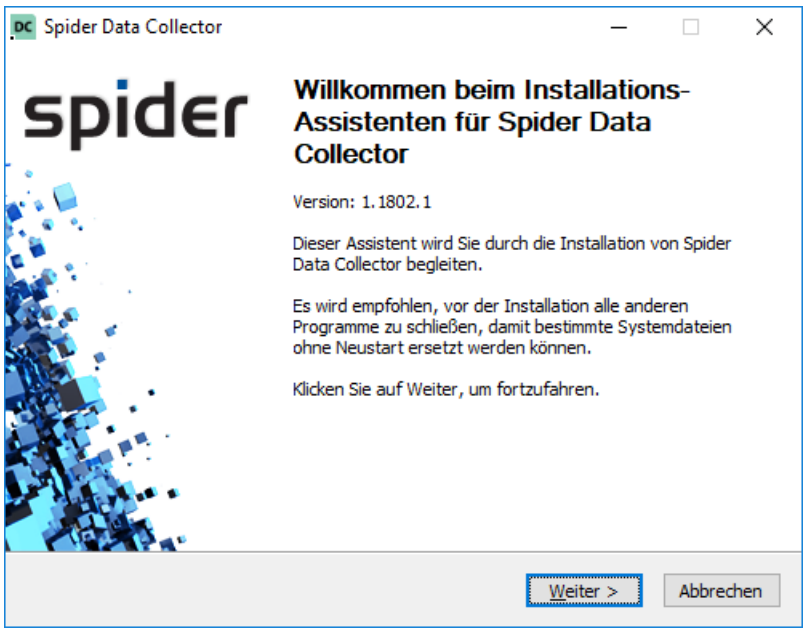


Abbildung - Willkommensseite

Markieren der Auswahlbox um die Lizenzvereinbarung zu akzeptieren.

Abbildung - Lizenzvereinbarung

Dieser Dialog prüft ob die Systemvoraussetzungen gegeben sind, falls dabei Unstimmigkeiten auftreten, werden diese angezeigt. Die Auswahl einer Zeile und das klicken von "Details" führt zu einer Seite der Knowledge Base, auf der Hilfestellungen zur Behebung des Problems angeboten werden.

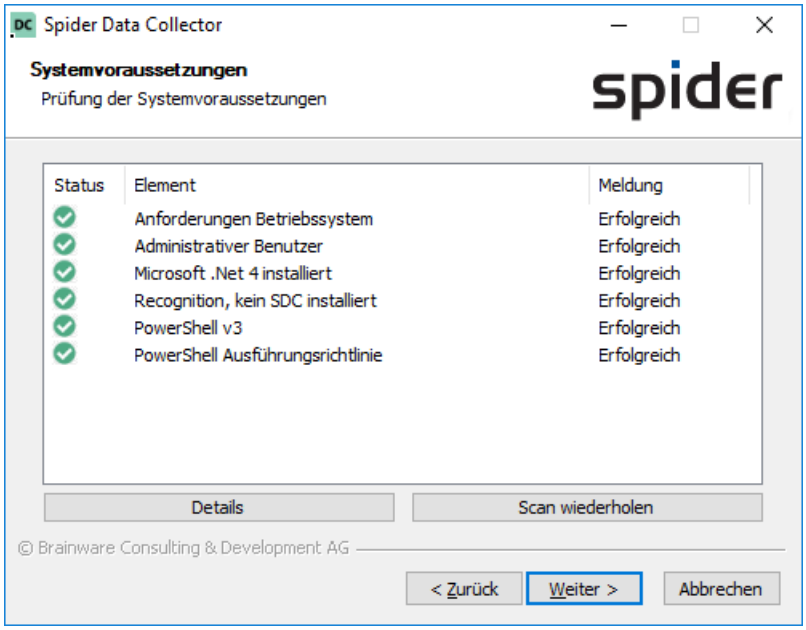


Abbildung - Systemvoraussetzungen

Wahl des Installationspfades, **Next** klicken.

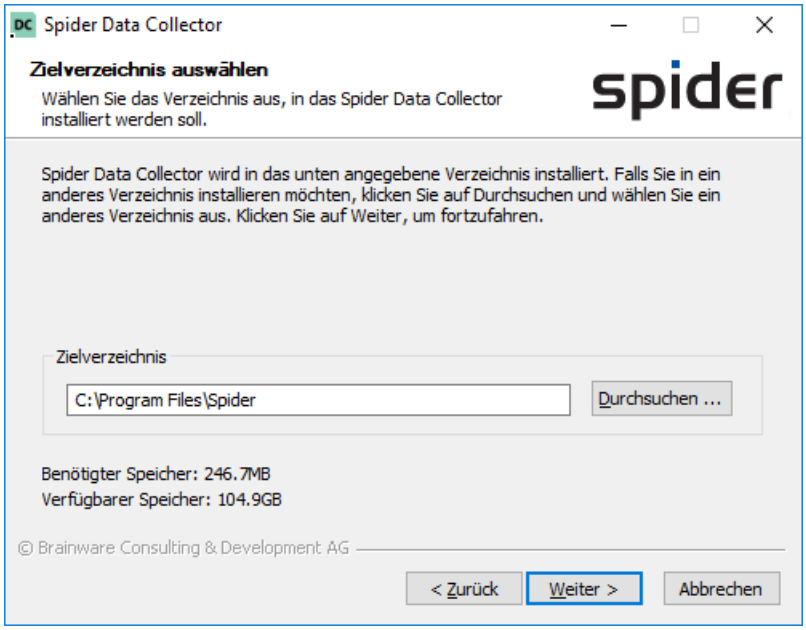


Abbildung - Installationspfad

Angabe des FQDN und Ports auf dem Software Recognition installiert ist. Angabe der CustomerID mit der der Data Collector sich gegenüber der Software Recognition authentifiziert.

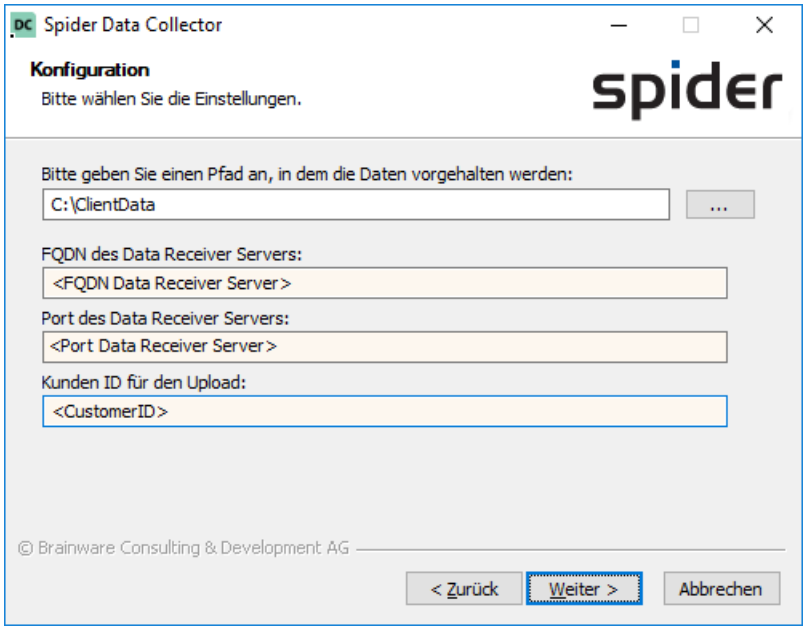


Abbildung - Konfiguration

Notiz Einige Editionen des Data Collectors (Beispielsweise die einer SAM Cloud Plattform), enthalten vorkonfigurierte Werte, es ist also möglich das diese nicht geändert werden können.

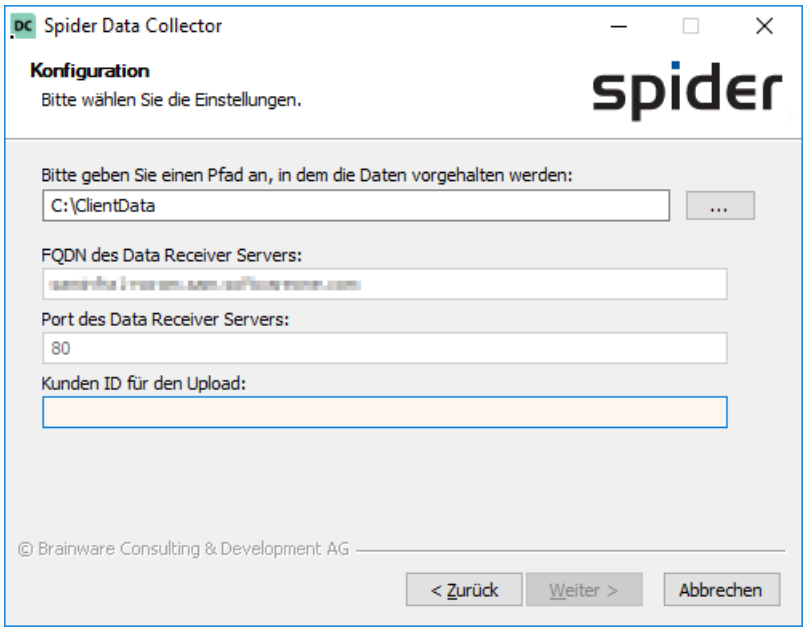


Abbildung - Konfiguration mit vorkonfigurierten Werten

Der Data Collector unterstützt die Kommunikation über einen SOCKS Proxy, ggfs. auch mit einer Authentifizierung. Die aktuelle Version des Data Collectors unterstützt SOCKS Proxys mit den Protokollversionen 4, 4A oder 5. Wie der Screenshot zeigt, müssen bei der Verwendung eines Proxys die IP-Adresse oder der DNS Name des Proxyservers und der Port auf dem der Proxy Server Anfragen erwartet angegeben werden. Wenn eine Authentifizierung nötig ist, muss die Checkbox **Proxy Server verwenden** ausgewählt und die entsprechenden Wert für Benutzer und Passwort angegeben werden.

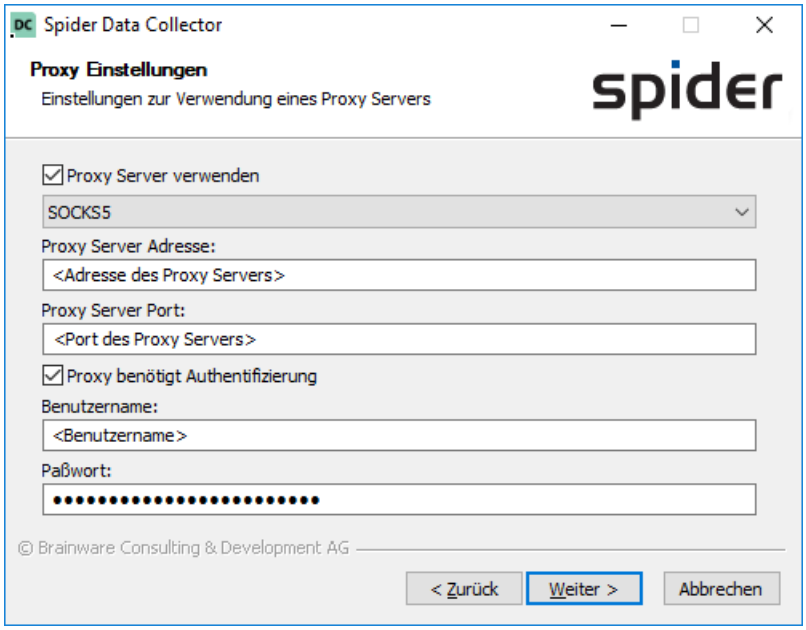


Abbildung - Proxy Konfiguration

Im nachfolgenden Dialog wird der Zeitplan für die Ausführung der Exports und der Übertragung an den Data Receiver.

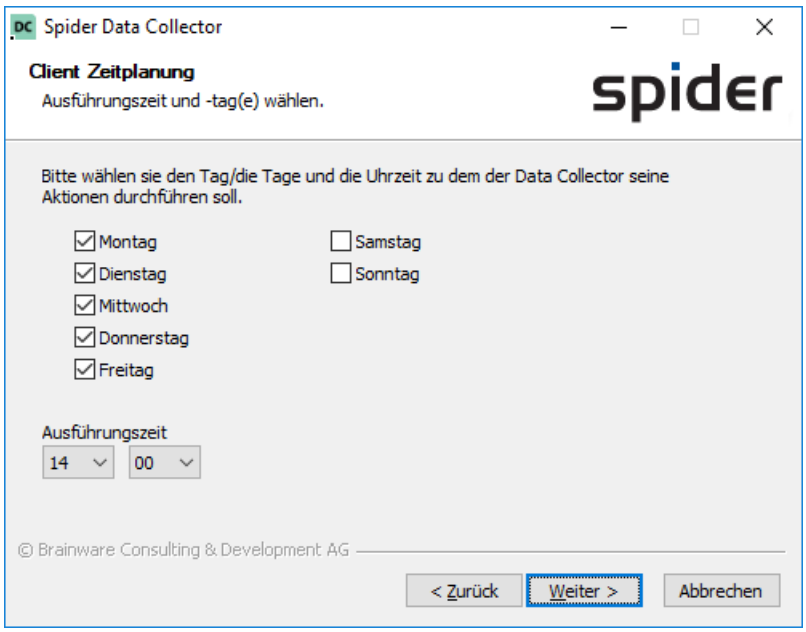


Abbildung - Zeitplan

Angabe des FQDN und Ports für den OTB Server, diese sind in der Regel vorausgefüllt. Zusätzlich kann eine Zeitspanne angegeben werden in der die Inventory Komponenten ihre Inventarisierung starten (nachdem Sie aufgerufen wurden).

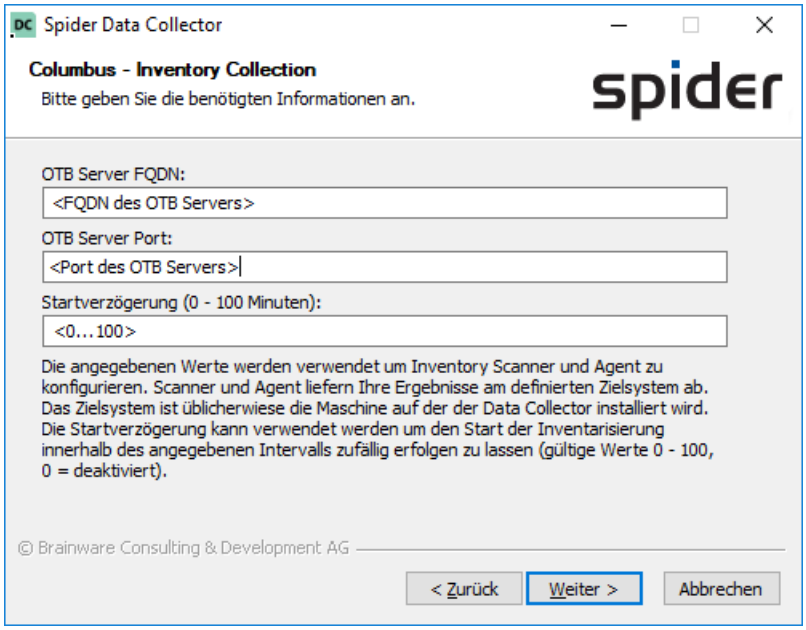


Abbildung - Columbus OTB Einstellungen

Auswahl der Konnektoren die durch das Setup konfiguriert werden sollen.

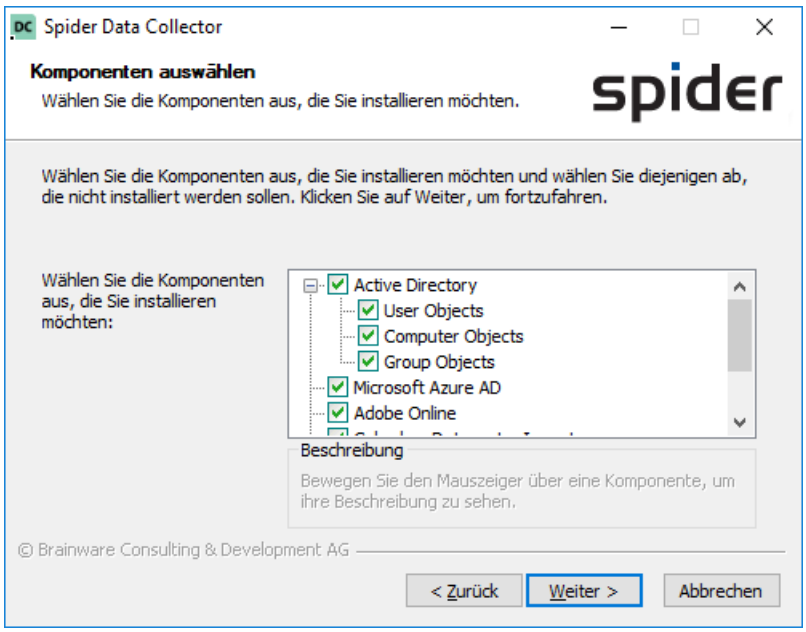


Abbildung - Konnektoren auswählen

In Abhängigkeit zu den gewählten Konnektoren werden bestimmte Systemvoraussetzungen geprüft und ggfs. Informationen dazu ausgegeben.

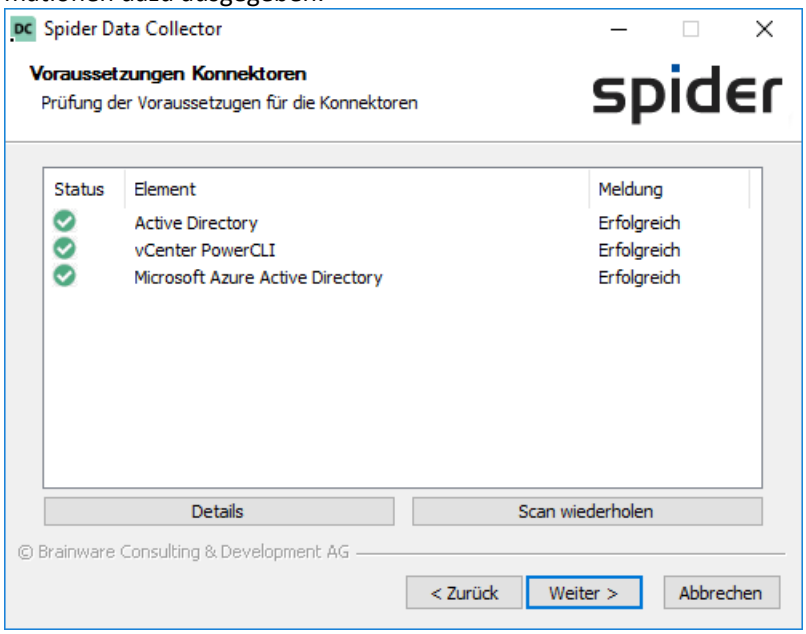


Abbildung - Systemvoraussetzungen der Konnektoren

Wenn der "Active Directory" Konnektor gewählt wurde wird der folgende Dialog angezeigt in dem die benötigten Informationen eingegeben werden müssen. Falls das benötigte PowerShell Modul nicht installiert ist, kann das Setup dies erledigen falls gewünscht.

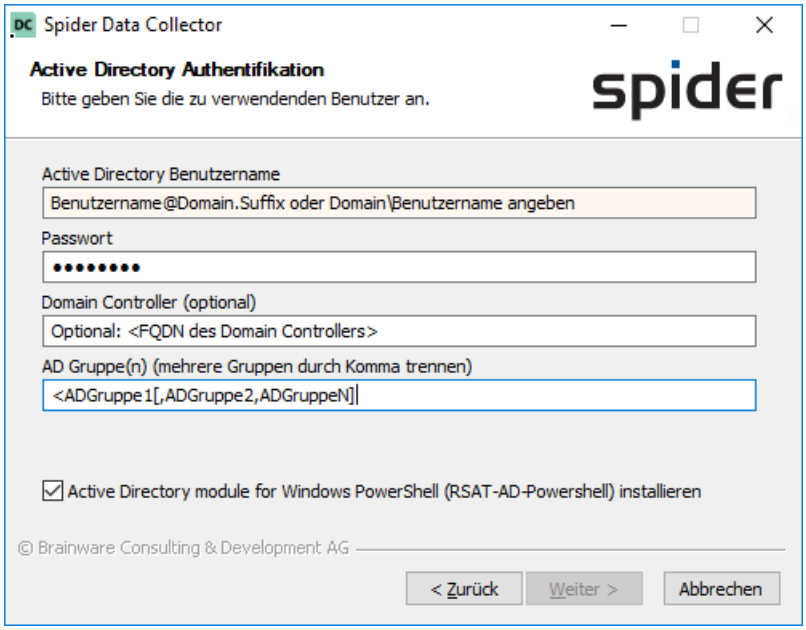


Abbildung - Konfiguration Active Directory Konnektor

Wenn der Microsoft Azure AD Konnektor gewählt wurde, müssen im folgenden Dialog der Benutzer und das dazugehörende Passwort für den Zugriff auf Azure eingegeben werden.

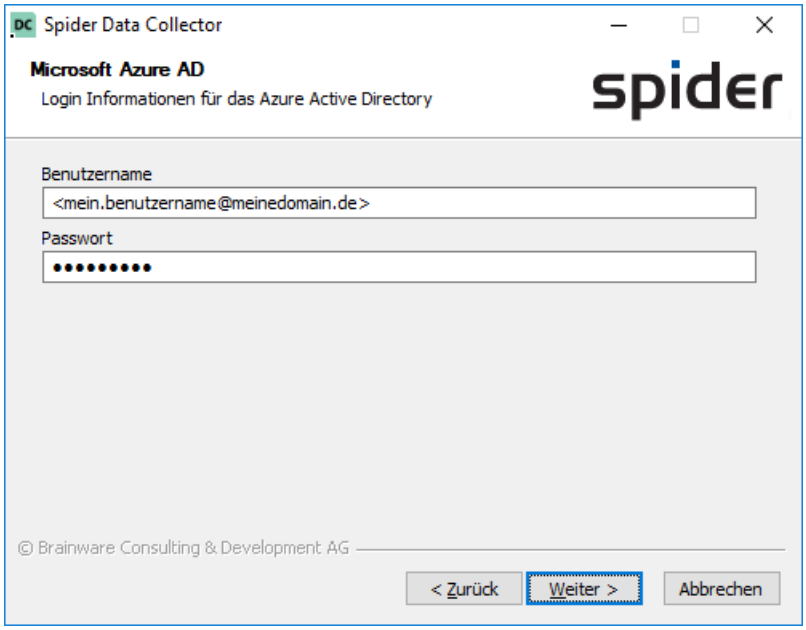


Abbildung - Microsoft Azure AD

Wenn Adobe Online gewählt wurde, werden jetzt die Dialoge zum Erfassen der Zugangsdaten angezeigt.

Es kann entweder ein vorhandenes Schlüsselpaar für die Kommunikation hinterlegt werden, oder selbst ein paar durch das Setup generiert werden. Der öffentliche Teil des Schlüssels muss beim Adobe Portal hinterlegt werden, der private Schlüssel wird durch den Data Collector verwendet.

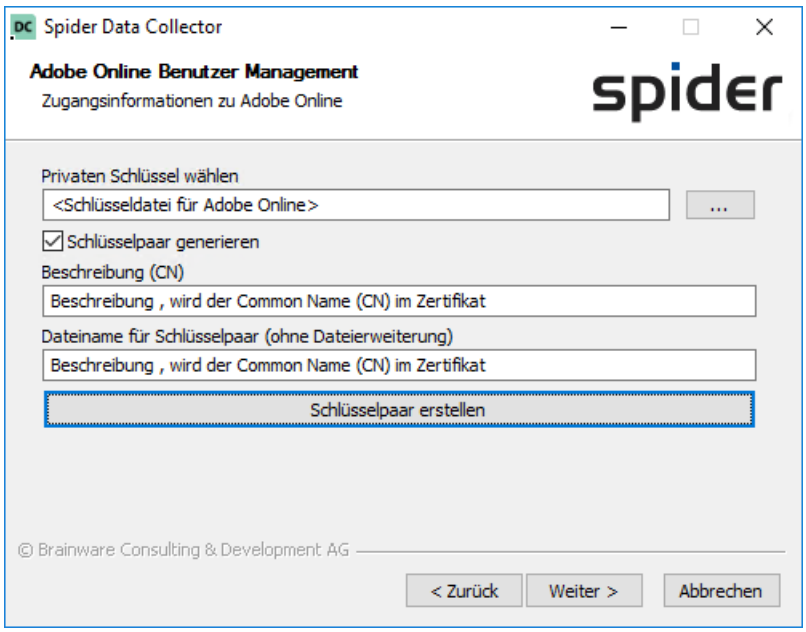


Abbildung - Adobe Online, Schlüsseldaten

Eine erfolgreiche Generierung wird durch folgende Meldung quittiert.

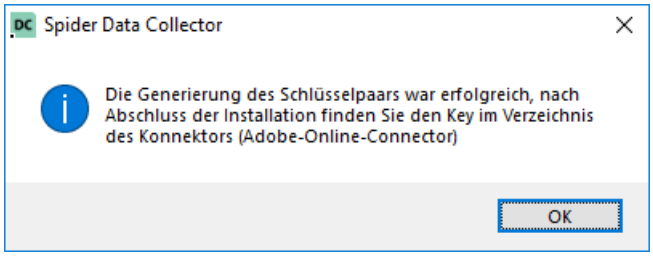


Abbildung - Bestätigung der Schlüsselgenerierung

Nachdem die Angaben zum Schlüssel gemacht wurden, werden nun die Daten der Integration im Adobe Portal abgefragt.

Details zur Einrichtung der Integration sind im Kapitel Adobe Online (auf Seite 63) zu finden.

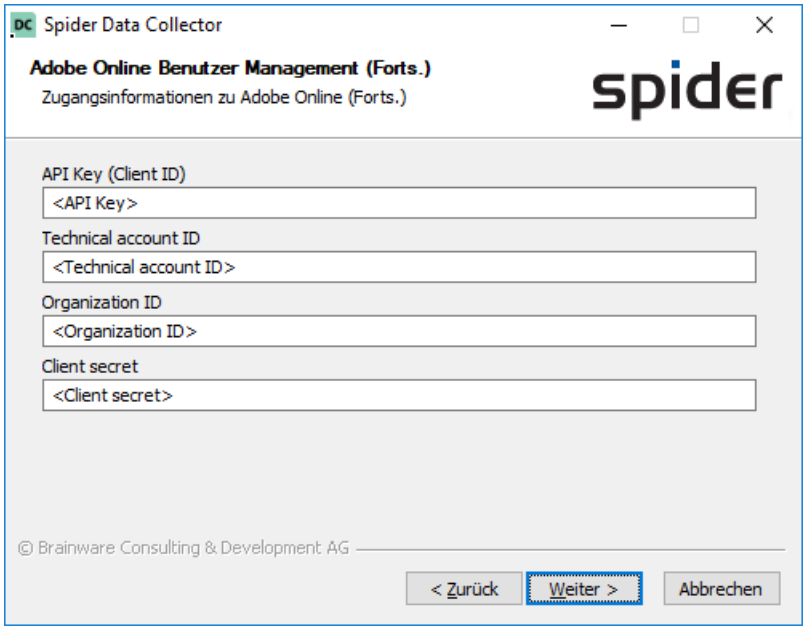


Abbildung - Daten der Adobe Integration

Wenn "Spider Data Center Inventory" ausgewählt wurde, müssen in diesem Dialog die Angaben für den Verbindungsaufbau eingegeben werden. Nach einem erfolgreichen Verbindungstest wird das Setup fortgesetzt.

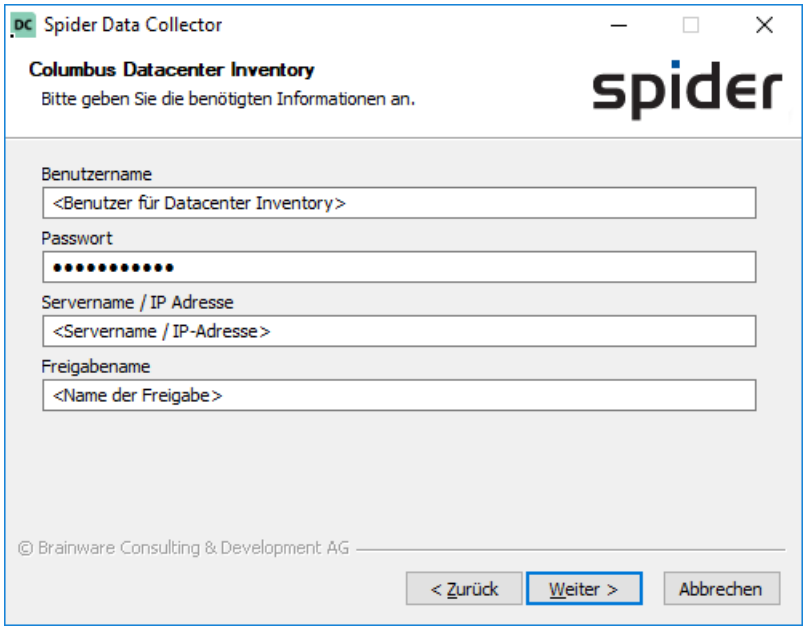


Abbildung - Konfiguration Spider Data Center Inventory

Wenn ein "Database" Konnektor ausgewählt wurde, müssen in diesem Dialog der Type des Inventory Systems sowie die Zugangsdaten eingegeben werden. Wenn die Instanz des Datenbankservers die Standardinstanz ist, muss in diesem Schritt nur die IP-Adresse oder der DNS Name des Servers eingegeben werden. Das Konto für den Zugriff muss entweder ein SQL Benutzerkonto oder ein Windows Domänenkonto sein.

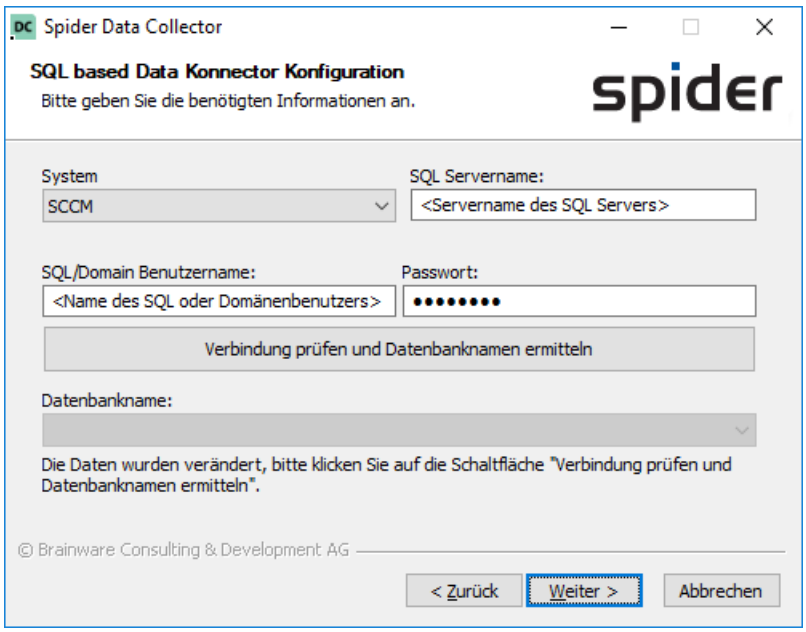


Abbildung - Konfiguration Datenbank Konnektor

Wenn "VMware vCenter" ausgewählt wurde, müssen hier die Daten für den Zugriff eingegeben werden.

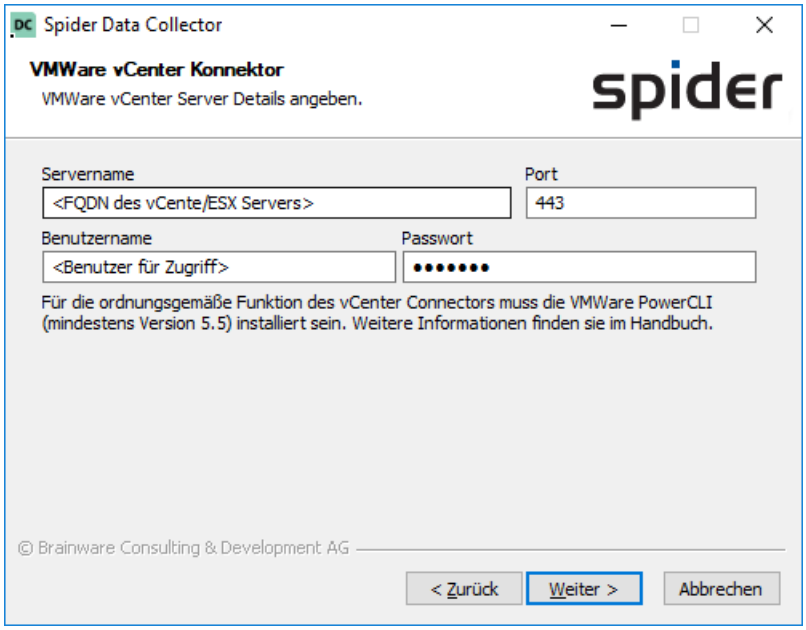


Abbildung - Konfiguration VMware vCenter Konnektor

Falls mehrere Konnektoren ausgewählt wurden, muss nun das Konto gewählt werden unter dem der Data Collector Dienst eingerichtet wird.

Achtung	Wenn ein Domänenbenutzer als Verbindungsbenutzer für den SQL Zugriff spezifiziert wurde, ist dieser vorausgewählt und kann nicht geändert werden. Die Passwörter aller anderen Benutzer die ggfs. während der Konfiguration angegeben wurden, werden verschlüsselt in der Konfiguration abgelegt.

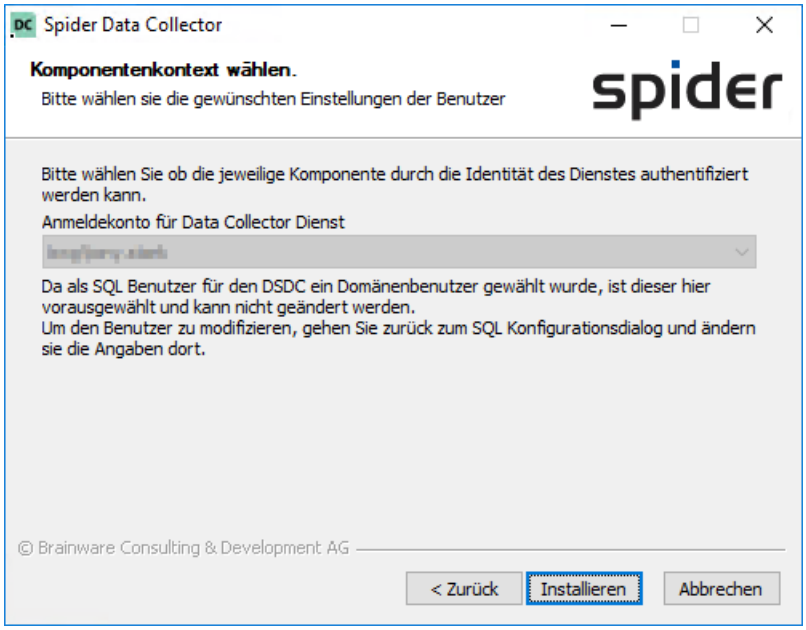


Abbildung - Konfiguration Dienstkonto

Wenn alle Angaben gemacht wurden, kann mit **Installieren** die Installation gestartet werden

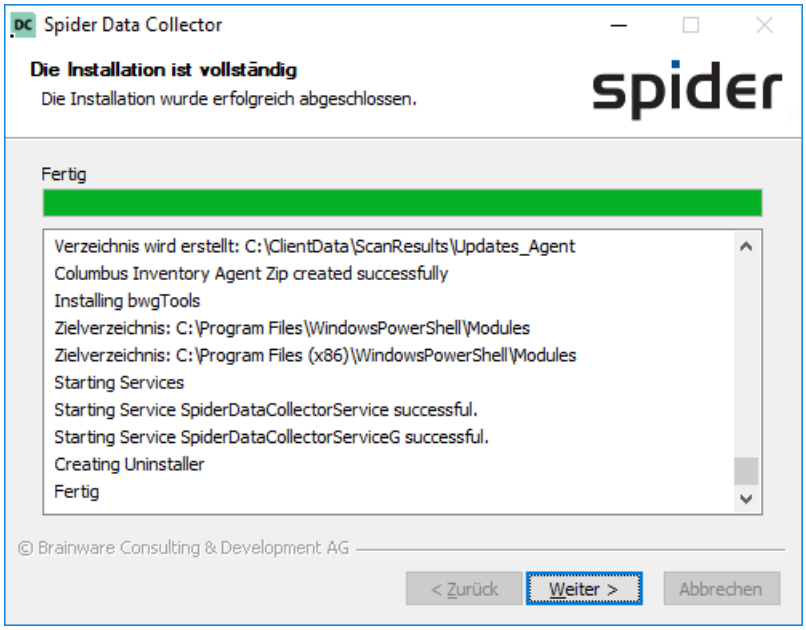


Abbildung - Installation

Sobald die Installation abgeschlossen wurde, kann mit **Beenden** das Setup verlassen werden.

Abbildung - Installation beendet

3.3 Konfiguration

3.3.1 Data Collector

In den meisten Fällen, reichen die während der Installation gemachten Angaben zur Konfiguration aus. Allerdings kann es Situationen geben in denen zusätzliche Konfigurationen gemacht werden müssen, bzw. die bestehende Konfiguration angepasst werden muss.

Alle Einstellungen für den Data Collector sind in der Konfigurationsdatei SpiderDataCollector.cfg abgelegt. Diese Datei liegt im selben Verzeichnis in dem auch der Dienst abgelegt ist ("Data Collector\SpiderDataCollector.cfg").

Sektion	Parameter	Mögliche Werte	Beschreibung
System	CustomerName		Identifiziert den Data Collector gegenüber dem Server und wird verwendet um die übertragenen Daten dem entsprechenden Mandanten zuzuordnen. Der Identifier kann über die Spider Weboberfläche bezogen werden.
System	SystemName		Name der Maschine auf dem der Data Collector installiert ist, wenn es keine Angabe gibt, wird automatisch der Name der lokalen Maschine verwendet.
Connection	OTBHost	<FQDN oder IP des OTB Servers>	Der Server der die Daten des Data Collectors entgegen nimmt. Multiple Endpunkte können durch Komma getrennt angegeben werden.
Connection	OTBPort	<Gültige TCP Port Nummer (0 - 65535)>	Port auf dem die Daten übertragen werden.
Connection	ProxyType	0 = Socks5 1 = Socks4A 2 = Socks4	Angabe welche SOCKS Protokollversion verwendet werden soll.
Connection	ProxyHost	<FQDN oder IP-Adresse des Proxy Servers>	IP-Adresse des Proxy Servers
Connection	ProxyPort	<Gültige TCP Port Nummer (0 - 65535)>	Port des Proxy Servers
Connection	ProxyAuthEnabled	0 = Keine Authentifizierung 1 = Authentifizierung benötigt	Gibt an ob der Proxy Server eine Authentifizierung verlangt.
Connection	ProxyUser		Benutzer für den Proxy Zugriff.
Connection	ProxyPassword		Verschlüsseltes Passwort für den o.g. Benutzer. Das Passwort kann unter Zuhilfenahme der CryptIt.exe verschlüsselt werden die im gleichen Verzeichnis abgelegt wird wie die SpiderDataCollector.cfg
OTBServer	OTBActive	1 = Active andere Werte = Deaktiviert	Gibt an ob der Data Collector auf dem angegeben Port auf Anfragen der Inventory Komponenten antwortet.
OTBServer	DataDirectory		Verzeichnis in dem die durch den Data Collector entgegengenommenen Zip Dateien abgelegt werden.
OTBServer	OTBPort	<Gültige TCP Port Nummer (0 - 65535)>	Port auf dem der Data Collector Anfragen der Inventory Komponenten annimmt.
OTBServer	MaxConnections	Integer	Anzahl der parallelen Verbindungen die der Data Collector zulässt. (Standard: 1000)
Schedule	ScheduleTime	0000-2359	Zeitpunkt zu dem der Data Collector das in Commandline angegebene Skript ausgeführt und im Anschluss daran die Übertragung ausführt. Das Format ist das 24h Format ohne trennenden Doppelpunkt. z.B. muss 17:00 als 1700 angegeben werden.

Sektion	Parameter	Mögliche Werte	Beschreibung
Schedule	ScheduleDaysOfTheWeek	0000000-1111111	Jede Stelle steht für den Wochentag an dem eine Ausführung möglich ist. Die erste Ziffer steht für den Montag, die letzte ist der Sonntag. Beispielsweise steht die Einstellung 0100101 für die Ausführung dienstags, freitags und samstags. Wenn kein Zeitplan angegeben ist, findet keine Verarbeitung statt.
General	Commandline		Kommandozeile die zum Verarbeitungszeitpunkt ausgeführt wird. (Umgebungsvariablen werden aufgelöst)
General	DataDirectory		Verzeichnis in dem die für den Upload generierten Daten bereitgestellt werden.
General	ExecutionTimeOut		Timeout der bestimmt nach wie vielen Minuten angenommen wird das die Verarbeitung fertig sein müssen. Das ausgeführte Skript wird dann abgebrochen und die Übertragung angestoßen.

Wichtig	Es wird nur das SOCKS Protokoll als Proxy Protokoll unterstützt.

Zeitplan

Der Data Collector prüft alle sechs Minuten auf neue Aktionen.

Verzeichnisse

Nach der Installation werden alle Dateien die zum Data Receiver hochgeladen werden müssen in das Upload Verzeichnis kopiert. Das konfigurierte Verzeichnis ist in der Sektion "General" Parameter "DataDirectory" in der SpiderDataCollector.cfg zu finden.

Batch Files

Es gibt Batchdateien die für den Betrieb nötig sind, diese werden alle durch das Setup in den entsprechenden Verzeichnissen abgelegt.

Verzeichnis und Name der Datei	Beschreibung
..\DataCollector\DC\StartDCsPS.cmd	Diese Datei wird durch den Data Collector aufgerufen um den Export der Daten anzustoßen, die für den Upload benötigt werden.

3.3.2 SFTP Server

Für die Aktualisierung und die Ablage der Linux-, Mac-Inventory-Komponenten, ist ein SFTP Server nötig. Dieser SFTP (ColumbusSftpServer.exe) Server wird automatisch mit installiert und durch die SpiderDataCollector.exe gestartet. Beim (ersten) Aufrufen werden die nötigen SSH Schlüssel die in der SpiderDataCollector.json hinterlegt sind automatisch erzeugt, sofern sie nicht vorhanden sind.

Hinweis:	Linux und UNIX Inventory wird jetzt direkt an die Data Center Applicane geliefert.

Die Einträge für die SpiderDataCollector.json sind in der folgenden Tabelle beschrieben:

Sektion	Mögliche Werte	Beschreibung
"uploads": [],		Ist für eine zukünftige Verwendung reserviert und wird derzeit nicht verwendet.
"sftpSettings": {...},		Konfigurationseinstellungen des SFTP Servers.

"sftpSettings": {
"Active": true,
...
}
	true|false	Bestimmt ob der SFTP Server aktive (true) ist oder nicht (false).

"sftpSettings": {
"RootDir": "<Verzeichnis>",
...
}
		Basisverzeichnis des SFTP Servers, in der Regel das "ClientData" Verzeichnis das während der Installation des SDC angegeben wurde.

"sftpSettings": {
"Port": 22,,
...
}
	Integer	Port auf dem der SFTP Server Verbindungen entgegen nimmt.

"sftpSettings": {
"PrivateKey": "<Key>"
...
}
		Dient der Identifikation des Servers gegenüber dem Client. Falls leer, wird dieser Key beim ersten Start automatisch generiert.

"Users": [{User1},{Usern}]		Hier werden die Benutzer definiert die auf den SFTP Server zugreifen dürfen

"Users": [
{
"UserName": "<Be-nutzername>",
...
}
]
		Name des Benutzers der eine Verbindung herstellen darf.

"Users": [
{
"HomeDir": "InvData\\cis",
...
}
]
		Ablageort der gelieferten Dateien und Quelle für eventuelle Updates der Inventory Komponenten

"Users": [
{
"PubKey": "<Key>",
...
}
]
		Public Key des Benutzers der sich verbindet, der Private Key wird mit den Inventory Komponenten verteilt. Falls dieser Wert leer ist wird automatisch ein Key generiert und eingetragen. Eine Kopie des privaten Schlüssels wird im Unterordner "cis" des Data Collectors, für die Verwendung mit den Inventory Komponenten abgelegt.

Sektion	Mögliche Werte	Beschreibung

"Users": [
{
"IsAdmin": false,
...
}
]
	true|false	Bestimmt ob der angegebene Benutzer ein Administrator ist, falls er kein Administrator ist darf er sich nur in dem ihm zugewiesenen HomeDir bewegen.

Achtung	Weder der Private noch der Public Key können durch eigene Schlüssel ersetzt werden. Wenn neue Schlüssel verwendet werden sollen, können die Parameter Pubkey und/oder PrivateKey geleert werden ("") und der Dienst generiert neue Schlüssel. Wenn ein neuer PubKey generiert wird, muss der zugehörige Private Key manuell(!) bei den Inventory Agents ausgetauscht werden Die Datei "cis.prv" wird im Unterordner "cis" des Data Collectors zur Verfügung gestellt. Die Konfiguration der Inventory Komponenten ist im Kapitel Columbus Inventory Scanner Konfiguration (siehe Seite 114) beschrieben.

3.4 Zurücksetzen des letzten Verarbeitungsdatums

Gelegentlich ist es nötig mehr als eine Übertragung pro Tag anzustoßen, dazu muss das Datum zurückgesetzt werden von dem der Data Collector annimmt das die letzte Übertragung stattgefunden hat.

Dies kann durch das zurücksetzen (Löschen) des folgenden Registry Keys erreicht werden.

```
Key: HKEY_LOCAL_MACHINE\SOFTWARE\<Wow6432Node>\Brainware\Columbus\7\OTB\Client
Value: LastScheduledActionCompleted
```

Wichtig	Auch wenn der o.g. Key zurückgesetzt wird muss der Zeitpunkt für die Ausführung des Exports/Übertragung der in der Konfigurationsdatei spezifiziert wurde erreicht sein, damit eine Verarbeitung angestoßen wird. Die Prüfung des Registry Keys bzw. das anstoßen der Verarbeitung wird alle sechs Minuten überprüft.

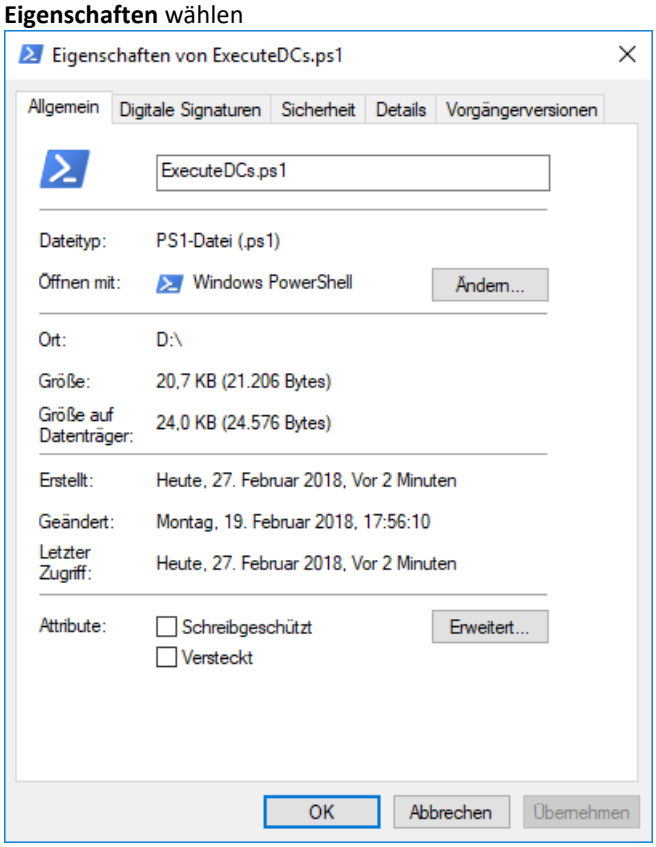
3.5 Hersteller zu Vertrauenswürdige Hersteller hinzufügen

Im Falle das für die Maschine auf der der Data Collector ausgeführt wird, die Maschinen Policy auf "AllSigned" eingestellt ist, ist es nötig den Hersteller des durch die brainwaregroup verwendeten Zertifikates dem Speicherort für Vertrauenswürdige Hersteller hinzuzufügen.

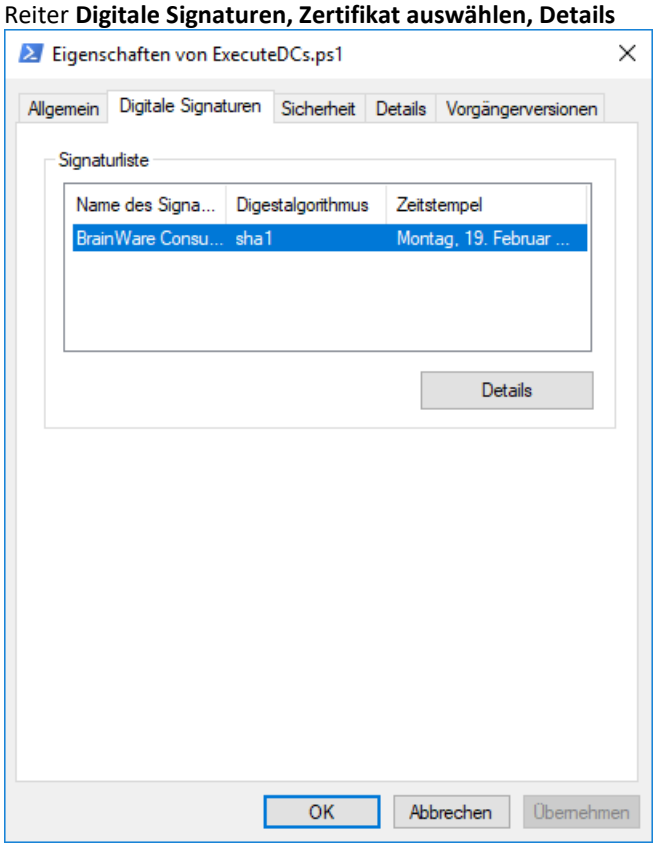
Achtung	Es ist wichtig, dass das Zertifikat dem Speicherort für Vertrauenswürdiger Hersteller der **Lokalen Maschine** hinzugefügt wird, und nicht (nur) dem angemeldeten Benutzer. Nur so ist Gewährleistet das auch andere Benutzer aus dem angemeldeten Benutzer die Informationen über den Hersteller verwenden können.

Um den Hersteller hinzuzufügen, müssen die folgenden Schritte ausgeführt werden.

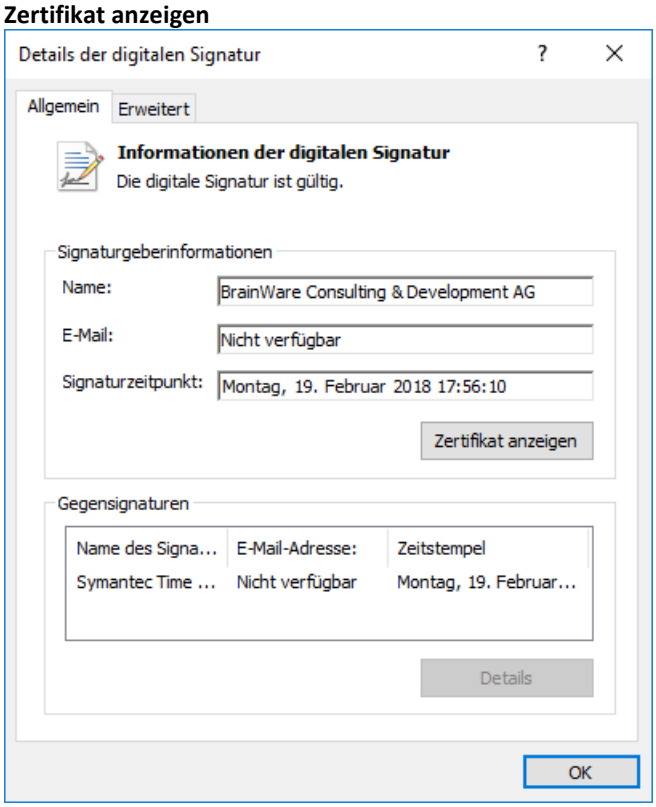
1. Rechts-Klick auf die "ExecuteDCs.ps1" ("DataCollector\DC\ExecuteDCs.ps1"), oder ein anderes signiertes .ps1 Skript, **Eigenschaften** wählen



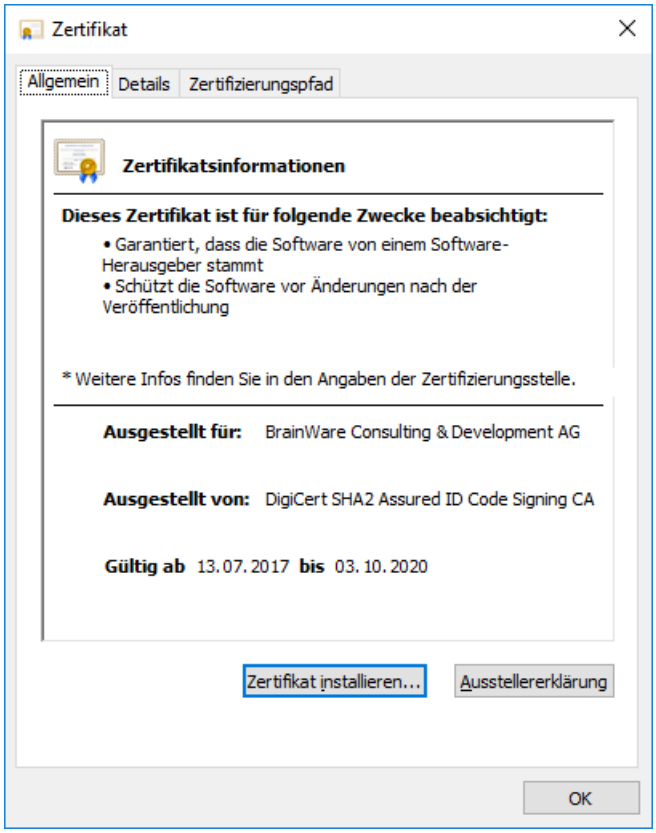
2. Reiter **Digitale Signaturen, Zertifikat auswählen, Details**



3. **Zertifikat anzeigen**

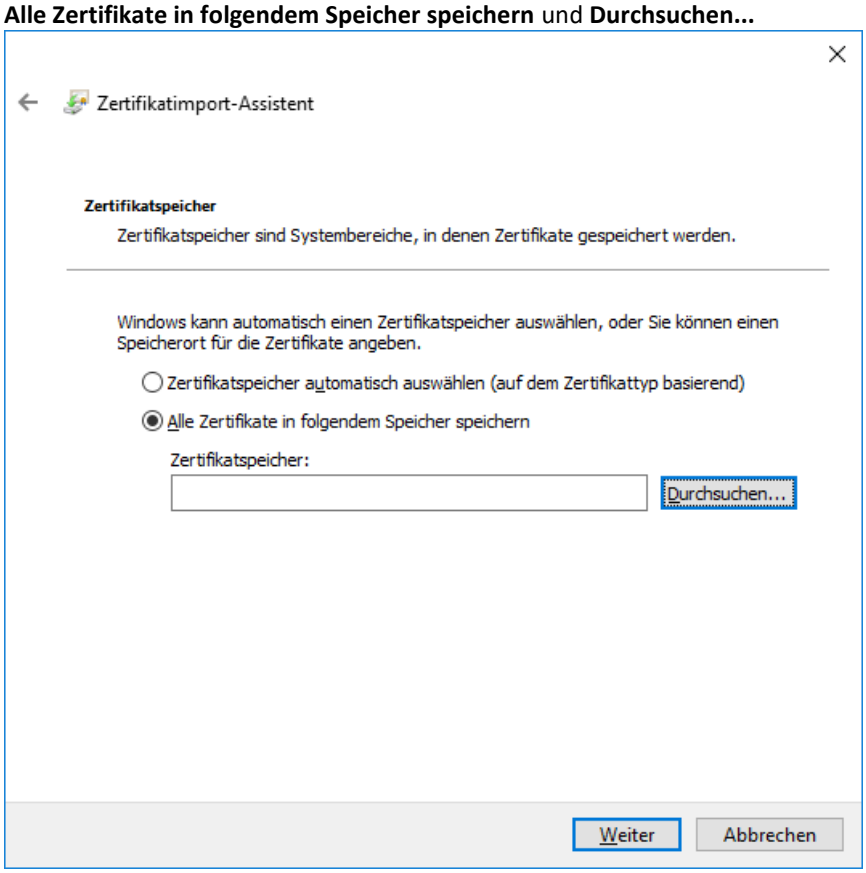


4. **Zertifikat installieren**

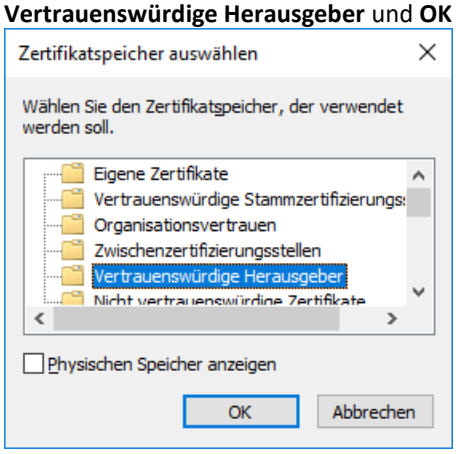


5. **Lokaler Computer** auswählen, **Weiter**

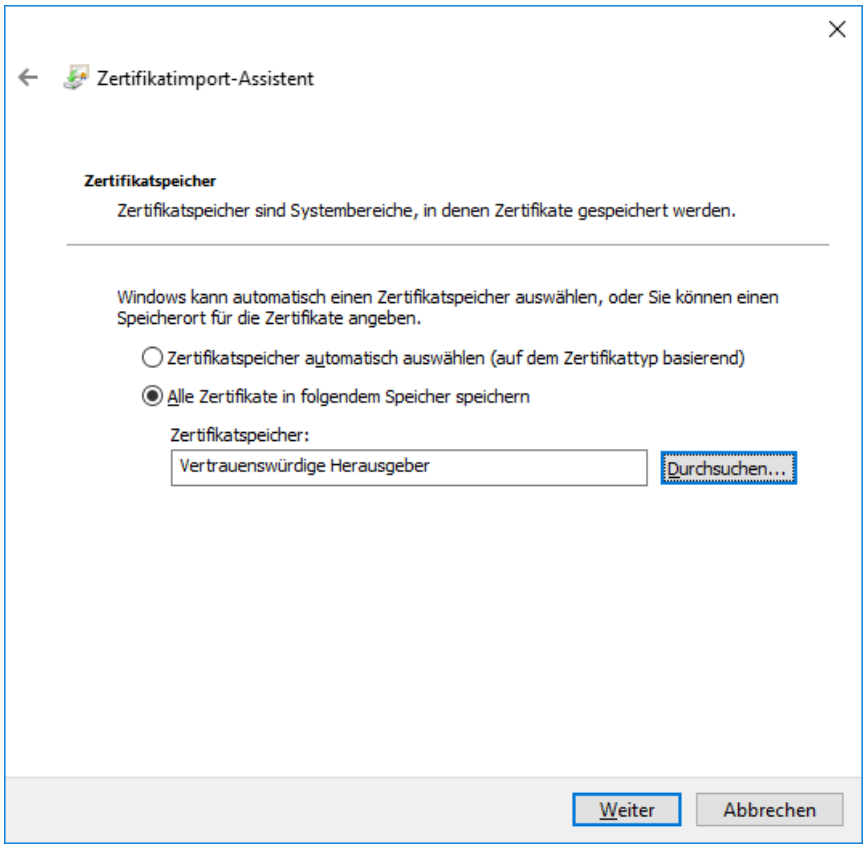
6. **Alle Zertifikate in folgendem Speicher speichern** und **Durchsuchen...**



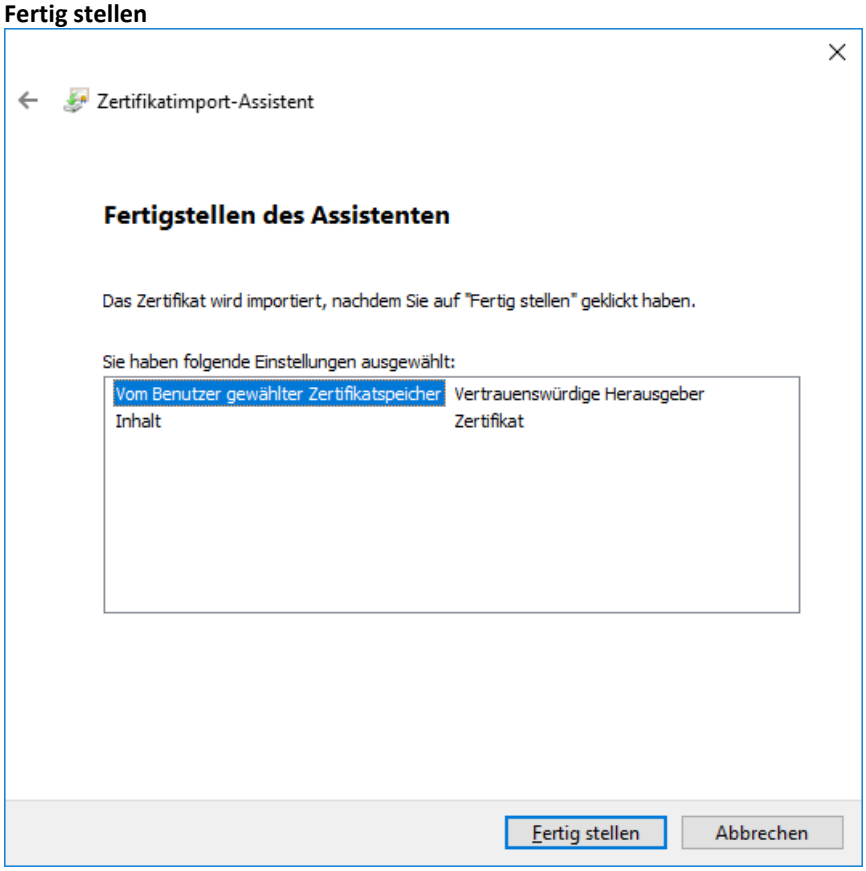
7. **Vertrauenswürdige Herausgeber** und **OK**



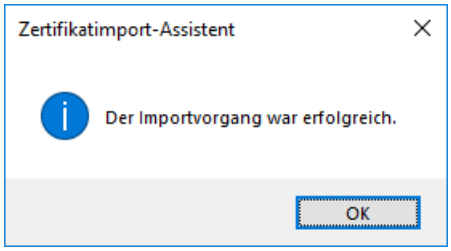
8. **Weiter**



9. **Fertig stellen**



10. **OK**



3.6 Deinstallation

Die Deinstallation des Data Collectors kann auf zwei Wegen erreicht werden, entweder durch das Ausführen der "Spider Data Collector_Uninstall.exe" im Installationspfad oder über den entsprechenden Eintrag in der Installierten Software des Computers.

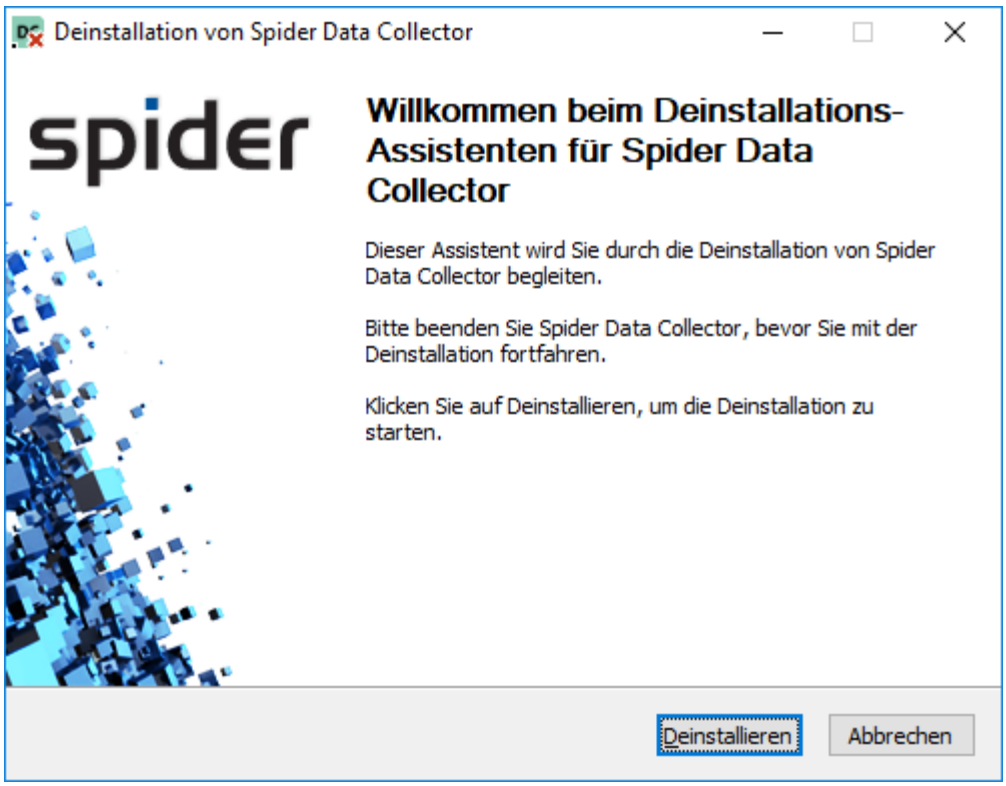


Abbildung - Willkommensseite

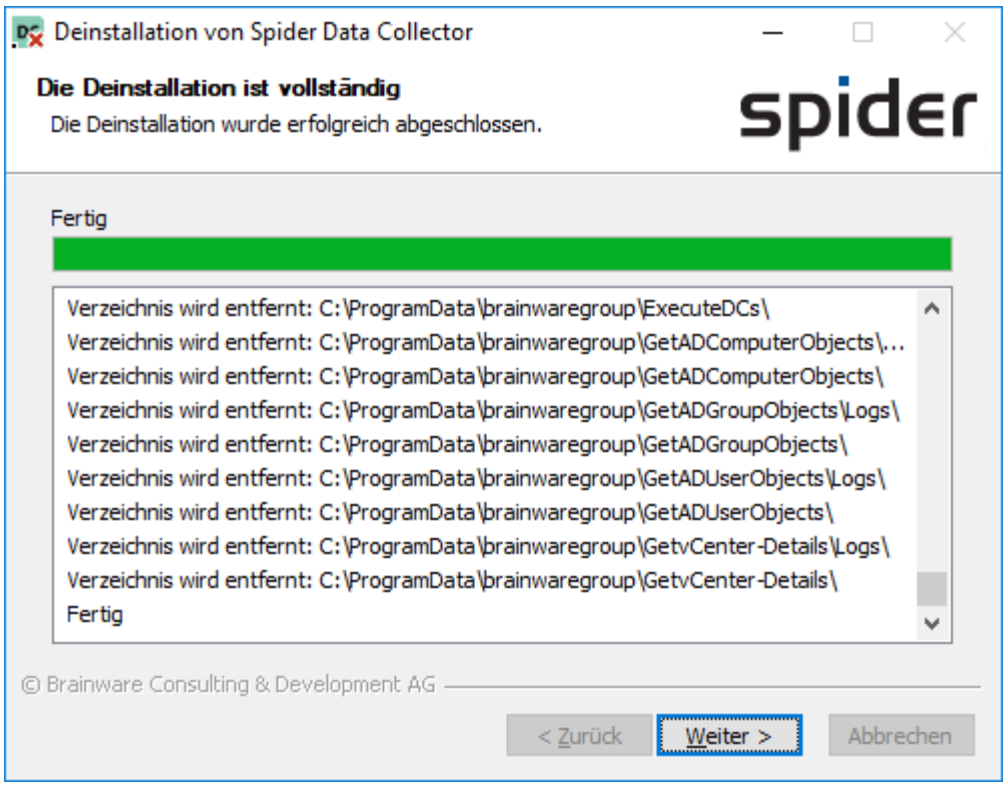


Abbildung - Deinstallation

Abbildung - Ende der Deinstallation

3.7 Behebung von Verbindungs-/Authentifizierungsproblemen

Wenn der Data Collector keine Inventardaten generiert und/oder diese nicht überträgt, sind die folgenden Punkte zu überprüfen.

Als erstes muss das brainware.log (in %windir% oder %ProgramData%\Columbus) überprüft werden.

Common Messages

Mitteilung	Beschreibung
SpiderDataCollector: Connecting to myserver:myport	gibt an das versucht wird eine Verbindung herzustellen.
SpiderDataCollector: Connection to myserver:myport has succeeded	Die Verbindung konnte erfolgreich hergestellt werden.
Connection from [EDC Client Manager] to [myserver:myport] failed using IP v4 [Socket Error # 10061; Connection refused.], trying IP v6	Bei Fehlern wie diesen muss überprüft werden ob Adresse und Port des Zielservers korrekt angegeben sind. Zusätzlich müssen evtl. dazwischenliegende Firewalls geprüft werden.
Connection from [EDC Client Manager] to [myserver:myport] failed using IP v6 [Socket Error # 11001; Host not found.]	
SpiderDataCollector: [ERROR] - Problem connecting to the OTB server on myserver:myport with message: Socket Error # 10061; Connection refused.	
SpiderDataCollector: Authenticating with myserver:myport]	Versuch der Authentifizierung
SpiderDataCollector: Authentication: client [{A74192A6-BF66-49F2-8271-90EEBEE61BDF}: <CustomerID>:<Servername>:7.5.2.39] is active on the OTB server [<Servername>.<Port>]	Der Data Collector konnte erfolgreich eine Verbindung zum Data Receiver herstellen und hat sich ebenfalls erfolgreich Authentifiziert.
SpiderDataCollector: Authentication: client [{AE97A9DC-5DFE-442B-B448-56ED1B92BDB1}: <CustomerID>:<Servername>:7.5.2.39] is new pending registration and activation with the OTB server [<Servername>.<Port>]	Gibt an das der Data Collector sich neu an der Maschine angemeldet hat und darauf wartet das die Anmeldung bestätigt wird.
SpiderDataCollector: [WARNING] - Failed authentication: Authentication: [{AE97A9DC-5DFE-442B-B448-56ED1B92BDB1}:<CustomerID>:<Servername>:7.5.2.39] has failed to authenticate with the OTB server [<Servername>.<Port>]	Gibt an das die Authentifizierung fehlgeschlagen ist. Das liegt in der Regel daran, dass die CustomerID auf dem Server unbekannt oder deaktiviert ist. In diesem Fall muss überprüft werden, ob die Recognition für diesen Mandanten aktiv ist und die korrekte CustomerID verwendet wird,

Achtung Ab Windows 2012 R2 kann der folgende PowerShell Befehl verwendet werden um zu prüfen, ob die Maschine auf der der Data Collector installiert ist, den Port auf der Maschine, auf der Recognition installiert ist, erreichen kann.
Mehr Informationen sind hier zu finden:
https://community.flexera.com/t5/Spider-Knowledge-Base/Operations-Manager-How-to-troubleshoot-Network-connections/ta-p/4791

```
Test-NetConnection -Computername <ServerName> -Port <Port> -InformationLevel Detailed
```

Notiz Das Datumsformat der brainware.log Datei kann sich in Abhängigkeit der Regionaleinstellungen der Maschine unterscheiden.

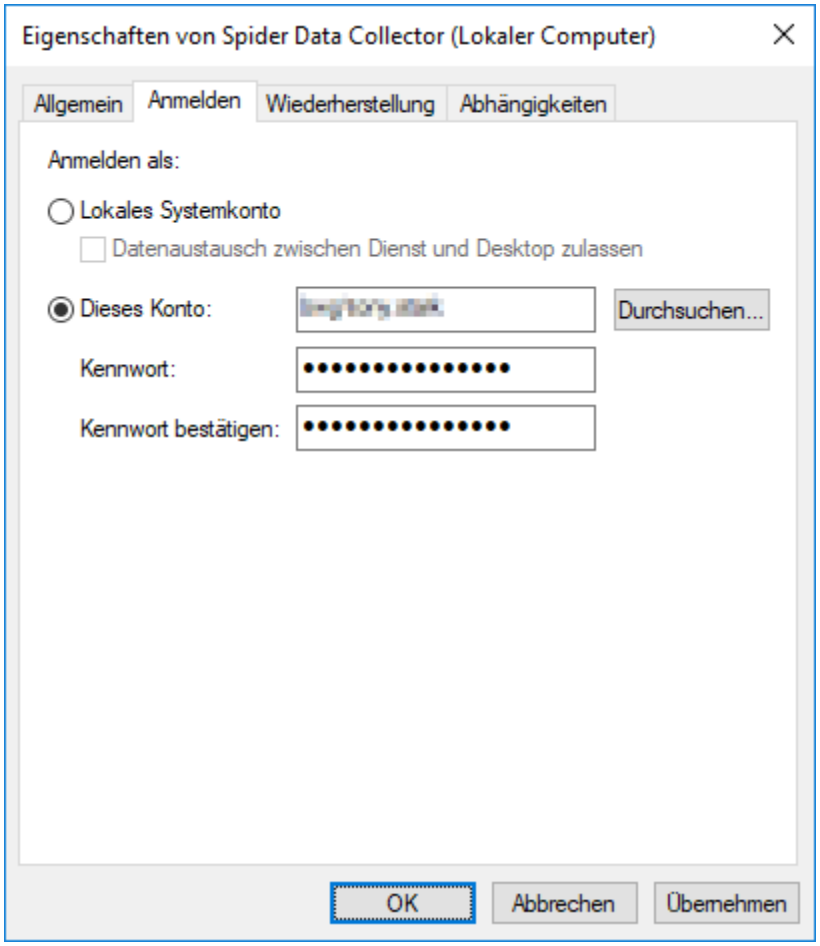
3.8 Dienstkonto des Data Collectors ändern

Bei der Verwendung der Impersonation für bestimmte Konnektoren (z.B. für den SQL Datenbankzugriff oder den Active Directory Konnektor), verwendet der Data Collector Dienst das Konto das während dem Setup angegeben wurde.

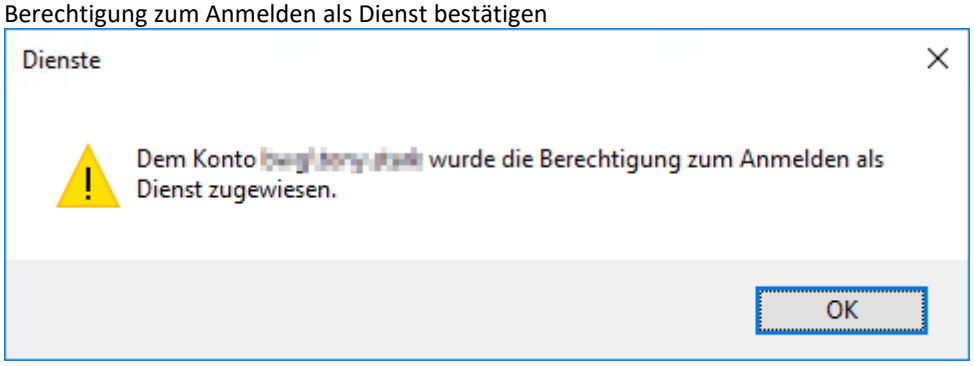
Manchmal ist es nötig dieses Konto zu ändern, um dies zu bewerkstelligen können die folgenden Arbeitsschritte ausgeführt werden:

1. Stoppen des Data Collector Guardian Dienstes über die (als Administrator gestartete) Kommandozeile mittels "net stop SpiderDataCollectorServiceG" oder über den Service Manager.
2. Stoppen des Data Collector Dienstes über die (als Administrator gestartete) Kommandozeile mittels "net stop SpiderDataCollectorService" oder über den Service Manager.

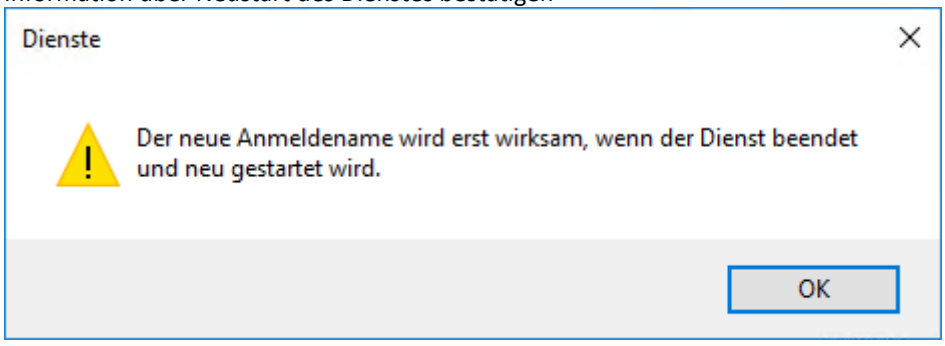
3. Öffnen der Konfiguration des "Spider Data Collector" Dienstes, Auswahl des Reiters "Log on".
4. Ändern der Konteninformationen und mit "OK" bestätigen.



5. Berechtigung zum Anmelden als Dienst bestätigen



6. Information über Neustart des Dienstes bestätigen



7. Änderung der Sicherheitseinstellungen im Pfad in dem die SpiderDataCollector.exe abgelegt ist, so dass der neue Benutzer Lese/Schreibrechte für diesen Ordner und die darunterliegenden Ordner erhält. Der Pfad in dem die SpiderDataCollector.exe abgelegt ist, kann aus den Serviceinformationen in der Registry ermittelt werden.

   `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\brainwaregroup\DataCollector Wert: Path`

8. Ändern der Sicherheitseinstellungen des Data Collector Datenpfades so dass der neue Benutzer Lese/Schreibberechtigungen in diesem Ordner und den darunterliegenden Ordnern hat. Der Datenpfad kann aus der SpiderDataCollector.cfg (Sektion: [General] Wert: Datadirectory) im Data Collector Dienstpfad (Siehe Schritt 5.) oder aus der Registry (siehe unten) ermittelt werden.

   `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\brainwaregroup\DataCollector Wert: DataCollectorDataPath`

9. Änderung der Registry Berechtigungen so dass der neue Benutzer "Full Control" Rechte auf dem Key

   `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BrainWare\Columbus\7\OTB\Client` erhält.

10. Starten des Data Collector Dienstes über die (als Administrator gestartete) Kommandozeile mittels "net start SpiderDataCollectorService" oder über den Service Manager.
11. Starten des Data Collector Guardian Dienstes über die (als Administrator gestartete) Kommandozeile mittels "net start SpiderDataCollectorServiceG" oder über den Service Manager.
12. Fertig!

Wichtig	Sollte der neue Benutzer kein Administrator der Maschine sein, ändert sich der Ablageort der Brainware.log Datei vom Windows Verzeichnis auf %ProgramData%\Columbus\Brainware.log

Konfiguration der Konnektoren

Achtung	Ab Version 1.1609 des DC, wurde das Management der Konnektoren verändert. Dieses Kapitel enthält wichtige Informationen dazu! Die automatische Migration der Konnektoren von Versionen vor 1.1609 berücksichtigt nur Konfigurationen die in den Standardverzeichnissen stattfinden. Jedwede manuelle Änderung wird überschrieben bzw. ignoriert.

Bis Version 1.1608, wurden die Konnektoren durch verkettete .cmd Dateien ausgeführt. Dieses Vorgehen wurde durch ein PowerShell Skript ersetzt, das eine zentrale Konfigurationsdatei ausliest.

Die Konfigurationsdatei (Connector.config) ist im Ordner "DC" abgelegt und enthält eine XML Struktur die pro auszuführendem Konnektor einen <connector> Eintrag enthält.

```
<?xml version="1.0" encoding="utf-8" ?>
<connectors>
    <connector name="DSDC SCCM Inventory" subfolder="DSDC" active="true" scriptname="DSDC.exe" srv="servername" db="database name"
t="SCCM" uid="UserID" pw="Password" h="true" s="true" f="true" sfx="_SCCM" />
    <connector name="vCenter Inventory" subfolder="vCenter" active="true" scriptname="GetvCenter-Details.ps1" srv="servername"
port="port number" uid="domain\username" pwd="password"  h="true" s="true" dr="true" sfx="suffix" />
    <connector name="ADUserObjects" subfolder="ADConnector" active="true" scriptname="GetADUserObjects.ps1"
uid="domain\username" pwd="password"  dc="" sfx="" filter="" ou="" />
    <connector name="ADComputerObjects" subfolder="ADConnector" active="true" scriptname="GetADComputerObjects.ps1"
uid="domain\username" pwd="password"  dc="" sfx="" filter="" ou="" InactiveDays="" />
    <connector name="GetADGroupObjects" subfolder="ADConnector" active="true" scriptname="GetADGroupObjects.ps1"
uid="domain\username" pwd="password"  dc="" sfx="" grp="group[,group]" strict="true" />
```

```
    <connector name="DataConnectorColumbus" subfolder="Columbus" active="true" scriptname="DataConnectorColumbus.exe"
noflag="true" nouploaddir="true" />
    <connector name="Spider Data Center Inventory" subfolder="DatacenterInventory" active="true"
scriptname="GetDatacenterInventory.ps1" uid="erunbook" server="10.1.2.3" share="spider" sfx="" />
    <connector name="Hyper-V" subfolder="Hyper-V" active="false" scriptname="GetHyper-VDetails.ps1" uid="" pwd="" srv="" sfx=""
/>
</connectors>
```

Die Konfigurationsdatei enthält eine Reihe von gemeinsamen Attributen pro Konnektor (siehe folgende Tabelle) und spezielle Einträge die pro Konnektor unterschiedlich sein können.

Allgemeine Einträge:

Attribut	Beschreibung
name="<Name des Konnektors>"	Name des Konnektors, dieser kann modifiziert werden um z.B. mehrfach den gleichen Typ von Konnektor auszuführen. Z.B. bei der Abfrage mehrerer SCCM Datenbanken.
subfolder="<Unterordner>"	Unterordner (relativ zu "DC") in dem das Skript oder die ausführbare Datei für den Konnektor abgelegt ist.
active="<true|false>"	(De-)Aktivierung des Konnektors.
scriptname="<Name des Skriptes oder der ausführbaren Datei>	Name des Skriptes oder der ausführbaren Datei.
timeout="Anzahl Sekunden"	Timeout nachdem die Verarbeitung des jeweiligen Konnektors gestoppt wird. Achtung, manche Konnektoren können sehr lange laufen, dies ist Abhängig von der Art und Größe der ermittelten Inventardaten, ein zu kurzer Timeout würde die Verarbeitung der Daten abbrechen!

Andere Attribute als die gemeinsamen Attribute für die Konnektoren werden direkt dem Konnektor übermittelt und können den Kapiteln über die jeweiligen Konnektoren entnommen werden

Allgemeine Hinweise zur Konfiguration:

- Die Reihenfolge der Verarbeitung wird durch die Reihenfolge der Einträge in der Connector.config Datei bestimmt.
- Das Directory Attribut des jeweiligen Konnektors (-dir oder ähnliches) darf nicht in der Connector.config angegeben sein, dieser wird automatisch aus der SpiderDataCollector.cfg ermittelt und dem Aufruf hinzugefügt.
- True|False Parameter (z.B. /h für den Hardware Scan der DSDC.exe) müssen als h="true" in den Konnektor Attributen angegeben sein, wenn der Parameter "false" sein soll, darf er gar nicht angegeben werden. (Also entweder h="true" oder Garnichts).
- Bei den PowerShell basierenden Konnektoren ist es möglich evtl. verwendete Passwörter zu verschlüsseln, Details dazu können in Passwortverschlüsselung bei PowerShell Konnektoren (siehe Seite 56) gefunden werden. Wenn die Passwortverschlüsselung verwendet wird, darf der Eintrag pwd="<Passwort>" nicht verwendet werden, pwd="" ist ebenfalls nicht erlaubt.
- Wenn mehr als ein Konnektor desselben Typs verwendet wird (z.B. um zwei oder mehr SCCM Datenbanken abzufragen), muss der Suffix jedes Konnektor Eintrags unterschiedliche Werte aufweisen, ansonsten überschreibt die Ausgabe des nächsten Konnektors die Ausgabe des vorherigen Aufrufs.

Notiz Die Ausführung des Management Skripts wird nach %ProgramData%\ExecuteDCs\ExecuteDCs.log geschrieben.

Schutz von Sonderzeichen

Wenn aus irgendeinem Grund Sonderzeichen in der Connector.config verwendet werden, müssen diese entsprechend geschützt (Escaped) werden, die folgende Tabelle zeigt gültige Ersetzungen:

Zeichen	Geschütztes Zeichen
<	&lt;
>	&gt;
&	&amp;
"	&quot;
'	&apos;

4.1 Passwortverschlüsselung bei Konnektoren

Die Konnektoren unterstützen die Verschlüsselung des Passwortes in einer Datei.

Um die Erstellung der nötigen Dateien zu unterstützen wurde das Skript "EncryptPassword.ps1" im Ordner "DC" des Data Collectors abgelegt, das Skript kann über eine PowerShell-Konsole wie folgt ausgeführt werden:

```
PowerShell.exe -executionpolicy remotesigned -File EncryptPassword.ps1 -uid "<Benutzer>" -pwd "<Passwort>"
```

Als Ergebnis wird eine Dateie erstellt

- <Benutzer-Hostname>.pwd
  Die Passwort Datei kann nur auf derselbe Maschine verwendet, wo sie erstellt wurde.

Das bisherige Passwortverschlüsselungsverfahren bleibt bestehen. Wenn dieses genutzt werden soll kann das Skript über eine PowerShell-Konsole wie folgt ausgeführt werden:

```
PowerShell.exe -executionpolicy remotesigned -File EncryptPassword.ps1 -uid "<Benutzer>" -pwd "<Passwort>" – mkey 1
```

Als Ergebnis werden zwei Dateien erstellt:

- <Benutzer>.key
- <Benutzer>.pwd

Achtung	Die erstellten Dateien müssen entweder direkt im Ordner des Konnektors der sie verwenden soll, oder im darüber liegenden Verzeichnis abgelegt werden. Das Modul das die Entschlüsselung des Passwortes verarbeitet sucht zuerst im Verzeichnis des jeweiligen Konnektors und sollten die Dateien dort nicht gefunden werden, eine Ebene höher nach den entsprechenden Dateien ("..\DC")

Die Connector.config Datei muss zur Verwendung des verschlüsselten Passwortes editiert werden, und für den betroffenen Konnektor muss das Attribut **pwd="<Passwort>"** entfernt werden. Danach verwendet der Konnektor die verschlüsselte Passwortinformation.

Achtung	Es werden immer beide Dateien (.key und .pwd) benötigt damit die Entschlüsselung korrekt funktioniert. Die Verschlüsselung ist reversibel, es muss also sichergestellt sein, dass kein unberechtigter Zugriff auf die Maschine möglich ist.

4.2 Datenbankbasierende Konnektoren

Einstellungen und Voraussetzungen für die Abfrage von Datenbanken werden in den folgenden Kapiteln beschrieben.

4.2.1 Discovery Systems Data Connector (DSDC.exe)

Das Programm um Inventardaten aus SQL Datenbanken zu exportieren kann im Unterordner "\DataCollector\DC\DSDC" des Data Collector Installationsverzeichnisses gefunden werden.

Spezifische Details über die verschiedenen Exports werden in den folgenden Kapiteln beschrieben.

Das Programm kann über die folgenden Parameter gesteuert werden:

Kommandozeile	Connector.config	Beschreibung
/srv:<Server>	srv="<Server>"	Name des SQL Servers
/db:<Datenbank>	db="<Datenbank>"	Name der Inventardatenbank
/lnk:<LinkedServer>	lnk="<LinkedServer>"	Optional: Linked Server Name

Kommandozeile	Connector.config	Beschreibung
/uid:<SQL Benutzer>	uid="<SQL Benutzer>"	SQL Benutzer für den Zugriff (Falls nicht angegeben, wird das aufrufende Benutzerkonto verwendet.)
/pw:<Passwort>	pw="<Passwort>"	Passwort für o.g. Benutzer
/t:<Inventory type>	t="<Inventory type>"	Typ des Inventarsystems: SCCM, LANDesk, Map, Discovery, GENERIC
/dir:<ExportDir>	nicht verwendet	Exportverzeichnis
/sfx:<File suffix>	sfx="<File suffix>"	Optional: Suffix für die Exportdatei
/h	h="true"	Hardware exportieren
/f	f="true"	Dateiinformationen exportieren
/s	s="true"	Installierte Software exportieren
/tmp:<Tempdir>	tmp="<Tempdir>"	Optional: Verzeichnis in dem die temporären Dateien abgelegt werden.
/kb	kb="true"	Optional: Unterdrücken der Microsoft Updates bei der Ausgabe der installierten Software
/ad	ad="true"	Export der Active Directory Benutzer (nur gültig für GENERIC Konnektor)
/dr	dr="true"	Export der Device Beziehungen (nur gültig für GENERIC Konnektor)
/m	m="true"	Metering Informationen exportieren (nur SCCM)
/adg	adg="true"	Export der Active Directory Gruppen (nur gültig für GENERIC Konnektor)

Notiz Wenn der DSDC im gleichen Benutzerkontext wie der Data Collector ausgeführt werden soll (Export über Domänenbenutzer) dann dürfen die Parameter für Benutzer (/uid, uid=) sowie für das Passwort (/pw, pw=) nicht angegeben sein.

Die Angabe eines Domänenbenutzers über die "uid" and "pw" Parameter ist nicht möglich, dort sind nur SQL Konten erlaubt!

4.2.2 Ausgabe von MAC und IP Informationen unterdrücken (DSDC.exe.config)

Wichtig Seit dem Release 1.1805 wird im Verzeichnis der DSDC.exe die Datei DSDC.exe.config ausgeliefert, über diese Datei lässt sich die Ausgabe von MAC- und IP-Adress-Informationen steuern.

Die Datei hat den folgenden Inhalt:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
   <appSettings>
      <add key="IncludeIPAddressFields" value="false" />
      <add key="IncludeMacAddressFields" value="false" />
   </appSettings>
</configuration>
```

Um die Ausgabe von IP- oder MAC-Adressen zu aktivieren, muss der entsprechende Wert auf "true" geändert werden.

4.2.3 Microsoft Endpoint Configuration Manager (MECM) ehemals System Center Configuration Manager (SCCM)

Element	Beschreibung
Unterstützte Versionen	System Center - Configuration Manager SCCM 2007 bis 2012 R2, Build 1511 bis Build 1810.2 Microsoft Endpoint Configuration Manager MECM 1902 bis 2107
SQL Datenbankrolle	db_datareader
Typ in Connector.config	t="SCCM"
Kommandozeile (überholt)	/t:SCCM
Abgefragte Tabellen	[dbo].[SetupInfo] [dbo].[v_GS_ADD_REMOVE_PROGRAMS] [dbo].[v_GS_ADD_REMOVE_PROGRAMS_64] [dbo].[v_GS_COMPUTER_SYSTEM] [dbo].[v_GS_INSTALLED_SOFTWARE] [dbo].[v_GS_INSTALLED_SOFTWARE_MS] [dbo].[v_GS_LOGICAL_Disk] [dbo].[v_GS_Operating_System] [dbo].[v_GS_PC_BIOS] [dbo].[v_GS_PROCESSOR] [dbo].[v_GS_SoftwareFile] [dbo].[v_GS_SoftwareProduct] [dbo].[v_GS_SYSTEM_ENCLOSURE] [dbo].[v_GS_VIDEO_CONTROLLER] [dbo].[v_GS_WORKSTATION_STATUS] [dbo].[v_GS_X86_PC_MEMORY] [dbo].[v_LU_MSProd] [dbo].[v_R_System] [dbo].[v_R_System_Valid] [dbo].[v_RA_System_IPAddresses]

Notiz Der Scan nach .exe Dateien durch Configuration Manager ist empfohlen, damit steigt die Erkennungsrate der Recognition.

Metering

Meteringdaten können aus Configuration Manager exportiert und in Spider verarbeitet werden. Beim Configuration Manager werden nur eingestellte Dateien gesammelt.

Element	Beschreibung
Unterstützte Versionen	System Center - Configuration Manager SCCM 2012 bis 2012 R2, Build 1511 bis Build 1810.2 Microsoft Endpoint Configuration Manager MECM 1902 bis 2107
SQL Datenbankrolle	db_datareader
Attribut in Connector.config	m="True" f="True"
Abgefragte Tabellen	[dbo].[v_MeterData] [dbo].[v_Users] [dbo].[v_ProductFileInfo]

SQL Server Editionserkennung

Damit per Configuration Manager zusätzliche Informationen zu den SQL Server Editionen ermittelt werden können, ist es nötig Configuration Manager so zu erweitern das bestimmte WMI Namespaces auf den Maschinen ausgewertet werden.

Die Configuration Manager Administratorkonsole erlaubt es nur einen Namenspace zu konfigurieren, allerdings gibt es mehrere Namespaces, die Informationen zu den installierten SQL Server Editionen enthalten.

Zurzeit sind diese Namespaces bekannt:

- \root\Microsoft\SqlServer\ComputerManagement
- \root\Microsoft\SqlServer\ComputerManagement10
- \root\Microsoft\SqlServer\ComputerManagement11
- \root\Microsoft\SqlServer\ComputerManagement12
- \root\Microsoft\SqlServer\ComputerManagement13
- \root\Microsoft\SqlServer\ComputerManagement14

Damit die zusätzlichen Namespaces abgefragt werden können, ist es nötig die MOF Dateien vom Configuration Manager zu erweitern. Eine detaillierte Beschreibung (inkl. Beispielen) ist hier zu finden: https://sccm-zone.com/sql-version-detection-and-report-sccm-2012-r2-12f299b5e63b

Achtung	Die oben genannte Website ist kein von Flexera betriebenes Angebot, die enthaltenen Informationen können sich jederzeit ändern, bzw. auch komplett verschwinden. Zur Konfiguration Ihres Configuration Manager Systems, fragen Sie bitte Ihren Configuration Manager Spezialisten.

Das Ergebnis der zusätzlichen Erkennung wird in zusätzlichen Tabellen im Configuration Manager abgelegt. Da diese Tabellen nicht zwingend auf allen Systemen den gleichen Namen haben müssen, müssen diese in einem speziellen View zusammengestellt werden damit der Data Collector sie abfragen kann.

Der Name des Views der (nur wenn er existiert) abgefragt wird, ist:

[dbo].[CUSTOM_SQLSERVEREDITION]

und muss die folgenden Spalten enthalten

- [MachineID] [int] NOT NULL
- [InstanceKey] [int] NOT NULL
- [TimeKey] [datetime] NOT NULL
- [RevisionID] [int] NOT NULL
- [AgentID] [int] NULL
- [rowversion] [timestamp] NOT NULL
- [IsReadOnly00] [int] NULL
- [PropertyIndex00] [int] NULL
- [PropertyName00] [nvarchar](255) NULL
- [PropertyNumValue00] [int] NULL
- [PropertyStrValue00] [nvarchar](255) NULL
- [PropertyValueType00] [int] NULL
- [ServiceName00] [nvarchar](255) NULL
- [SqlServiceType00] [int] NULL

Ein Beispiel wie der View erstellt werden kann:

```
CREATE VIEW [dbo].[CUSTOM_SQLSERVEREDITION] As
Select * FROM [CM_BWG].[dbo].[CUSTOM_SQL_2016_Property_2_0_DATA]
UNION ALL
Select * FROM [CM_BWG].[dbo].[CUSTOM_SQL_2014_Property_2_0_DATA]
UNION ALL
Select * FROM [CM_BWG].[dbo].[CUSTOM_SQL_2012_Property_2_0_DATA]
UNION ALL
Select * FROM [CM_BWG].[dbo].[CUSTOM_SQL_2008_Property_2_0_DATA]
UNION ALL
Select * FROM [CM_BWG].[dbo].[CUSTOM_SQL_Legacy_Property_2_0_DATA]
```

Achtung	Weder die Namen noch der Tabelle, noch die Spalten müssen in einer Configuration Manager Umgebung existieren, weitere Informationen um die nötigen Inventarinformationen zu erheben, können Sie bei Ihrem Configuration Manager Spezialisten erhalten.

4.2.4 Spider Data Center Inventory

Das Spider Data Center Inventory ermittelt die Daten von der Appliance.

Systemvoraussetzungen

Keine

Konfiguration

Connector.config Attribut	Beschreibung
server="<IP>"	IP-Adresse unter der die Appliance erreichbar ist.
share="<Share name>"	Name der Freigabe auf der Appliance
uid="<Benutzer>"	Benutzer mit dem sich auf die Freigabe verbunden werden kann.
pwd="<Passwort>"	Passwort für den o.g. Benutzer. Die Verschlüsselung nach den Standards für die PowerShell Konnektoren ist möglich, Details dazu sind hier zu finden: Passwortverschlüsselung bei PowerShell Konnektoren (siehe Seite 56)

Beispiel

Export mit Benutzer und Passwort

```
<connector name="Spider Data Center Inventory" subfolder="DatacenterInventory" active="true"
scriptname="GetDatacenterInventory.ps1" uid="<UserID>" pwd="<Passwort> server="<Server IP>" share="<Sharename>" sfx="" />
```

Export mit Benutzer, Passwort wird verschlüsselt in einer externen Datei abgelegt.

```
<connector name="Spider Data Center Inventory" subfolder="DatacenterInventory" active="true"
scriptname="GetDatacenterInventory.ps1" uid="<UserID>" server="<Server IP>" share="<Sharename>" sfx="" />
```

4.2.5 Heat Discovery

Element	Beschreibung
Unterstützte Versionen	2014.2 und höher
SQL Datenbankrolle	db_datareader
Typ in Connector.config	t="HEAT"
Kommandozeile (überholt)	/t:HEAT
Abgefragte Tabellen	[dbo].[CI] [dbo].[FRS_CIComponent] [dbo].[FRS_IM_FileInstance] [dbo].[FRS_IM_SoftwareFile] [dbo].[SoftwareIdentity] [dbo].[SoftwareType]

4.2.6 Frontrange Discovery

Element	Beschreibung
Unterstützte Versionen	9.3 und höher
SQL Datenbankrolle	db_datareader
Typ in Connector.config	t="DISC"
Kommandozeile (überholt)	/t:DISC
Abgefragte Tabellen	[dbo].[ClientType] [dbo].[Client] [dbo].[defOSType] [dbo].[defOS] [dbo].[Hardware] [dbo].[Manufacturer] [dbo].[OperatingSystem] [dbo].[Products] [dbo].[SoftwareAud] [dbo].[SoftwareFile] [dbo].[SoftwarePackage] [dbo].[SoftwarePath] [dbo].[SRDFile] [dbo].[SRDVersion] [dbo].[System]

4.2.7 Landesk

Element	Beschreibung
Unterstützte Versionen	9.x
SQL Datenbankrolle	db_datareader
Typ in Connector.config	t="LANDESK"
Kommandozeile (überholt)	/t:LANDESK
Abgefragte Tabellen	[dbo].[AppSoftwareSuites] [dbo].[AppSoftware] [dbo].[BIOS] [dbo].[BoundAdapter] [dbo].[CompSystem] [dbo].[Computer] [dbo].[FileInfo] [dbo].[LogicalDrives] [dbo].[Memory] [dbo].[NetworkSoftware] [dbo].[Operating_System] [dbo].[Processor] [dbo].[VideoAdapter]

4.2.8 Lansweeper

Element	Beschreibung
Unterstützte Versionen	x
SQL Datenbankrolle	db_datareader
Typ in Connector.config	t="Lansweeper"
Kommandozeile (überholt)	/t:Lansweeper
Abgefragte Tabellen	[dbo].[tblAssetCustom] [dbo].[tblAssets] [dbo].[tblBIOS] [dbo].[tblComputersystem] [dbo].[tblComputerSystemProduct] [dbo].[tblDiskdrives] [dbo].[tblOperatingsystem] [dbo].[tblProcessor] [dbo].[tblSoftware] [dbo].[tblSoftwareUni] [dbo].[tblSystemEnclosure] [dbo].[tblVideoController] [dbo].[tblSqlServers]

4.2.9 Altiris 7

Item	Description
Unterstützte Versionen	7.5

Item	Description
SQL Datenbankrolle	db_datareader
Typ in Connector.config	t="ALT7"
Abgefragte Tabellen	[dbo].[Inv_AddRemoveProgram] [dbo].[vComputer] [dbo].[vHWBaseboard] [dbo].[vHWChassis] [dbo].[vHWComputerSystem] [dbo].[vHWDisplayController] [dbo].[vHWProcessor] [dbo].[vSWBIOSElement]

4.2.10 Generischer Konnektor

Der generische Konnektor erwartet einen Satz von Stored Procedures mit einem bestimmten Namen in der abzufragenden Datenbank.

Element	Beschreibung
SQL Datenbankrolle	db_datareader auf den verwendeten Tabellen Execute Rechte an den Stored Procedures
Typ in Connector.config	t="Generic"
Kommandozeile (überholt)	/t:Generic
Stored Procedures	[dbo].[swrGetWorkList] [dbo].[swrGetHardwareScan] [dbo].[swrGetFileScan] [dbo].[swrGetSoftwareScan] [dbo].[swrGetDeviceRelationship] [dbo].[swrGetADUserObject] [dbo].[swrGetADGroupObject] [dbo].[swrGetADGroupMember] [dbo].[swrGetSwidScan]

Achtung Es ist sicherzustellen das der Benutzer der die Verbindung auf die Datenbank aufbaut, EXECUTE Rechte an den Stored Procedures hat.

Ein Satz von Beispielprozeduren kann unter: https://docs.flexera.com/Spider64/GenericDataConnectorTemplates.zip heruntergeladen werden.

All diese Stored Procedures haben verpflichtende und Optionale Spalten, die Information dazu, sowie welchen Datentyp die jeweilige Spalte haben muss ist hier zu finden: Generic Connector Stored Procedures (siehe Seite 140)

4.2.11 Microsoft Assessment and Planning Toolkit (MAP)

Element	Beschreibung
Unterstützte Versionen	8.5 9.0 9.1 9.2 9.4 9.7
SQL Datenbankrolle	db_datareader
Typ in Connector.config	t="MAP"
Kommandozeile (überholt)	/t:MAP
Abgefragte Tabellen	[AllDevices_Assessment].[CategorizedDevices] [AllDevices_Assessment].[HardwareInventoryCore] [AllDevices_Assessment].[HardwareInventoryEx] [Core_Inventory].[Devices] [SqlServer_Assessment].[SqlInstances] [UT_Exchange_Inventory].[AdServers] [Win_Assessment].[VirtualMachinesView] [Win_Assessment].[WindowsInstalledSoftwareFull] [Win_Inventory].[ComputerSystemProduct] [Win_Inventory].[DataFile] [Win_Inventory].[LogicalDisks] [Win_Inventory].[Processors] [Win_Inventory].[VideoControllers] Für die zusätzliche Erkennung muss der Benutzer das EXECUTE Recht an den folgenden Funktionen erhalten: [UT_Exchange_Reporting].[GetExchangeEditionFromTypeValue] [UT_Exchange_Reporting].[GetExchangeProductNameFromVersionNumber] [UT_SCCM_Reporting].[GetSccmProductName]

Das Microsoft Assessment and Planning Tool (MAP) kann mit zwei Arten von Datenbanksystemen betrieben werden.

Die erste ist eine mit dem Produkt gebündelte applikationsspezifische SQL Server Version, bekannt unter dem Namen LocalDB. Der Vorteil von LocalDB ist der geringe Aufwand der benötigt wird um eine Installation von MAP zu installieren. Alles ist mit einem Setup zu implementieren und abhängig von der Geschwindigkeit der Maschine kann die Installation in unter fünf Minuten abgeschlossen sein.

Dies Bedeutet allerdings auch, das der Data Collector Dienst mit dem Benutzerkonto laufen muss, unter dem die MAP Software (mit LocalDB) installiert wurde, da die Datenbank im Benutzerverzeichnis der Maschine abgelegt wird. Zusätzlich muss eine Erkennung der LocalDB Konfiguration ermöglicht werden, da die Datenbank nur bei der Benutzung aktiviert wird und jedes Mal eine andere Pipe Adresse erhält.

Die zweite von Flexera empfohlene Methode ist die Installation von MAP mit einer SQL (Express) Server Edition (auf der gleichen Maschine auf der MAP installiert wird), die folgenden Einstellungen müssen während der Installation des SQL Servers berücksichtigt werden:

Instanzname:	MAPS
SQL Server Kollation:	SQL_Latin1_General_CP1_CI_AS

Ein mit den genannten Einstellungen installierter SQL Server, wird durch das MAP Setup erkannt und verwendet. Die Verwendung eines vollwertigen SQL Servers erlaubt den Betrieb unter einem Windows Dienst, und, unter anderem, den Zugriff unterschiedlicher Benutzer auf die Datenbank.

Falls der Benutzer der den Export durchführt, kein Administrator auf der Datenbank ist, müssen dem abfragenden Benutzer EXECUTE Rechte für die folgenden Funktionen gewährt werden:

- Funktionen
  - [UT_Exchange_Reporting].[GetExchangeEditionFromTypeValue]
  - [UT_Exchange_Reporting].[GetExchangeProductNameFromVersionNumber]
  - [UT_SCCM_Reporting].[GetSccmProductName]

Wichtig	Die oben genannte Konfiguration des SQL Server Instanz sollte vor der Ausführung des MAP Setups abgeschlossen sein. Eine Nachträgliche Einrichtung des SQL Servers führt dazu, dass das MAP Setup keine gültige Instanz vorfindet und die Installation in einer LocalDB ausführt.

4.2.12 Matrix42 (Beta)

Element	Beschreibung
Unterstützte Versionen	NV
SQL Datenbankrolle	db_datareader
Typ in Connector.config	t="MATR"
Abgefragte Tabellen	[dbo].[SPSApplicationClassBase] [dbo].[SPSAssetClassBase] [dbo].[SPSAssetPickupType] [dbo].[SPSComputerClassBase] [dbo].[SPSComputerClassGraphicCard] [dbo].[SPSComputerClassGraphicCard] [dbo].[SPSComputerClassHardDisk] [dbo].[SPSComputerClassOS] [dbo].[SPSInventoryClassApplication] [dbo].[SPSStockKeepingUnitClassBase] [dbo].[SPSSupplierClassBase] [dbo].[SPSUserClassbase]

4.2.13 Empirum Workplace Management (Beta)

Element	Beschreibung
Unterstützte Versionen	15.2
SQL Datenbankrolle	db_datareader
Typ in Connector.config	t="EMPI"
Abgefragte Tabellen	[dbo].[DMISystem] [dbo].[InvComputer] [dbo].[InvFiles] [dbo].[InvSoftware] [dbo].[WMIBattery] [dbo].[WMIProcessor]

4.2.14 Baramundi (Beta)

Element	Beschreibung
Unterstützte Versionen	NV
SQL Datenbankrolle	db_datareader
Typ in Connector.config	t="BARA"
Abgefragte Tabellen	[dbo].[machine] [dbo].[inventory_nodes] [dbo].[inventory_nodeproperties] [dbo].[InventoriedSoftware] [dbo].[SwDetectionRule]

4.2.15 Snow (Beta)

Item	Description
Unterstützte Versionen	NV
SQL Datenbankrolle	db_datareader
Typ in Connector.config	t="SNOW"
Abgefragte Tabellen	[dbo].[vClient] [dbo].[vDisplayAdapter] [dbo].[vLogicalDisks] [dbo].[vMemory] [dbo].[vNetworkAdapter] [dbo].[vOperatingSystem] [dbo].[vProcessor] [dbo].[vSoftware]

4.2.16 Übersicht der ermittelten Daten

Die Exporte und Inventory Tools geben eine unterschiedliche Anzahl an Inventarinformationen zurück.

Eine Übersicht, welches Tool, welche Daten liefert ist in den folgenden Tabellen zu finden

Hersteller	Columbus Inventory		Microsoft			Ivanti		Lansweeper	Landesk
Produkt	Scanner	Agent	MAP	SCCM	SCCM 2012	Discovery	Heat		
DomainName	•	•	• (1)	• *1	• (1)	• (1)	• (1)	- (1)	• (1)
BIOSDate	•	•	•	•	•	•	•	•	•
BIOSVendor	•	•	•	•	•	•	•	•	•
BIOSVersion	•	•	•	•	•	•	•	•	•
ChassisType	-	-	-	•	•	-	•	-	-
CorePerCPU	-	-	•	-	•	•	-	•	•
CPUArchitecture	•	•	•	•	•	-	•	•	•
CPUCoreCount	•	•	•	-	•	-	•	-	•
CPUCount	•	•	•	•	•	•	•	•	•
CPULogicalCount	•	•	•	-	•	-	-	•	-
DeviceChassis	•	•	•	-	-	•	-	•	•
DiskFreeMB	•	•	•	•	•	•	•	•	•
DiskTotalMB	•	•	•	•	•	•	•	•	•
DomaineNameNetBios	•	•	- (1)	• (1)	• (1)	- (1)	- (1)	• (1)	- (1)
GraphicAdapter	•	•	-	•	•	•	•	•	•
GraphicMemoryMB	•	•	-	•	•	•	•	•	-
HostName	•	•	• (1)	• (1)	• (1)	• (1)	• (1)	• (1)	• (1)
IPAddressV4	•	•	•	-	•	•	•	•	•
IPAddressV6	•	•	-	-	•	-	-	-	-
LastLoggedOnUser	•	•	•	•	•	•	•	•	•
MAC1	•	•	-	-	-	•	•	•	•

Hersteller	Columbus Inventory		Microsoft			Ivanti		Lansweeper	Landesk
MAC2	•	•	-	-	-	•	-	-	-
MAC3	•	•	-	-	-	•	-	-	-
MAC4	•	•	-	-	-	-	-	-	-
Manufacturer	•	•	•	•	•	•	•	•	•
MemoryMB	•	•	•	•	•	•	•	•	
Model	•	•	•	•	•	•	•	•	•
OSCaption	•	•	•	•	•	•	•	•	•
OSClass	•	•	-	-	-	•	•	•	•
ProcessorManufacturer	•	•	•	•	•	•	•	•	•
ProcessorSpeed	•	•	•	•	•	•	•	•	•
ProcessorType	•	•	•	•	•	•	•	•	•
ScanDate	•	•	•	•	•	•	•	•	•
Serial	•	•	•	•	•	•	•	•	•
UUID	•	•	• (1)	• (1)	• (1)	• (1)	• (1)	• (1)	- (1)
Worklist									
Identifier	-	-	•	•	•	•	•	•	•
UUID	•	•	•	•	•	•	•	•	-
URN	-	-	-	-	-	-	-	-	-
Hostname	•	•	•	•	•	•	•	•	•
DomainName	•	•	•	•	•	•	•	-	•
DomainNameNetBios	•	•	-	•	•	-	-	•	-
Software Scan	•	•	•	•	•	•	•	•	•
SQL Server Edition	•	•		•	•	-	-	•	-
Autodesk	•	•		-	-	-	-	-	-
Embedded OS	•	•		-	-	-	-	-	-
File Scan	•	•	-	•	•	•	•	-	•

Hersteller	Columbus Inventory		Microsoft			Ivanti		Lansweeper	Landesk
Metering	-	•	-	•	•	-	-	-	-
SWID Tags	-	-	•	•	•	-	-	-	-

Beta Konnektoren

Hersteller	Snow	Matrix42		Baramundi
Produkt	Snow Inventory	Matrix42	Empirum	Baramundi
DomainName	• (1)	• (1)	• (1)	• (1)
BIOSDate	•	•	•	•
BIOSVendor	•	•	-	•
BIOSVersion	•	•	•	•
ChassisType	-	-	-	-
CorePerCPU	•	-	-	•
CPUArchitecture	-	-	•	-
CPUCoreCount	•	-	•	•
CPUCount	•	•	•	•
CPULogicalCount	•	•	•	•
DeviceChassis	•	•	•	•
DiskFreeMB	•	-	-	•
DiskTotalMB	•	•	-	•
DomaineNameNetBios	• (1)	• (1)	• (1)	-
GraphicAdapter	•	•	•	-
GraphicMemoryMB	-	•	-	-
HostName	• (1)	• (1)	• (1)	• (1)
IPAddressV4	•	•	-	•
IPAddressV6	-	-	-	-
LastLoggedOnUser	•	-	•	•

MAC1	•	•	•	•
MAC2	-	-	´-	-
MAC3	-	-	-	-
MAC4	-	-	-	-
Manufacturer	•	•	•	•
MemoryMB	•	•	•	•
Model	•	•	•	•
OSCaption	•	•	•	•
ProcessorManufacturer	•	-	•	•
ProcessorSpeed	•	•	•	•
ProcessorType	•	•	•	•
ScanDate	•	•	•	•
Serial	•	•	•	•
UUID	• (1)	• (1)	• (1)	-
Worklist				
Identifier	•	•	•	• (1)
UUID	-	-	•	-
URN	-	-	-	-
Hostname	•	•	•	• (1)
DomainName	•	•	•	• (1)
DomainNameNetBios	-	•	•	-
Software Scan	•	•	•	•
SQL Server Edition	-	-	-	-
Autodesk	-	-	-	-
Embedded OS	-	-	-	-
File Scan	•	-	•	-
Metering	-	-	-	-
SWID Tags	-	-	-	-

(1) - Daten werden aus der Worklist übernommen

4.3 API basierende Konnektoren

4.3.1 Einführung

Verwendung eines Proxy Servers

Einige Konnektoren müssen auf das Internet zugreifen um Informationen einzusammeln, dies sind die folgenden Konnektoren:

- Microsoft Azure
- Adobe Online

Damit diese Konnektoren das Internet über einen Proxy erreichen können, können diese zusätzlichen Einträge in der Connector.config gemacht werden.

Konfiguration

Attribut	Pflichtangabe	Beschreibung
proxyAddress=<Adresse des Proxy Servers>	Nein	Adresse des Proxy Servers inklusive http(s)://
proxyPort =<Port des Proxy Servers>	Nein	Port auf dem der Proxy Server Anfragen entgegen nimmt.
proxyUser=<Benutzername>	Nein	Falls nötig, Benutzer für den Zugriff.
proxyUserPassword=<Passwort>	Nein	Passwort für den o.g. Benutzer

Notiz: Das Passwort für den Benutzer kann mittels der Standardverschlüsselung für PowerShell Konnektoren verschlüsselt werden, Details dazu finden sie hier: Passwortverschlüsselung bei PowerShell Konnektoren (siehe Seite 56)

Fehlersuche bei der PowerShell Konnektor Ausführung

Es kann vorkommen das nach der Durchsicht der Logdateien ein einzelner Konnektor ausgeführt werden muss um dem Fehler auf den Grund zu gehen.

Die beste Möglichkeit dazu ist, den Konnektor außerhalt des Management Skriptes (ExecuteDCs.ps1) auszuführen.

Um einen beliebigen Konnektor manuell auszuführen sollte eine Batch Datei (z.B. <ConnectorName>.cmd) im Verzeichnis des Konnektors erstellt werden. Der Inhalt der Batch Datei sollte wie folgt aussehen:

```
PowerShell -File .\<Name des the Konnektor PowerShell Scripts>.ps1 -dir "<Ausgabeverzeichnis der Ergebnisse>" <Zusätzliche Parameter>
Pause
```

Danach wird die Batch Datei ausgeführt und das Konsolenfenster nach evtl. Fehlern überprüft.

<Zusätzliche Parameter> können direkt aus der Connector.config entnommen werden, zum Beispiel so:

Zeile in der Connector.config

```
<connector name="GetADGroupObjects" subfolder="ADConnector" active="true" scriptname="GetADGroupObjects.ps1"
dc="dc01.domain.ocal" sfx="GroupObjects" grp="ManagementGroup,Developers,Sales" strict="false" queryParentGroup="true" />
```

Die Attribute "name", "subfolder", "active", "scriptname" können ignoriert werden, alle anderen (falls sie verwendet werden, es müssen nicht immer alle Attribute Verwendung finden), können nach folgendem Muster übernommen werden.

Jedes Attribut bekommt ein "-" vorangestellt, und das "=" wird durch ein Leerzeichen ersetzt, im Beispiel sieht das dann so aus:

```
-dc "dc01.domain.ocal" -sfx "GroupObjects" -grp "ManagementGroup,Developers,Sales"
```

Attribute die einen "True|False" Wert enthalten, wird wie folgt vorgegangen, bei "True" wird dem Attribut ein "-" vorangestellt und der Rest ab dem "=" wird gelöscht. Attribute mit "False" als Wert, werden weggelassen, aus "strict="false" queryParentGroup="true"" wird: "-queryParentGroup"

Der vollständige Text für die Zeile die mit PowerShell beginnt, sieht dann so aus:

```
PowerShell -File .\GetADGroupObjects.ps1 -dir "<Output directory for the result>" -dc "dc01.domain.ocal" -sfx "GroupObjects" -grp
"ManagementGroup,Developers,Sales" -queryParentGroup
```

Achtung:	Für <Ausgabeverzeichnis der Ergebnisse> ist es am besten wenn das Verzeichnis des Konnektors gewählt wird, damit findet die Ausgabe im Verzeichnis statt aus dem die Verarbeitung gestartet wird und die Navigation zu einem anderen Verzeichnis wird eingespart.

PSRemoting - Befehle auf entfernten Computern ausführen

Einige Konnektoren verwenden sogenannte Remote Sessions in PowerShell um Befehle auf den Maschinen auszuführen auf den die Daten vorhanden sind, ggfs. sind dort PowerShell Module vorhanden die nicht auf den Server auf dem der SDC läuft installiert werden können.

Details zum Aktivieren und zur Fehlerbehebung, findet man bei Microsoft unter Enable-PSRemoting

Ein einfaches Beispiel um nach der Aktivierung von PSRemoting die Verbindung zu testen, sieht so aus:

```
$computer = "<Remote server>"
$username = "<User to connect to remote server>"
```

```
$password = "<Password for above user>"

$psSessionSplat = @{
   computerName = $computer
}

if($username -and $password) {
   $pw = convertto-securestring -AsPlainText -Force -String $password
   $cred = new-object -typename System.Management.Automation.PSCredential -argumentlist $username,$pw
   $psSessionSplat.Credential = $cred
}

Write-Host @psSessionSplat -ForeGroundColor Yellow
$session = new-pssession @psSessionSplat
Get-PSSession
Remove-PSSession $session
```

PowerShell Execution Policy

Die Windows PowerShell Execution Policies bestimmen unter welchen Bedingungen PowerShell Skripte lädt und ausführt.

In manchen Fällen muss die Policy angepasst werden. Um zu bestimmen welche Policy für die jeweilige Umgebung am passendsten ist, wird der folgende Artikel empfohlen: About Execution Policies

Die Skripte des Data Collectors sind mit der Einstellung "RemoteSigned" getestet.

Die Aktuelle Einstellung der Execution Policy kann mit dem folgenden Befehl überprüft werden:

```
Get-ExecutionPolicy -List | fl
```

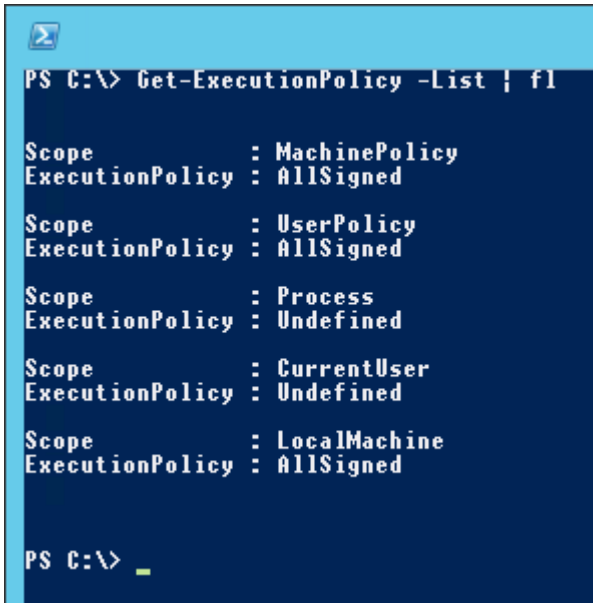


Abbildung - ExecutionPolicy

Um die Execution Policy zu ändern, kann der folgende Befehl verwendet werden:

```
Set-ExecutionPolicy RemoteSigned
```

4.3.2 VMware vCenter / ESX Server

Systemvoraussetzungen

Damit der vCenter Konnektor funktionieren kann, ist es nötig die von VMware zur Verfügung gestellten PowerShell PowerCLI Tools zu installieren.

Achtung	Ab Version 6.5.1 der PowerCLI, veröffentlicht VMware keine installierbare MSI Datei mehr, sondern stellt die Software auf der PowerShell Gallery zur Verfügung. Details zur Installation werden auf der folgenden Seite beschrieben: https://blogs.vmware.com/PowerCLI/2017/04/powercli-install-process-powershell-gallery.html Es ist empfohlen die PowerCLI aktuell zu halten und ggf. regelmässig auf Aktualität zu prüfen.

Damit auch der Servicebenutzer die installierte PowerCLI nutzen kann, muss diese für "Alle Benutzer" der Maschine installiert werden.

```
Install-Module -Name VMware.PowerCLI –Scope Allusers
```

In manchen Fällen ist es möglich, das andere PowerShell Module den Namen einer Methode verwenden die auch die PowerCLI verwendet. Dann ist es nötig der PowerCLI Installation zu erlauben den/die anderen Namen zu überschreiben. Dies geschieht über der -AllowClober Parameter bei der Installation.

```
Install-Module -Name VMware.PowerCLI –Scope Allusers -AllowClobber
```

Achtung	Bei nicht korrekten Zertifikaten, bzw. ungültigen Zertifikatsketten, kann die Warnung die die Ausführung des Konnektors verhindert mit: **Set-PowerCLIConfiguration -InvalidCertificateAction "Ignore" -Scope AllUsers** ausgeschaltet werden.

Einstellungen

Dieser Konnektor ermittelt Hardware, Software und Beziehungsinformationen von entweder einem vCenter oder einem eigenständigen ESX Server.

Für eine verbesserte Unterstützung von VDI, erfolgt bei Client-Betriebssystemen (wie „Windows 10” ,„Windows 8” oder “Windows 7”) die Device Identifikation über Hostname/Domainname. In dem Fall wird keine “UUID” exportiert.

Konfiguration

Connector.config Attribut	Beschreibung
srv="<Server>"	Name des vCenter oder ESX Servers
port="<port>	Port des vCenter oder ESX Servers
uid="<Benutzer>"	Benutzer der für die Abfrage verwendet werden soll
pwd="<Passwort>"	Passwort für o.g. Benutzer
h="true"	Export der Hardwareinformationen
dr="true"	Export der Beziehungsinformationen

Connector.config Attribut	Beschreibung
s="true"	Export der Lizenzinformationen
SerialNumber="true"	Export der Seriennummer des ESX Hosts
NoGuest	Kein Guests Export
OnlyWindows	Nur Windows Guests exportieren

Beispiel

```
<connector name="DSDC vCenter Inventory" subfolder="vCenter" active="true" scriptname="GetvCenter-Details.ps1"
srv="vcenter.domain.com" -port="443" uid="domain\username" pwd="password" h="true" s="true" dr="true" onlyWindows=true
sfx="suffix" />
```

Attribute

Der vCenter Konnektor ermittelt die folgende Werte von einem vCenter oder ESX Server.

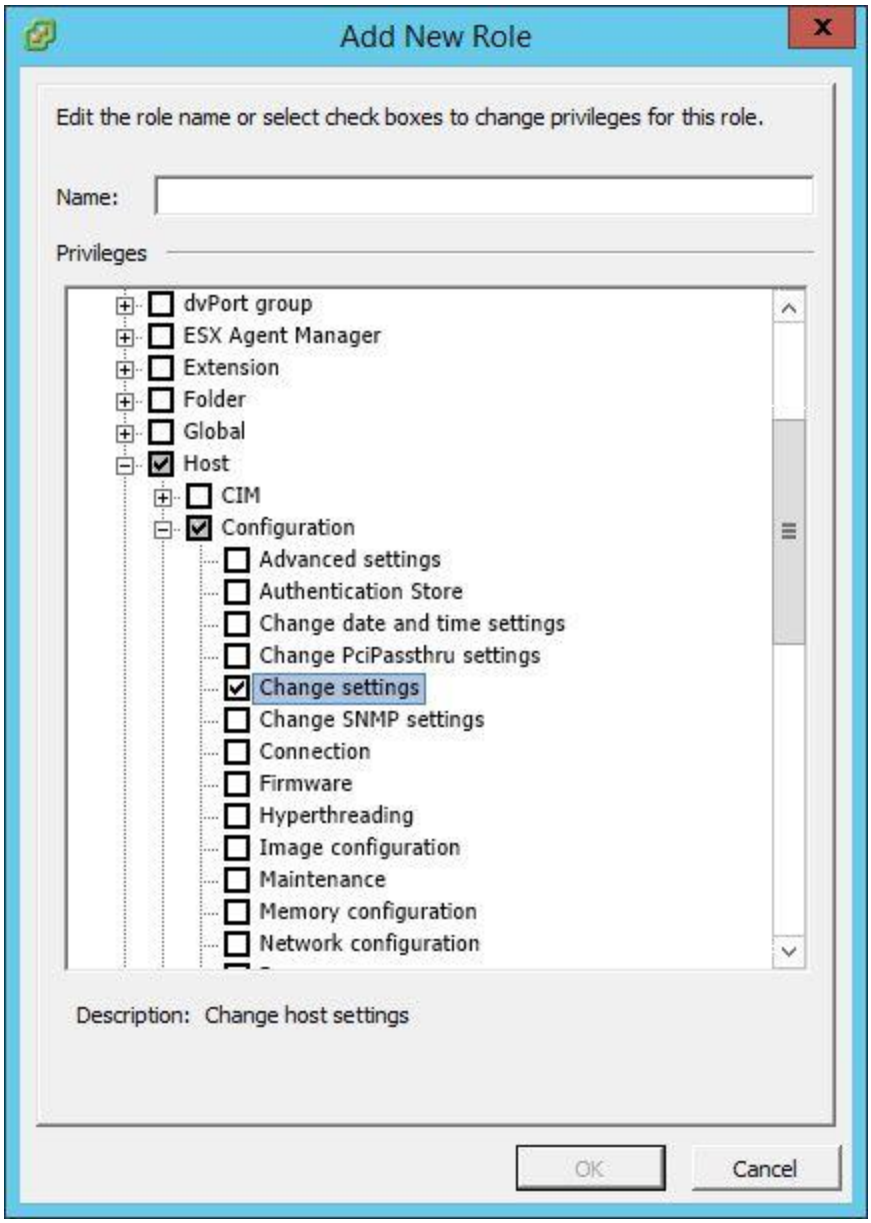
- Hardware (des ESX Hosts)
  - UUID
  - Urn
  - DomainName
  - HostName
  - DomainNetBIOS
  - Manufacturer
  - ScanDate
  - Model
  - Mac1
  - Mac2
  - Mac3
  - Mac4
  - ProcessorManufacturer
  - ProcessorType
  - ProcessorSpeed
  - CPUCount
  - CPUCoreCount
  - CorePerCPU
  - CPULogicalCount
  - DiskTotalMB
  - DiskFreeMB
  - MemoryMB
  - IPAddressV4
  - IPAddressV6
  - OSCaption
  - BIOSVersion
  - BIOSDate
  - InventorySource
  - Class
- Beziehungen

- DeviceRelationshipTypeID (1 für ESX - Guest Relation, 2 für Cluster - ESX Beziehung)
- ChildDeviceUUID
- ParentDeviceUUID
- ParentDeviceURN
- ScanDate

Export der Seriennummer der ESX Server

Die Ermittlung der Seriennummer des ESX Servers ist nur ab einer ESX Version von 5 oder höher möglich

Das Benutzerkonto mit dem die Informationen zu vCenter/ESX Server abgefragt werden, braucht zusätzliche Einstellungen um die Seriennummer abfragen zu dürfen. Dem Benutzer muss das Privileg **"Host.Configuration.Change Settings"** zugewiesen werden**.**



Der Export der Seriennummer wird nur durchgeführt, wenn in der Connector.config das Attribut **SerialNumber="True"** angegeben ist.

VCenter Diagnostic Tool

Das Powershell basierte Tool listet folgende Elemente inklusive ihrer Status auf

- Connected VM Hots
- Disconnecd VM Hosts
- VM Hosts in Maintenance
- Not responding VM Hosts
- running Guests
- Not Runing Guests
- "Powered ON" related VMs
- "Powered Off" related VMS
- Connected Clusters (Master /Child cluster/hosts)

Konfiguration

Connector.config Attribut	Beschreibung
srv="<Server>"	Name des vCenter oder ESX Servers
port="<port>	Port des vCenter oder ESX Servers
uid="<Benutzer>"	Benutzer der für die Abfrage verwendet werden soll
pwd="<Passwort>"	Passwort für o.g. Benutzer

Das Tool erfasst neben der PowerShell auch die Version der PowerCli des vCenters.

Für den Hardwarescan werden nur Connected Hosts und Running Guest exportiert.

Konnektor für Datacenter-Modul

Für die Funktionalität mit dem Datacenter-Modul ist dieser zusätzliche Konnektor zur Ermittlung der ESX/vCenter Informationen notwendig.

Der Konnektor besteht aus zwei separaten Skripten, eines zur Ermittlung der ESX/vCenter Daten, und das zweite zum Verpacken der Ergebnisse in das .CDC Format. Diese werden zur Verarbeitung an das Datacenter-Modul weitergeleitet.

Konfiguration - Ermitteln der ESX/vCenter Daten

Connector.config Attribut	Beschreibung
server="<Server>"	Name des vCenter oder ESX Servers
port="<port>	Port des vCenter oder ESX Servers
username="<Benutzer>"	Benutzer der für die Abfrage verwendet werden soll
password="<Passwort>"	Passwort für o.g. Benutzer

Konfiguration - Verpacken der Ergebnisse

Dieser Konnektor benötigt keine Parameter

Beispiel

```
<connector name="vCenter Inventory" subfolder="vCenter" active="true" scriptname="get_esxihosts_vm.ps1"
server="vcenter.domain.com" -port="443" username="domain\username"  password=password"  />
<connector name="vCenter Inventory novaratio CDC" subfolder="vCenter" active="true" scriptname="get_esxihosts_vm_to_CDC.ps1" />
```

Achtung:	Das Skript get_esxihosts_vm_to_CDC.ps1 MUSS immer NACH dem/n Konnektor(en) ausgeführt werden, eine Änderung der Reihenfolge würde verhindern das die Ergebnisse korrekt verpackt werden!

4.3.3 Adobe Online

Der Adobe Online Konnektor wird verwendet um Nutzerbasierende Lizensierungsdaten aus dem Adobe Online Portal zu erheben. Z.B. Photoshop CC, Illustrator CC, All Apps Plan...

Der Connector liegt in der Version zwei vor und wird im Falle eines Updates automatisch aktualisiert. Die alte Version wird als "Version legacy" weitergeführt um Umgebungen, die mit dem neuen Konnektor nicht kompatibel sind, weiterhin zu unterstützen. Das Update wurde nötig, da die angesprochene Adobe API veraltet ist.

Systemvoraussetzungen

Achtung	Damit der Konnektor funktionieren kann, wird auf bestimmten Systemen die Installation des Visual C++ 2010 Redistributable Package (x64) benötigt, dieses kann unter: Microsoft Visual C++ 2010 Redistributable Package (x64) heruntergeladen werden.

Achtung	Um eine Integration zu erstellen, wird eine Enterprise ID mit administrativen Privilegien benötigt. Details dazu sind unter: https://www.adobe.io/apis/cloudplatform/console/authentication/gettingstarted.html "Creating an Integration" -> "Service Account Authentication zu finden".

Konfiguration

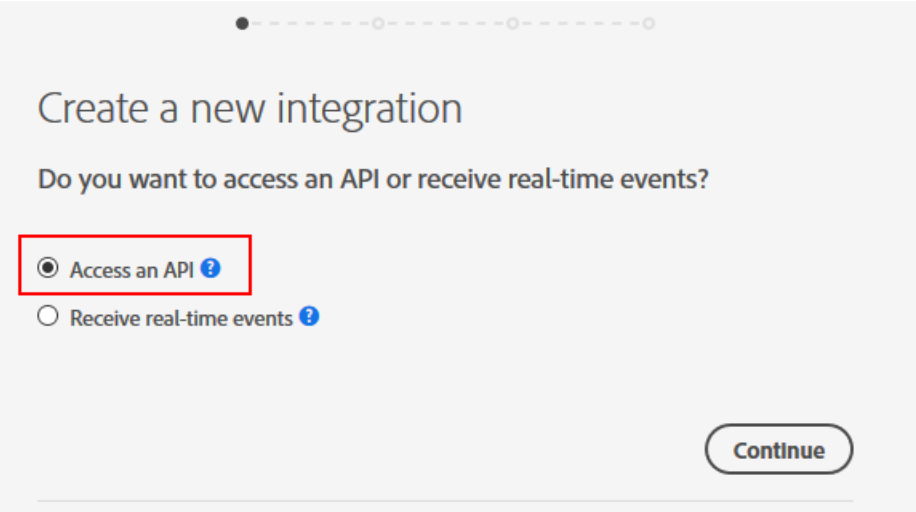
Connector.config Attribute	Pflichtangabe	Beschreibung
organizationID="<organizationID>"	Ja	organizationID aus der adobe.io Konsole
technicalAccountID="<technicalAccountID>"	Ja	technicalAccountID aus der adobe.io Konsole
apiKey="<apiKey>"	Ja	apiKey aus der adobe.io Konsole
clientSecret="<clientSecret>"	Ja	clientSecret aus der adobe.io Konsole
privateKeyName="<privateKeyName>"	Ja	Name der Datei die den privaten Schlüssel enthält
privateKeyPassword="<privateKeyPassword>"	Nein	Optional: Falls die Schlüsseldatei mit einem Passwort gesichert ist, muss dieses hier angegeben werden.
proxyAddress="<Proxy>"	Nein	Adresse des Proxy Servers (falls benötigt) um die Verbindung in das Internet zuzulassen. Proxy Adresse inklusive http(s)://
proxyPort="<Port>"	Nein	Port des Proxy Servers
proxyUser="<Benutzer>"	Nein	Benutzer der sich gegenüber dem Proxy Server authentifizieren kann.
proxyUserPassword="<Passwort>"	Nein	Passwort für o.g. Benutzer
TLS="<True|False>"	Nein	Verschlüsselte Übertragung aktivieren

Konfiguration der Integration

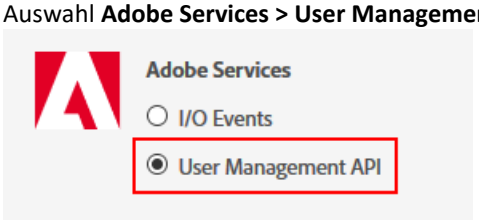
Damit der Konnektor sich verbinden kann, muss eine sog. Integration in der adobe.io Konsole eingerichtet werden.

1. Anmelden an https://console.adobe.io/integrations

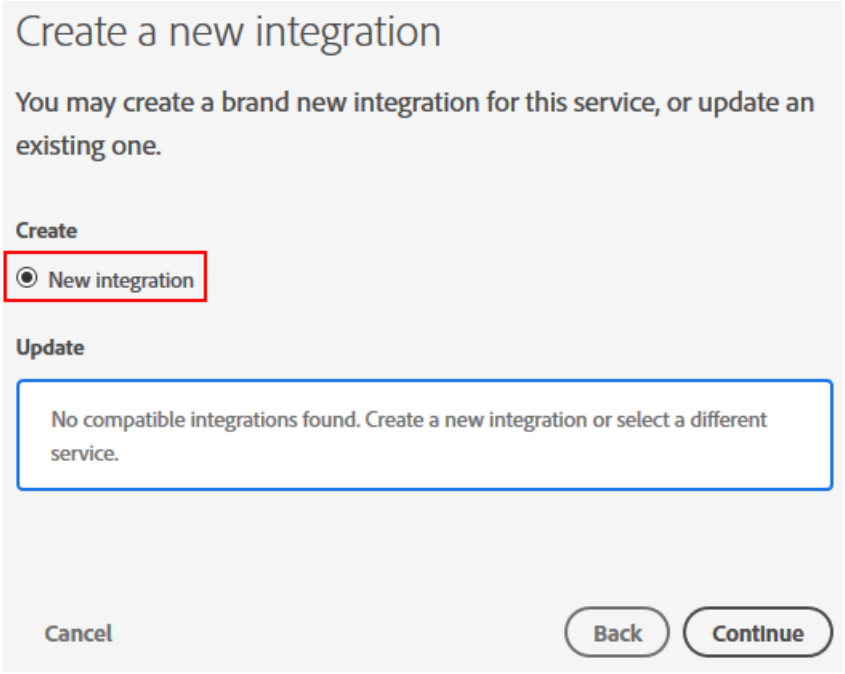
2. Klick auf "New Integration"
3. Auswahl **Access an API > Continue**



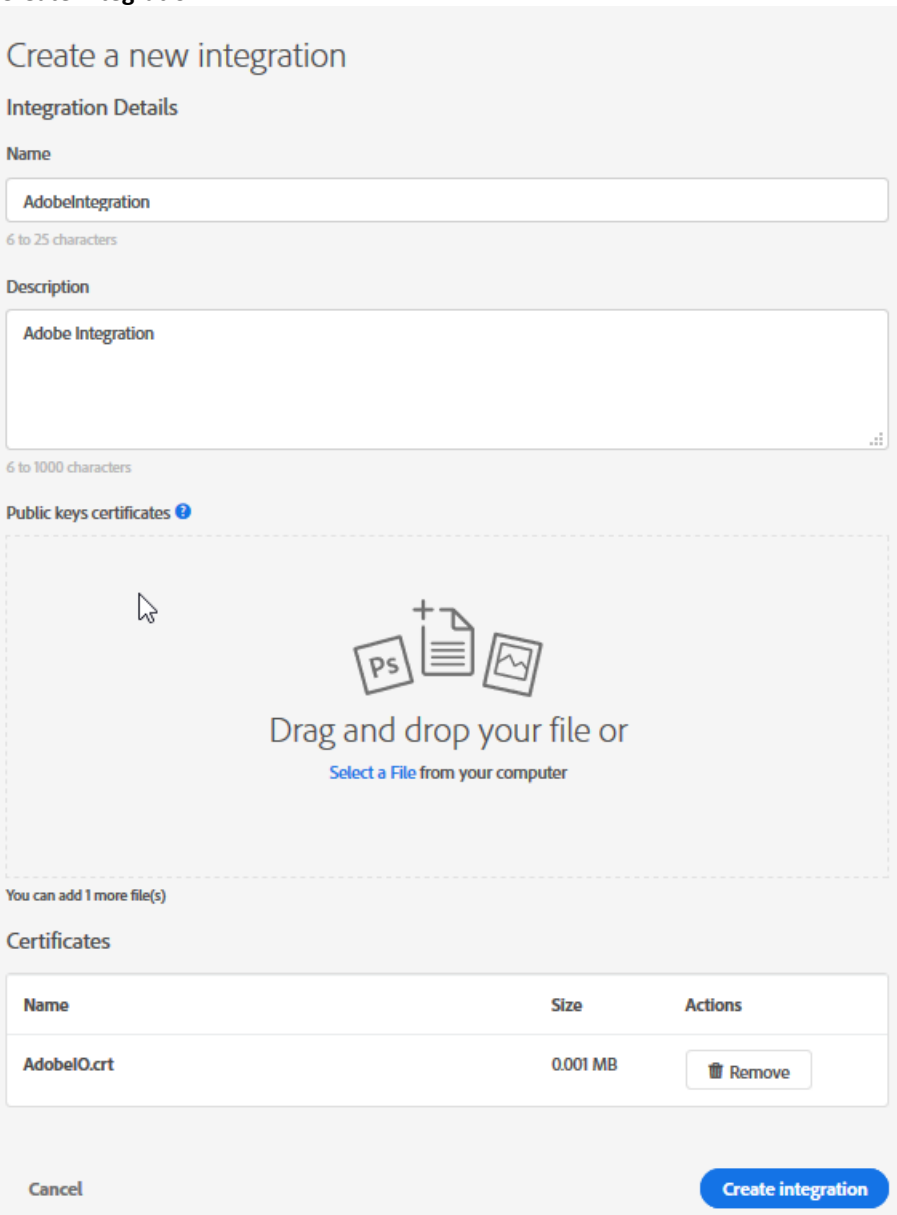
4. Auswahl **Adobe Services > User Management API > Continue**



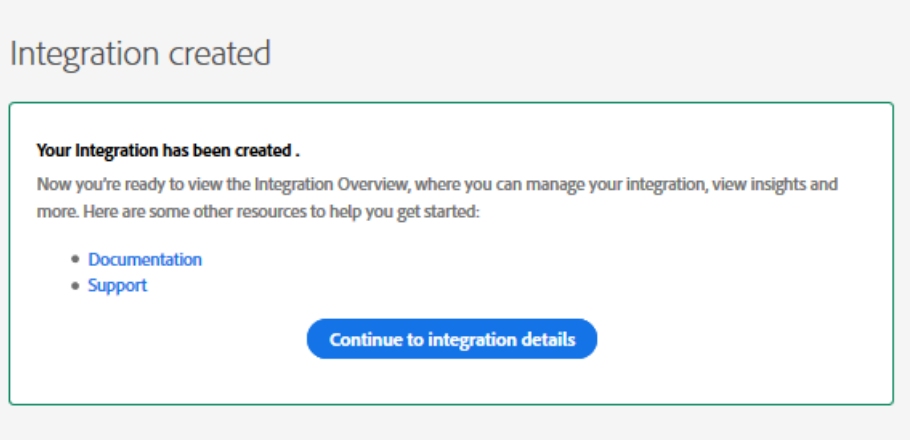
5. Auswahl **Create > New Integration > Continue**



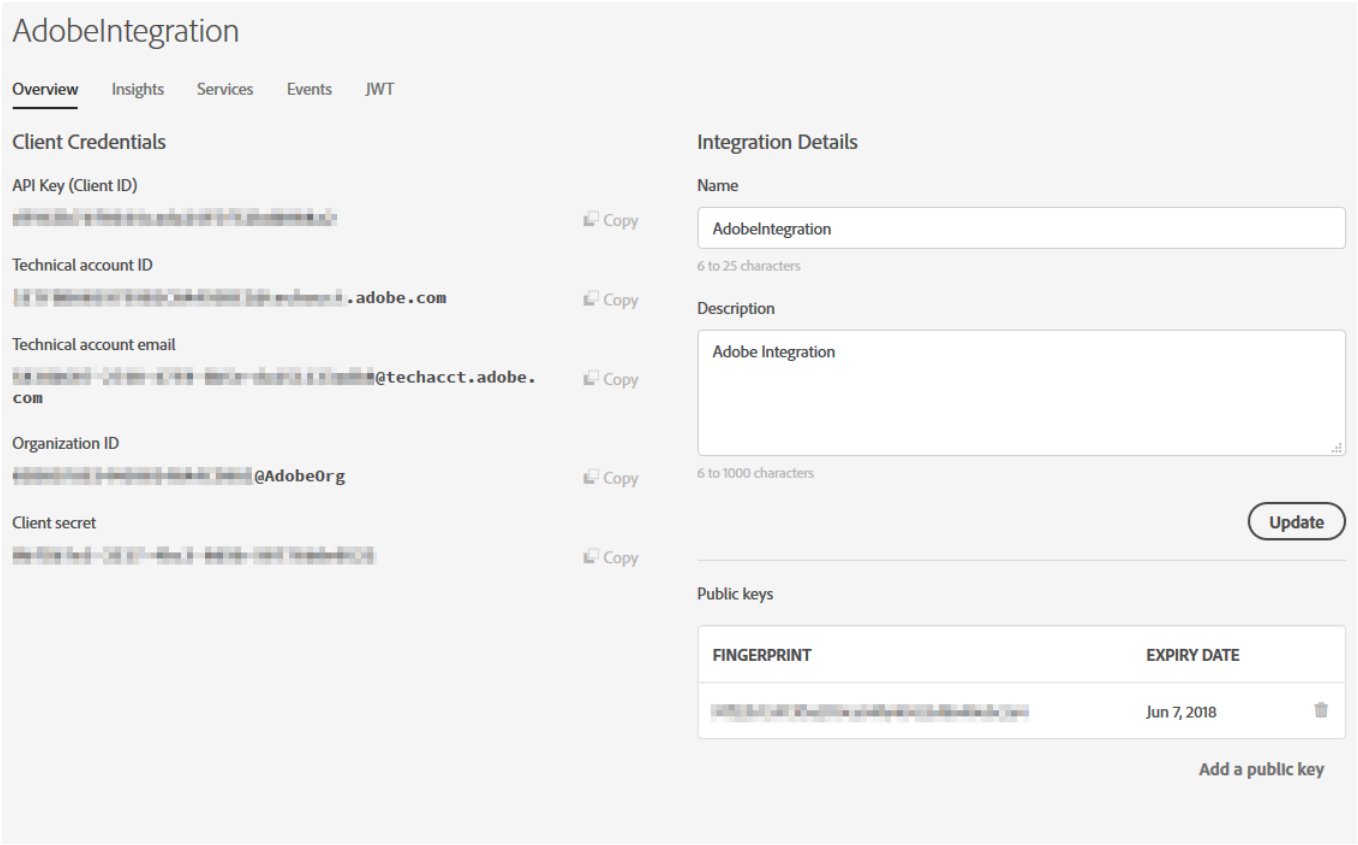
6. **Eingabe des Namens > Eingabe der Beschreibung (Description) >Auswahl des öffentlichen Schlüsselzertifikates > Create Integration**



7. Fertig



Nachdem die Integration eingerichtet ist, können die angezeigten Werte im Data Collector eingegeben und verwendet werden.



Achtung	Das Setup ermöglicht es ein eigenes Schlüsselpaar zu erstellen falls keines zur Verfügung steht. Nach der Erstellung liegt das Zertifikat in dem Ordner aus dem das Setup gestartet wurde. Falls ein verschlüsseltes Zertifikat verwendet werden soll, bitte das Setup fortsetzen und dann im Anschluss die Schlüsseldatei mit der Batchdatei "CreateCertificateAndKeyForAdobeCloud.cmd" im OpenSSL Ordner des Adobe Konnektors erstellen. Bitte nicht vergessen die neu erstellten Zertifikate in der Adobe Konsole und der Connector.config zu hinterlegen.

4.3.4 Microsoft Azure (Microsoft Online)

Systemvoraussetzungen

Achtung	Damit die Informationen über ein Benutzerkonto abgefragt werden können, braucht diese nur die Active Directory Rolle "User" im Azure AD. Damit eine Applikation die Daten abfragen kann, muss "Read directory data" bei den "APPLICATION PERMISSIONS" definiert sein.

Damit der Konnektor funktioniert muss das Windows PowerShell Modul **"AzureAD"** installiert sein.

Der nachstehende PowerShell Befehl kann verwendet werden um auf das Modul zu prüfen und es ggfs. zu installieren.

```
if (!(Get-Module -ListAvailable | where { $_.Name -eq "AzureAD"}))
{

        Install-Module AzureAd
}
```

Ein Skript mit dem o.g. Code mit dem Namen InstallModule_AzureAD.ps1 wurde in den Microsoft-Azure-Connector Ordner kopiert.

Achtung	Die Einstellungen für die PowerShell Execution Policy sind unbedingt zu beachten: PowerShell Execution Policy (siehe Seite 57)

Konfiguration für alle Konnektoren

Notiz	Microsoft Azure AD besteht aus mehreren Konnektoren, hier wird die Konfiguration beschreiben, die für alle Konnektoren gleich ist.

Connector.config Attribut	Pflichtangabe	Beschreibung
uid="<User>"	Nein	Benutzer der das Portal abfragen darf
pwd="<Password>"	Nein	Passwort des o.g. Benutzers
TenantId="<TenantID>"	Nein	TenantID für den Zugriff über eine Applikation
ApplicationId="<ApplicationID>"	Nein	ApplicationID für den Zugriff über eine Applikation
CertificateThumbprint="<ThumbPrint>"	Nein	Thumbprint für den Zugriff über eine Applikation
proxyAddress="<Proxy>"	Nein	Adresse des Proxy Servers (falls benötigt) um die Verbindung in das Internet zuzulassen. Proxy Adresse inklusive http(s)://
proxyPort="<Port>"	Nein	Port des Proxy Servers
proxyUser="<User>"	Nein	Benutzer der sich gegenüber dem Proxy Server authentifizieren kann.
proxyUserPassword="<Passwort>"	Nein	Passwort für o.g. Benutzer

Achtung	Es müssen entweder die Angaben "uid" und "pwd" ODER "TenantID", "ApplicationID" und "Thumbprint" angegeben sein, es MUSS eine von beiden Methoden verwendet werden, aber nicht beide gleichzeitig.

Beispieleinträge Connector.config:

Zugriff durch Applikation

```
<connector name="Microsoft-AzureAD" subfolder="Microsoft-Azure-Connector" active="true"
scriptname="Microsoft-AzureAD-Connector.ps1" TenantId="abc" ApplicationId="def" CertificateThumbprint="xyz" sfx="" />
```

Zugriff mit Benutzer

```
<connector name="Microsoft-AzureAD" subfolder="Microsoft-Azure-Connector" active="true"
scriptname="Microsoft-AzureAD-Connector.ps1" uid="my.user@domain.com" pwd"..." sfx="" />
```

Microsoft AzureAD - Nutzerbasierende Lizenzinformationen

Der Azure AD Lizenz Konnektor wird verwendet um Lizensierungsdaten aus Microsoft Azure zu exportieren.

Z.B. Office 365, EMS, Visio 365, Project 365, ...

Dieser Konnektor ersetzt den Microsoft Online Connector.

Konfiguration

Connector.config Attribut	Pflichtangabe	Beschreibung
additionalLicenseDetails="<True|False>"	Nein	Zusätzliche Lizenzinformationen ermitteln

Der Konnektor kann so eingestellt werden das nur bestimmte Attribute aus dem Azure Active Directory exportiert werden. Die Einstellung dazu erfolgt in der Datei **Microsoft-AzureAD-Connector-TenantAttributeConfig.txt** die im gleichen Pfad wie die .ps1 Datei liegt.

Die Microsoft-AzureAD-Connector-TenantAttributeConfig.txt ist eine Datei im CSV Stil.

Achtung	Änderungen in den Spalten ADAttribute, SWRDAttribute und Type sind nicht erlaubt, und können zu unerwarteten Ergebnissen führen.

Um den Export eines bestimmten Attributes zu unterdrücken, muss der Wert in der Spalte "Process" auf "Disabled" geändert werden.

ADAttribute,SWRDAttribute,Type,Process

ADAttribute,SWRDAttribute,Type,Process

DisplayName,DisplayName,System.String,Enabled

ObjectId,ObjectId,System.GUID,Enabled

DirSyncEnabled,DirSyncEnabled,System.String,Enabled

ObjectType,ObjectType,System.String,Enabled

PreferredLanguage,PreferredLanguage,System.String,Enabled

PostalCode,PostalCode,System.String,Disabled

CountryLetterCode,CountryLetterCode,System.String,Disabled

City,City,System.String,Disabled

State,State,System.String,Disabled

Country,Country,System.String,Disabled

TelephoneNumber,TelephoneNumber,System.String,Disabled

Street,Street,System.String,Disabled

Microsoft AzureAD - Benutzer Export

Der Azure Active Directory Benutzer Export ist ein Konnektor der alle(!) Benutzer aus dem Azure Active Directory exportiert.

Der Konnektor kann so eingestellt werden das nur bestimmte Attribute aus dem Azure Active Directory exportiert werden. Die Einstellung dazu erfolgt in der Datei **Microsoft-AzureAD-Connector-GetUsersAttributeConfig.txt** die im gleichen Pfad wie die .ps1 Datei liegt.

Die Microsoft-AzureAD-Connector-GetUsersAttributeConfig.txt ist eine Datei im CSV Stil.

Achtung	Änderungen in den Spalten ADAttribute, SWRDAttribute und Type sind nicht erlaubt, und können zu unerwarteten Ergebnissen führen.

Um den Export eines bestimmten Attributes zu unterdrücken, muss der Wert in der Spalte "Process" auf "Disabled" geändert werden.

ADAttribute,SWRDAttribute,Type,Process

OnPremisesSecurityIdentifier,ObjectSid,System.String,Enabled

AccountEnabled,UserAccountControl,System.Int32,Enabled

UserPrincipalName,UserPrincipalName,System.String,Enabled

Mail,EmailAddress,System.String,Enabled

Wichtig	Bei der Verwendung dieses Konnektors ist zu berücksichtigen, dass Benutzerkonten die aus dem Azure Active Directory exportiert werden, KEINEN SamAccountName besitzen.

Microsoft AzureAD - Gruppen Export

Der Azure Active Directory Group Export ist ein Konnektor der alle spezifizierten Gruppen und ihre Mitglieder aus dem Azure Active Directory exportiert.

Konfiguration

Connector.config Attribut	Pflichtangabe	Beschreibung
queryChildgroups="<true|false>"	Nein	Falls "True" werden die in den o.g. Gruppen enthaltenen Gruppen ebenfalls verarbeitet.
grp="Group1[,GroupN]	Ja	Liste der Gruppen die abgefragt werden sollen, getrennt durch Komma. Oder Angabe von File:<Dateiname> mit einer Datei die die Gruppen enthält.

Beispiele

```
-grp "Group1,AdminGropup,CitrixGroup"
```

Fragt alle Gruppen mit den Namen **Group1, AdminGroup, CitrixGroup** ab.

Übergabe der Gruppe(n) durch eine Textdatei.

```
-grp "File:MyGroups.txt"
```

Fragt alle in der Datei MyGroups.txt angegebenen Gruppen ab. MyGroups.txt muss eine Gruppe pro Zeile enthalten und im gleichen Verzeichnis liegen die die Microsoft-AzureAD-GetGroups.ps1 Datei.

Achtung	**File:** die Groß-/Kleinschreibung muss unbedingt beachtet werden! Die Verwendung von Platzhalterzeichen wie * ist nicht erlaubt!

Microsoft AzureAD - Einrichtung API Zugriff über eine Applikation

Alternativ zu der Verwendung eines Domänenbenutzers, kann eine Azure Applikation mit lesendem Zugriff eingerichtet werden. Die Verbindung wird durch ein (selbst erstelltes) Zertifikat realisiert, das auf der Maschine abgelegt ist auf der der Data Collector installiert ist.

Microsoft AzureAD - Erstellen des Zertifikates

Für den Zugriff wird ein Zertifikat mit privatem und öffentlichen Schlüssel benötigt. Falls ein solches Zertifikat nicht vorhanden ist, kann es mit dem Skript **Microsoft-AzureAD-CreateSelfSignedCertificate.ps1,** das im Ordner des Konnektors liegt, erstellt werden.

Achtung	Bei der Ausführung muss das Skript als Administrator ausgeführt werden, damit ist gewährleistet das das Zertifikat im Speicher der Maschine und nicht des angemeldeten Benutzers angelegt werden kann.

Parameter	Pflichtangabe	Beschreibung
-outputPath "<Pfad>"	Nein	Ausgabepfad der Zertifikatsdateien, falls der Parameter nicht angegeben wird, erfolgt die Ausgabe nach %TEMP% des ausführenden Benutzers.
-pwd "<Passwort>"	Ja	Passwort mit dem das Zertifikat geschützt wird.
-dnsName "<DNS Name>"	Nein	DNS Name der im Zertifikat hinterlegt wird, falls kein DNS Name angeben wird, wird der String "my.domain.local" verwendet. Gleichzeitig dient der DNS Name als Vorlage für die ausgegebenen Dateien.
-validfornYears =<xx>	Nein	Anzahl der Jahre für die das Zertifikat ab dem Generierungstag gültig ist, falls kein Wert angegeben wird, wird das Zertifikat für 10 Jahre ausgestellt.

Beispiel

Erstellen eines Zertifikats mit dem Password "MeinPasswort", dem DNS Namen "name.meinefirma.de" mit einer Gültigkeit von 5 Jahren nach C:\Temp

```
PowerShell -File ".\Microsoft-AzureAD-CreateSelfSignedCertificate.ps1" -outputPath "C:\Temp" -pwd "MeinPasswort" -dnsName
"name.meinefirma.de" -validfornYears 5
```

Ausgabe der Dateien in C:\Temp

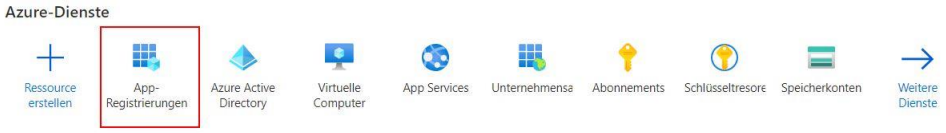
- name.meinefirma.de.pfx
- name.meinefirma.de.crt

Microsoft AzureAD - Einrichten der Applikation im Azure Portal

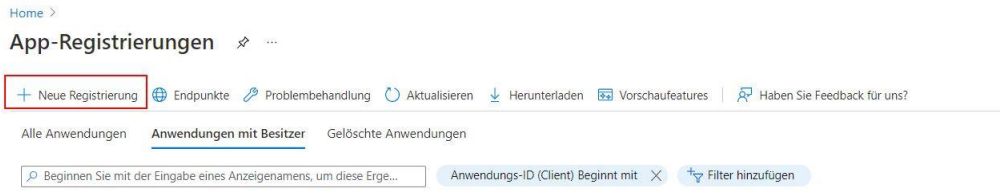
Der Zugriff erfolgt über eine Applikation, diese muss im Azure Portal https://portal.azure.com eingerichtet werden.

Anleitung

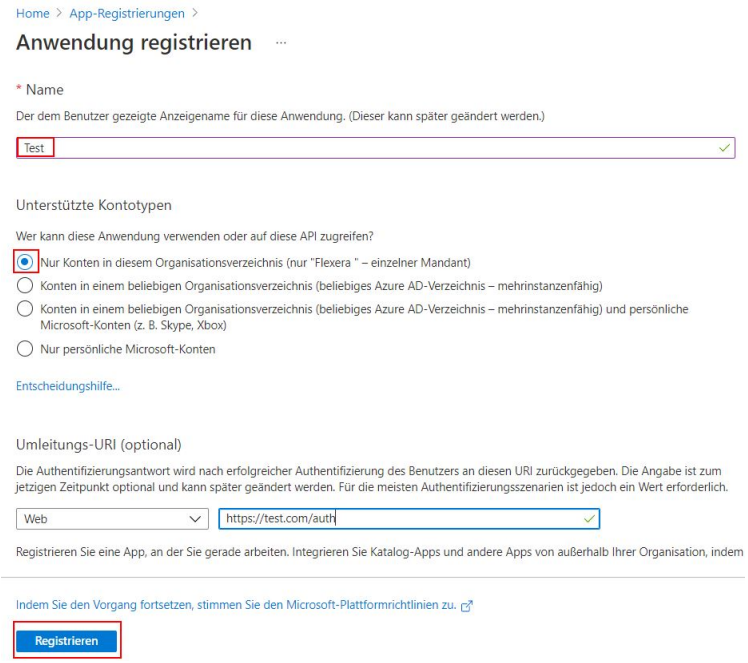
1. **Anmeldung am Portal**



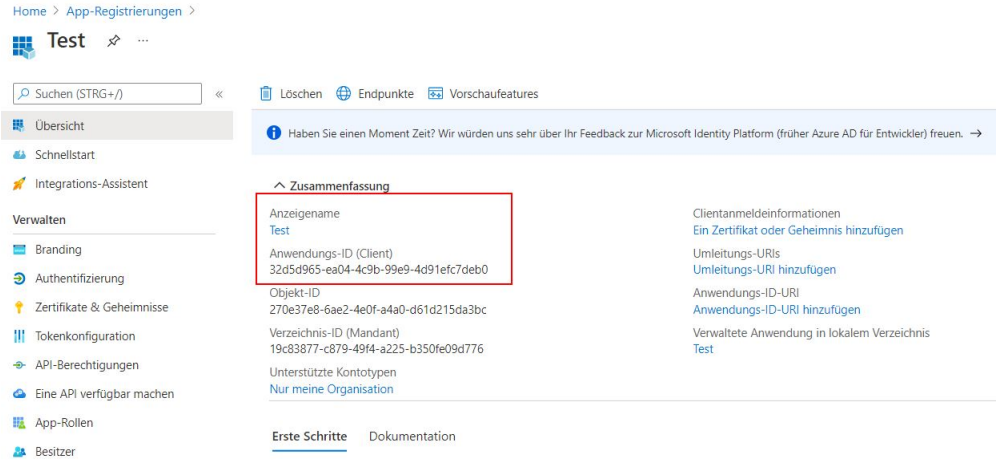
2. Auswahl von **App retistrations**, **New registration**



3. **Angabe von**
   Name: kann frei gewählt werden
   Ujnterstützte Kontotypen auswählen.
   Umleitungs-URI: kann frei gewählt werden (https:// muss angegeben werden)
   mit Registrieren bestätigen.

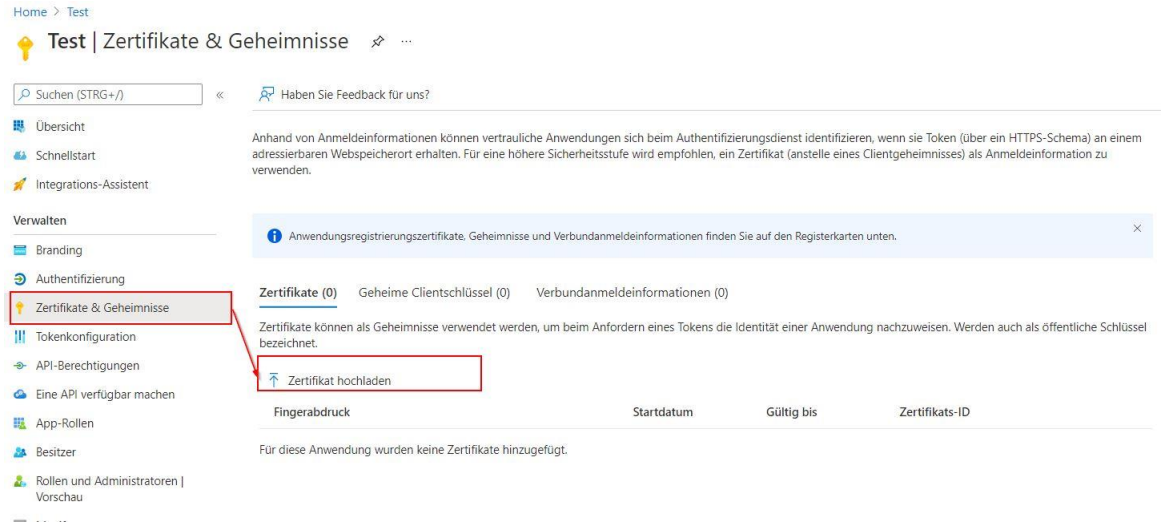


4. **Angelegte Applikation**

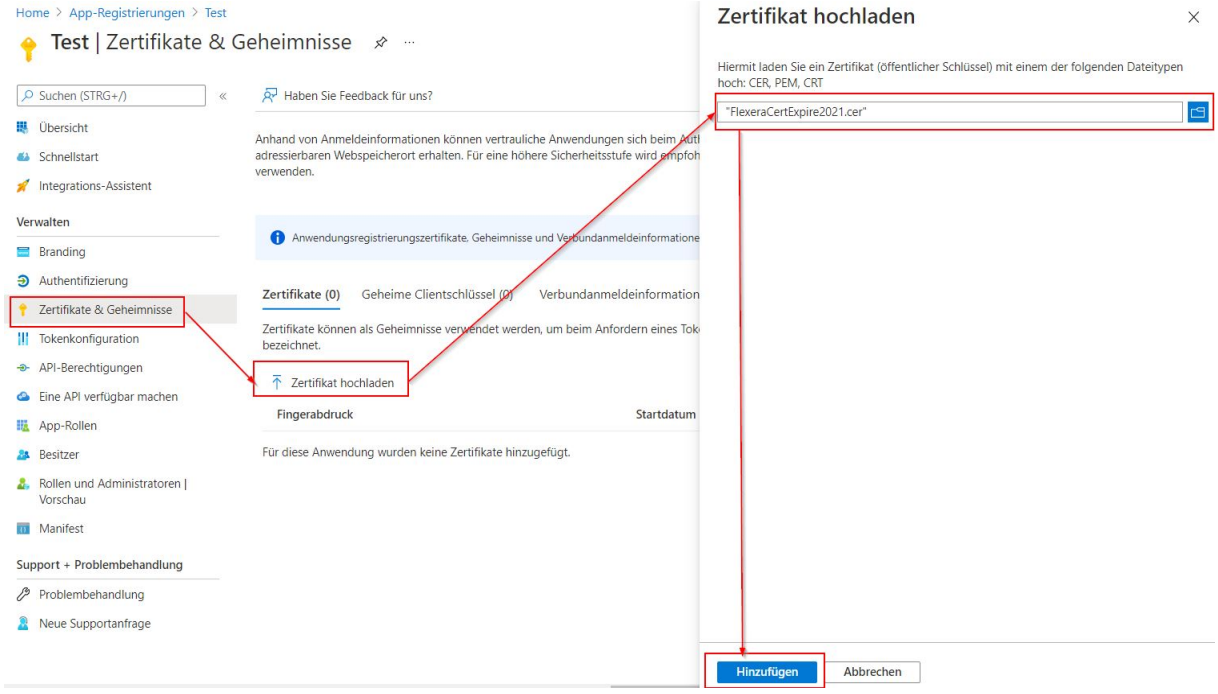


5. **Kopieren der ApplicationID und für spätere Verwendung sichern.**

6. **Settings**

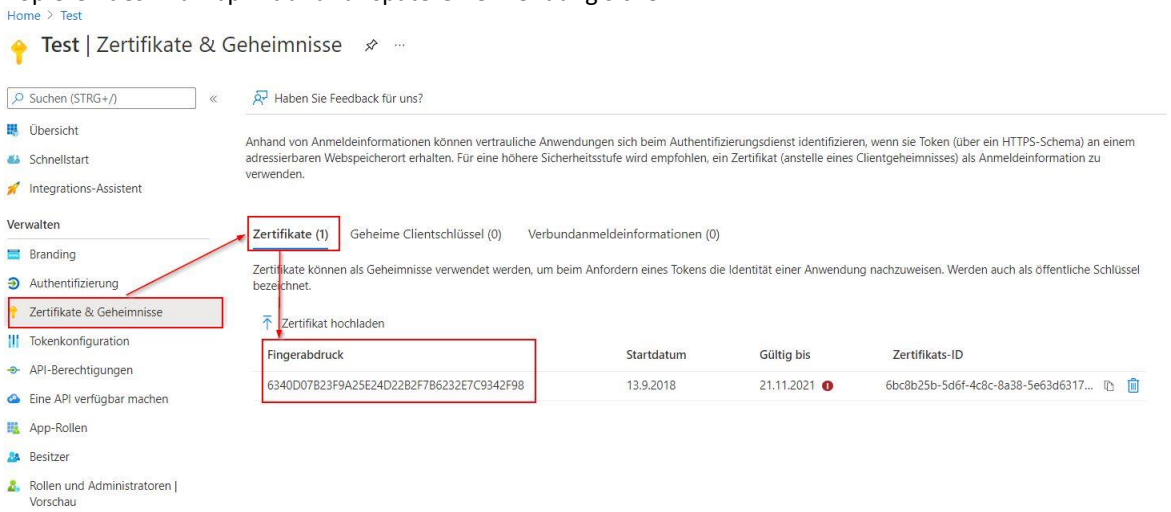


7. **Keys, Upload Public Key**

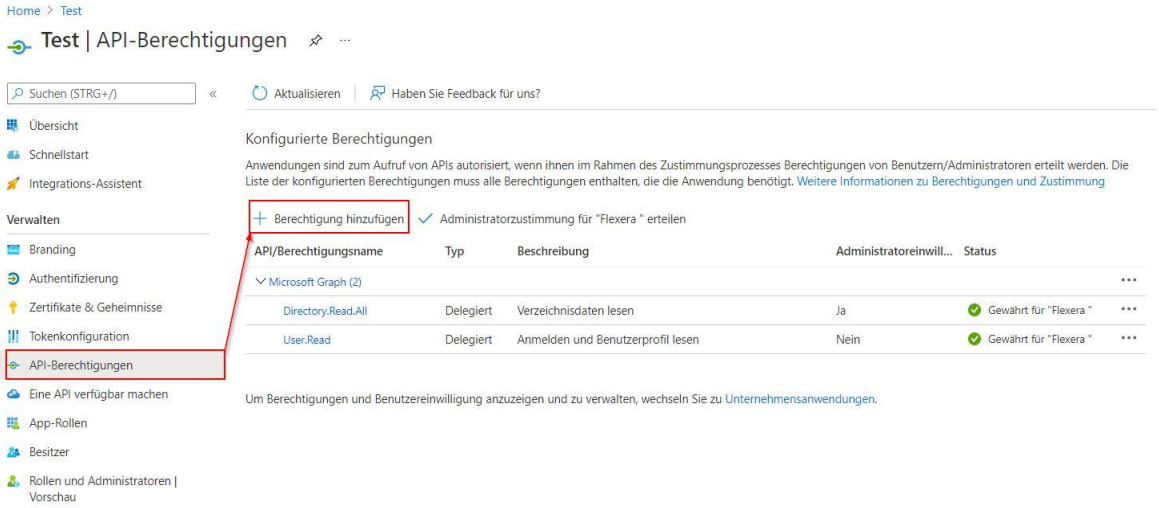


8. Key wählen, **Hinzufügen**

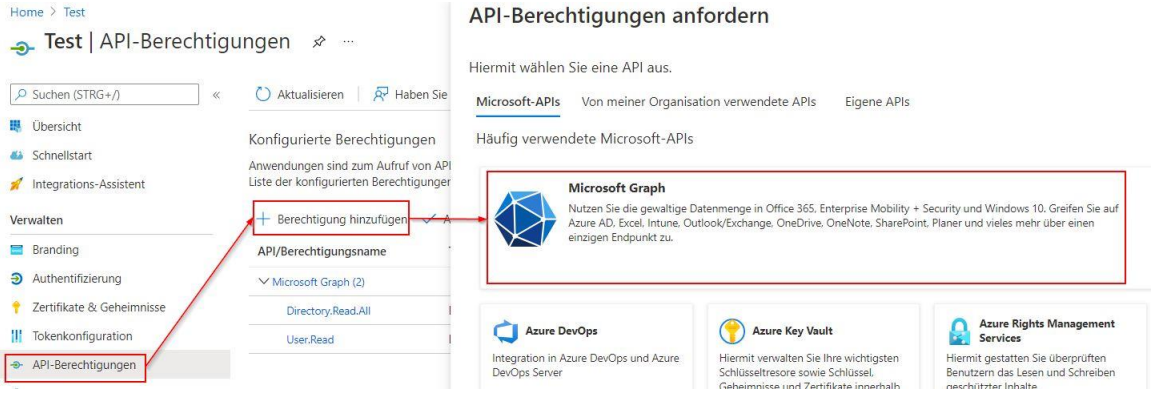
9. Kopieren des Thumbprint und für spätere Verwendung sichern.



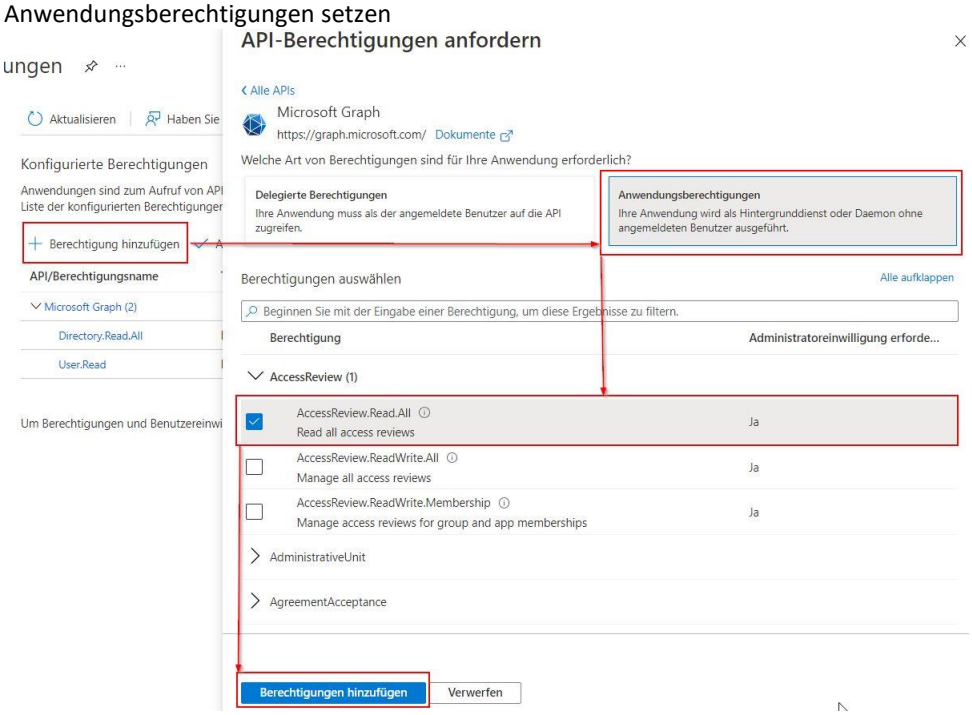
10. Required Permissions, Windows Azure Active Directory
11. Unter "API-Berechtigungen", **Berechtigung hinzufügen** wählen,



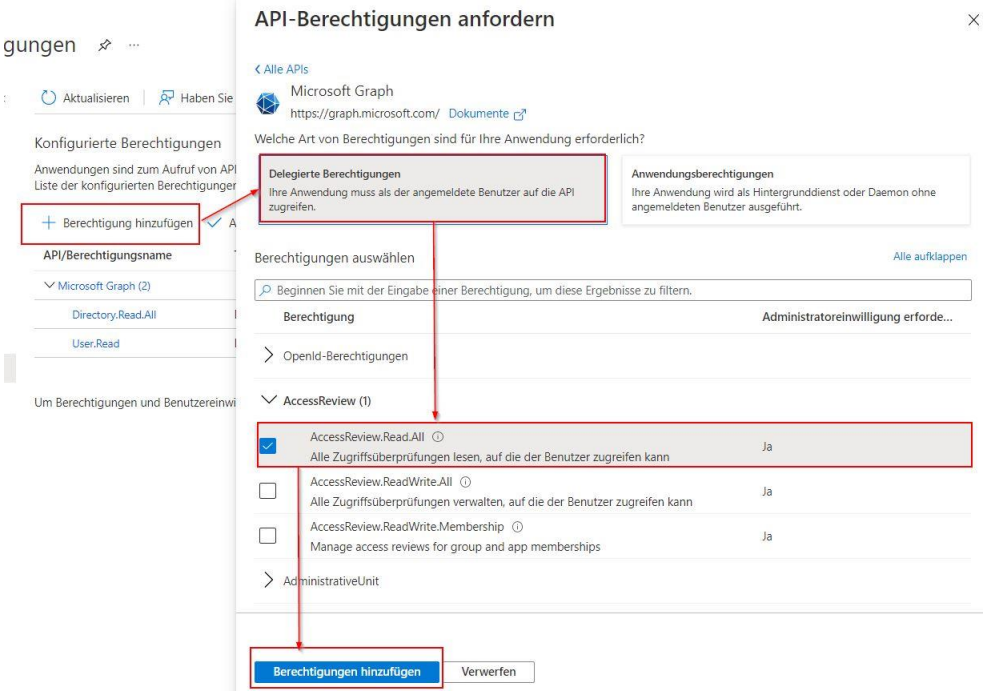
12. **Microsoft API auswählen**



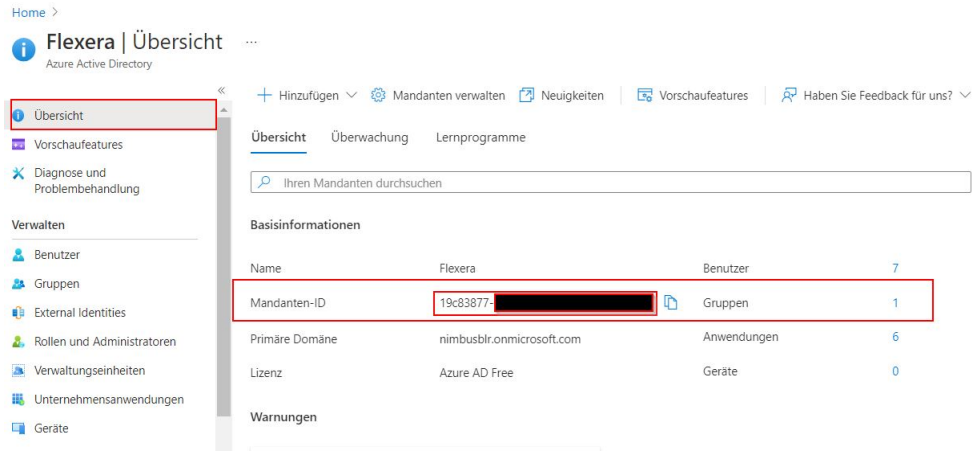
13. Anwendungsberechtigungen setzen



14. **Delegierte Berechtigungen setzen**



15. Fertig



Microsoft AzureAD - TenantID ermitteln

Azure Active Directory, Properties, Directory ID notieren und für spätere Verwendung sichern.

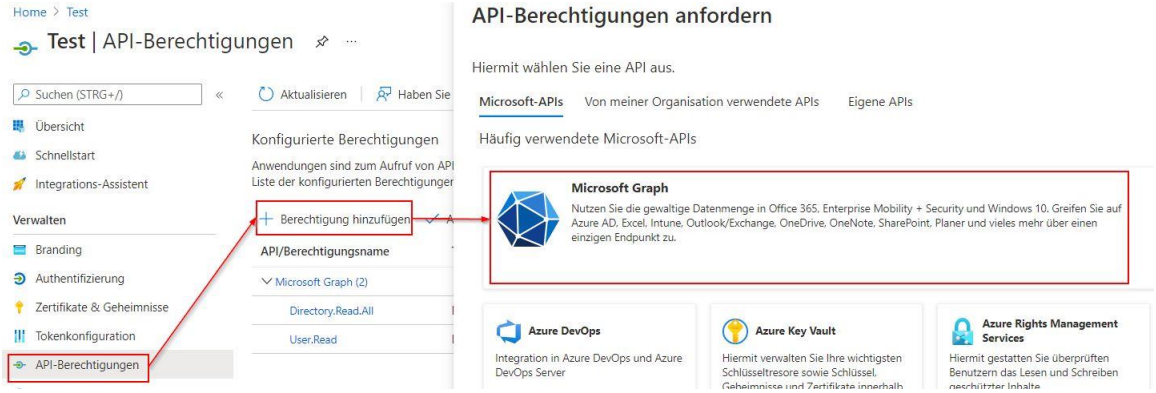


Abbildung - DirectoryID

4.3.5 Microsoft Intune

Die Device-Erkennung erfolgt über das Feld SerialNo.

Systemvoraussetzungen

Damit der Konnektor funktioniert muss das Windows PowerShell Modul **"AzureAD"** installiert sein.

Der nachstehende PowerShell Befehl kann verwendet werden um auf das Modul zu prüfen und es ggfs. zu installieren.

```
if (!(Get-Module -ListAvailable | where { $_.Name -eq "AzureAD"}))
{

       Install-Module AzureAd
}
```

Ein Skript mit dem o.g. Code mit dem Namen InstallModule_AzureAD.ps1 wurde in den Microsoft-Azure-Connector Ordner kopiert.

Achtung Die Einstellungen für die PowerShell Execution Policy sind unbedingt zu beachten: PowerShell Execution Policy (siehe Seite 57)

Konfiguration

Connector.config Attribut	Pflichtangabe	Beschreibung
uid="<User>"	Ja	Benutzer der das Portal abfragen darf
pwd="<Password>"	Nein	Passwort des o.g. Benutzers
ApplicationId="<ApplicationID>"	Ja	ApplicationID für den Zugriff über eine Applikation
h="true|false"	Nein	Export der Hardware
s="true|false"	Nein	Export der Software
proxyAddress="<Proxy>"	Nein	Adresse des Proxy Servers (falls benötigt) um die Verbindung in das Internet zuzulassen. Proxy Adresse inklusive http(s)://
proxyPort="<Port>"	Nein	Port des Proxy Servers
proxyUser="<User>"	Nein	Benutzer der sich gegenüber dem Proxy Server authentifizieren kann.
proxyUserPassword="<Passwort>"	Nein	Passwort für o.g. Benutzer

Beispieleinträge Connector.config:

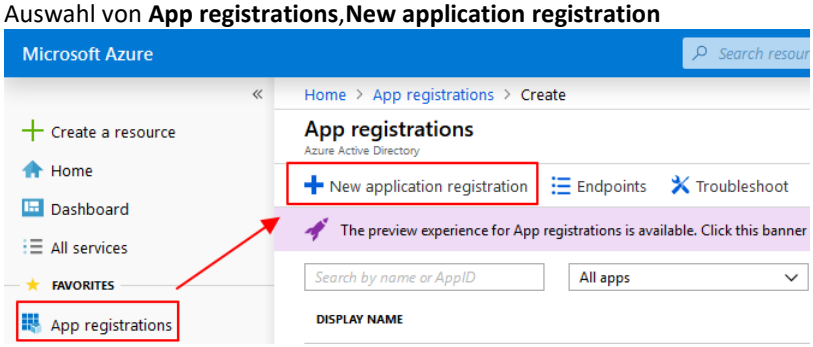
```
<connector name="Microsoft-Intune-Connector" subfolder="Microsoft-Intune-Connector" active="false"
scriptname="Microsoft-Intune-Connector.ps1" uid="user@domain.com" pwd=""
-applicationID="12345678-abcd-efgh-ijkl-mnopqrstuvwxy" />
```

Microsoft Intune - Einrichten der Applikation im Azure Portal

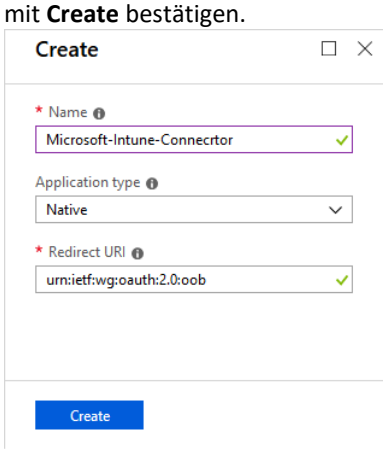
Der Zugriff erfolgt über eine Applikation, diese muss im Azure Portal https://portal.azure.com eingerichtet werden.

Anleitung

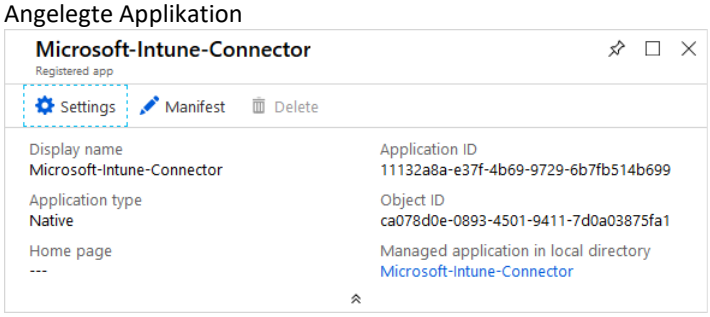
1. Anmeldung am Portal
2. Auswahl von **App registrations**,**New application registration**



3. Angabe von
   Name: kann frei gewählt werden
   Application Type: Native
   Sign-on URI: urn:ietf:wg:oauth:2.0:oob
   mit **Create** bestätigen.



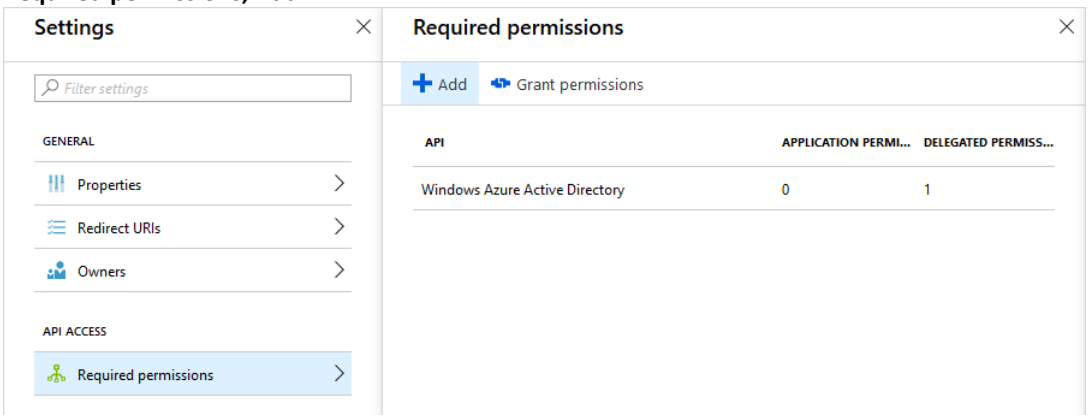
4. Angelegte Applikation



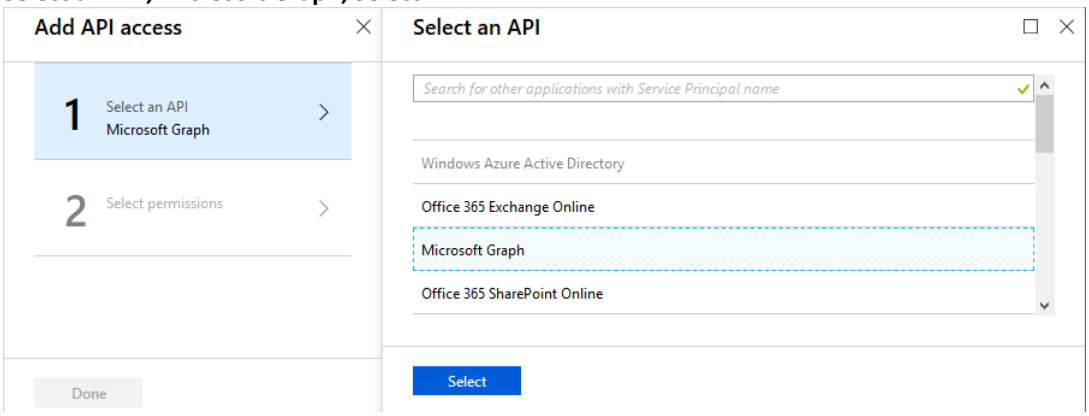
5. Kopieren der ApplicationID und für spätere Verwendung sichern

6. **Settings**

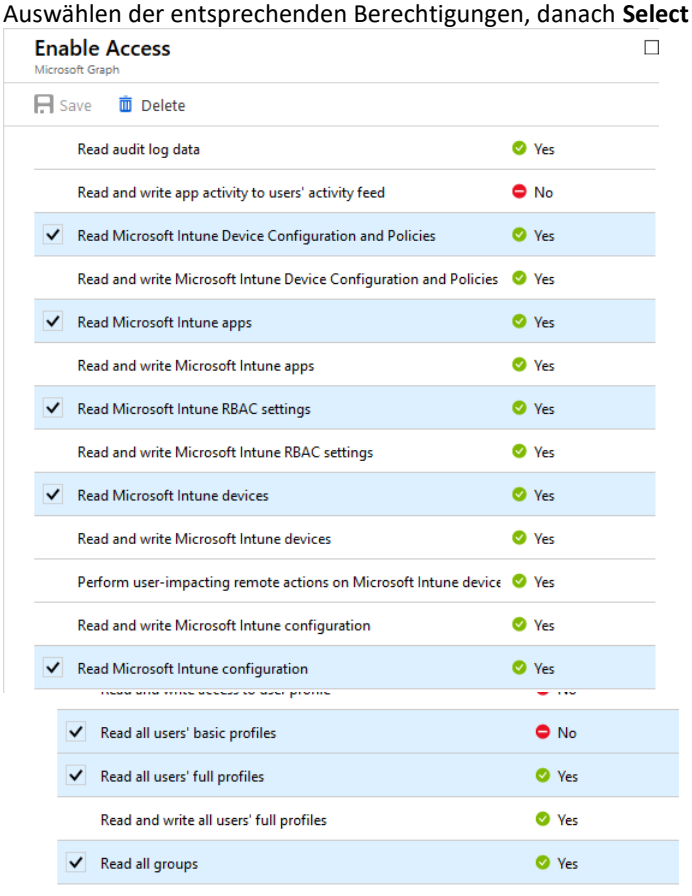
7. **Required permissions, Add**



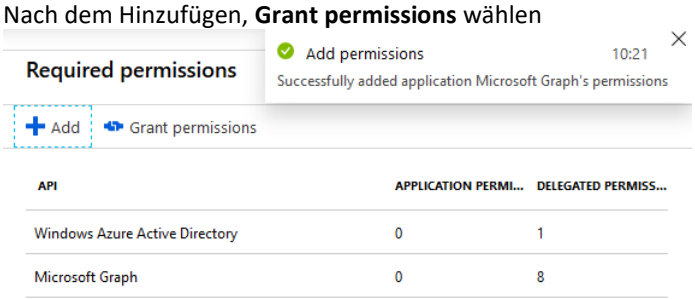
8. **Select an API, Microsoft Graph, Select**



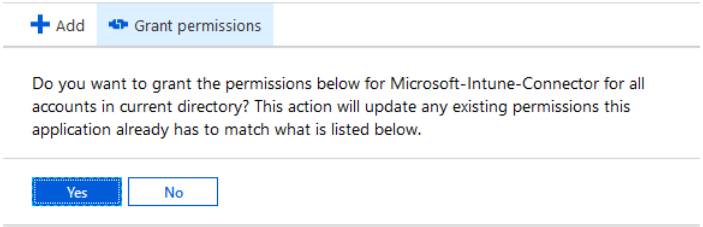
9. Auswählen der entsprechenden Berechtigungen, danach **Select**



10. Nach dem Hinzufügen, **Grant permissions** wählen



11. **Yes,** danach ist die Einrichtung abgeschlossen



12. Fertig

4.3.6 Microsoft Active Directory

Die Active Directory Konnektoren sind ein Satz an unterschiedlichen Konnektoren um Daten aus dem Active Directory zu exportieren.

Systemvoraussetzungen

Die Basis für das Funktionieren der Konnektoren, ist das Windows Feature **"RSAT-AD-PowerShell".**

Mit dem folgenden Codebeispiel kann die Installation von "RSAT-AD-PowerShell" überprüft und ggfs. das Feature installiert werden.

```
if (Get-WindowsFeature | Where-Object {($_.Name.Trim() -eq "RSAT-AD-PowerShell") -and ($_.Installed -eq $False)})
{
        Write-Host "RSAT-AD-PowerShell not found, installing...."
        Add-WindowsFeature  -name RSAT-AD-PowerShell
}
```

Ein Skript mit dem obigen Code wird im ADConnector Order abgelegt.

Achtung	Die Einstellungen für die PowerShell Execution Policy sind unbedingt zu beachten: PowerShell Execution Policy (siehe Seite 57)

Notiz	Die PowerShell Komponenten verwenden die Active Directory Web Services für den Zugriff auf das AD, Details hierzu sind im folgenden Artikel zu finden: https://blogs.msdn.microsoft.com/adpowershell/2009/04/06/active-directory-web-services-overview Der Standard Port für die Active Directory Web Services ist: 9389

Benutzerobjekte

Konfiguration

Connector.config Attribut	Pflichtangabe	Beschreibung
dc="<Domain Controller>"	Nein	DNS Name eines bestimmten Domain Controllers, nötig falls die automatische Auflösung unzuverlässig funktioniert.
uid="<Benutzer>"	Nein	Benutzerkonto mit Leseberechtigung in der Domain (in der Regel hat dies jeder Benutzer)
pwd="<Passwort>"	Nein	Passwort für o.g. Benutzer
filter="<Filter String>"	Nein	Um nicht erwünschte Benutzerobjekte auszufiltern, kann ein Filter implementiert werden um die Ausgabe zu Unterdrücken. Details zu den Filtereinstellungen sind hier beschrieben: https://technet.microsoft.com/en-us/library/hh531527.aspx
ou="<OU String>"	Nein	Einschränkung der Benutzer auf eine bestimmte OU. OU=user,OU=Test,OU=domain,DC=domain,DC=com

Notiz	Die angewandte Filtertechnik ist Active Directory Filterung und NICHT LDAP Filterung. Einige Filter sind bereits fest vorgegeben, d.h. alle im "filter" Attribut angegebenen Filter werden mit einem "und" an den bestehenden Filter angehängt!

Der Konnektor ist so eingestellt, dass nur ein bestimmter Satz an Attributen eines Active Directory Objektes exportiert werden. Die Einstellung findet in der Datei **GetADUserObjectsAttributeConfig.txt** die im gleichen Pfad wie die .ps1 Datei liegt.

Die GetADUserObjectsAttributeConfig.txt ist eine Datei im CSV Stil.

Achtung	Änderungen in den Spalten SWRDAttribute und Type sind nicht erlaubt.

Um den Export einer Spalte zu erlauben, muss der Wert in der Spalte "Process" auf Enabled stehen, um die Spalte nicht zu exportieren, ist "Disabled" zu verwenden.

ADAttribute,SWRDAttribute,Type,Process

objectsid,ObjectSid,System.String,Enabled

objectguid,ObjectGUID,System.Guid,Enabled

distinguishedname,DistinguishedName,System.String,Enabled

userprincipalname,UserPrincipalName,System.String,Enabled

Beispiele für Filter

Beschreibung	Filter
Nur Benutzer exportieren bei denen ein Vorname eingetragen ist	filter="(givenname -like '*')"
Nur Benutzer exportieren bei denen ein Nachname eingetragen ist	filter="(sn -like '*')"
Nur Benutzer exportieren die Vor- und Nachname eingetragen haben.	filter="((givenname -like '*')" -and (sn -like '*'))"
Nur Benutzer mit einer Emailadresse exportieren.	filter="(EmailAddress -like '*')"
Nur aktive Benutzer exportieren (=not disabled)	filter="(Enabled -ne $false)"
Nur Benutzer exportieren, deren Emailadresse auf "bwg.testing" endet	filter="(EmailAddress -like '*@bwg.testing')"

Notiz	Die korrekte Verwendung der einfachen und doppelten Anführungszeichen, ist unbedingt zu beachten!

Attribute

Die folgenden Attribute werden durch den Konnektor für Benutzerobjekte abgefragt. (Der in Klammern angegebene Name bezeichnet die Zielspalte in der Ausgabedatei.)

- c (CountryCode)
- cn (Name)
- co (Country)
- company
- department
- displayName
- distinguishedName

- employeeID (StaffNo)
- facsimileTelephoneNumber (FaxNo)
- givenName (Firstname)
- homePhone (PrivatePhoneNo)
- l (Location)
- mail (EmailAddress)
- mobile (MobilePhoneNo)
- NETBIOSName (NetBIOSDomainName)
- objectGUID
- objectSID
- physicalDeliveryOfficeName
- postalCode
- sAMAccountName
- sAMAccountType
- sn (Lastname)
- st (State)
- streetAddress
- telephoneNumber (PhoneNo)
- title (JobTitle)
- userAccountControl
- userPrincipalName

Computerobjekte

Der Active Directory Konnektor für Computer-Objekte ist ein Skript das die Computer-Objekte aus dem Active Directory exportiert.

Konfiguration

Connector.config Attribute	Pflichtangabe	Description
dc="<Domain Controller>"	Nein	DNS Name eines bestimmten Domain Controllers, nötig falls die automatische Auflösung unzuverlässig funktioniert.
uid="<Benutzer>"	Nein	Benutzerkonto mit Leseberechtigung in der Domain (in der Regel hat dies jeder Benutzer)
pwd="<Passwort>"	Nein	Passwort für o.g. Benutzer
InactiveDays="<xx>"	Nein	Nur Maschinen zurückgeben die Ihr Passwort in den letzten xx Tagen geändert haben.
ou="<OU String>"	Nein	Einschränkung der Benutzer auf eine bestimmte OU. OU=user,OU=Test,OU=domain,DC=domain,DC=com
filter="<Filter String>"	Nein	Um nicht erwünschte Computer-Objekte auszufiltern, kann ein Filter implementiert werden um die Ausgabe zu unterdrücken. Details zu den Filtereinstellungen sind hier beschrieben: https://technet.microsoft.com/en-us/library/hh531527.aspx

Attribute

Die folgenden Attribute werden durch den Konnektor für Computerobjekte abgefragt.

- name
- cn
- description
- distinguishedName
- dNSHostName
- objectGUID
- objectSid
- operatingSystem
- operatingSystemVersion
- operatingSystemServicePack
- userAccountControl
- sAMAccountType
- pwdLastSet

Gruppen-Objekte

Konfiguration

Connector.config Attribute	Pflichtangabe	Beschreibung
dc="<Domain Controller>"	Nein	DNS Name eines bestimmten Domain Controllers, nötig falls die automatische Auflösung unzuverlässig funktioniert.
uid="<Benutzer>"	Nein	Benutzerkonto mit Leseberechtigung in der Domain (in der Regel hat dies jeder Benutzer)
pwd="<Passwort>"	Nein	Passwort für o.g. Benutzer
grp="<GroupName(s)>"	Ja	Liste der Gruppen, mit Komma getrennt
strict="<true|false>"	Nein	Nur Informationen der angegebenen Gruppen sammeln, verbundene Gruppen (z.B. enthaltene Gruppen) werden ignoriert.
queryParentGroup="<true|false>"	Nein	Falls aktiviert, werden die sog. Elterngruppen der angegebenen Grupp(n) ermittelt.

Platzhalterzeichen und Gruppen die in einer Textdatei übergeben werden.

Bei der direkten Verwendung des -grp Parameters, können Platzhalterzeichen wie * angegeben werden.

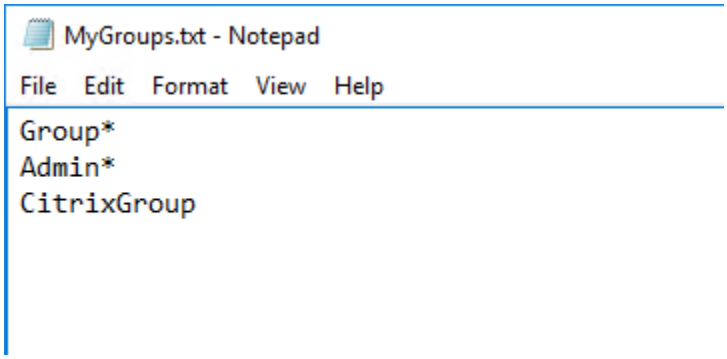
Beispiele

```
-grp "Group*,Admin*,CitrixGroup"
```

findet alle Gruppen die mit **Group** beginnen, alle Gruppen die mit **Admin** beginnen und die Gruppe **CitrixGroup.**

```
-grp "File:MyGroups.txt"
```

Fragt alle Gruppen ab, die in der Datei MyGroups.txt enthalten sind. MyGroups.txt, muss eine Gruppe pro Zeile enthalten und im gleichen Verzeichnis wie die .ps1 Datei liegen. Das * als Platzhalterzeichen ist erlaubt.



Achtung	**File:** es muss die Groß-/Kleinschreibung unbedingt beachtet werden! Microsoft Active Directory limitiert die Anzahl der zurückgegebenen Gruppenmitglieder auf 5.000, eine detaillierte Beschreibung kann hier gefunden werden: https://technet.microsoft.com/en-us/library/dd391908(WS.10).aspx . Als Workaround kann der Microsoft AzureAD - Group Export "" verwendet werden der diese Einschränkung nicht aufweist.
Notiz	Ein alternatives Skript "GetADGroupObjects2.ps1" liegt ebenfalls im ADConnector Pfad, hiermit ist der Export mit mehr als 5.000 Gruppenmitgliedern möglich. Allerdings werden hierfür zusätzliche Abfragen im Active Directory ausgeführt was die Performance beeinflussen kann.

Attribute

Die folgenden AD Attribute werden durch den Konnektor abgefragt.

- Group
  - ObjectGUID
  - ObjectSID
  - Name
  - DistinguishedName
  - samAccountName
- Group member
  - GroupObjectSID
  - GroupObjectGUID
  - GroupDistinguishedName
  - MemberObjectGUID
  - MemberObjectSID
  - MemberName
  - MemberDistinguishedName
  - MemberSamAccountName
  - MemberObjectClass

4.3.7 Microsoft Application Virtualization (App-V) Konnektor

Dieser Konnektor besteht aus mehreren Teilen.

Konnektor	Beschreibung
Microsoft-AppV-Connector.ps1	Konnektor der das App-V PowerShell Modul verwendet das mit der Installation der App-V Management Konsole zur Verfügung gestellt wird.
Microsoft-AppV-SQL-Connector.ps1	SQL basierender Konnektor, ermittelt die gleichen Informationen wie der vorherige Konnektor direkt aus der SQL Datenbank.
Microsoft-AppV-SQL-PackageApplicationUsage.ps1	SQL basierender Konnektor der die Verwendungsdaten (Usage Data) der App-V Pakete direkt aus der Datenbank ermittelt..

Wichtig:	Zur Ermittlung der Paketdaten kann entweder das Skript "Microsoft-AppV-Connector.ps1" oder das Skript "Microsoft-AppV-SQL-Connector.ps1" verwendet werden. Die Unterschiede werden in den folgenden Kapiteln beschrieben.

Achtung	Die Einstellungen für die PowerShell Execution Policy sind unbedingt zu beachten: PowerShell Execution Policy (siehe Seite 57)

App-V Paketdaten, PowerShell basierend

Systemvoraussetzungen

Die Funktion des Konnektors basiert auf dem App-V PowerShell Module für App-V Server, dieses wird im Zuge der App-V Management Konsole mit installiert.

Um die Daten zu exportieren, baut das Skript von der Maschine auf der der SDC installiert ist eine Verbindung zum App-V Server auf. Dort wird das mit der App-V Installation bereitgestellte PowerShell Modul "AppVServer" verwendet, um die Daten zu ermitteln.

Damit eine Verbindung möglich ist muss PSRemoting auf der Zielmaschine aktiviert sein, Details dazu sind im Kapitel PSRemoting (siehe Seite 56) zu finden.

Konfiguration

Connector.config Attribut	Pflichtangabe	Beschreibung
uid="<Benutzer>"	Nein	Optional: Benutzer zum Verbinden auf den App-V Server
pwd="<Passwort>"	Nein	Passwort zu o.g. Benutzer
server="<App-V Server>"	Ja	Name des App-V Servers der die Management Konsole und Datenbank installiert hat. Wenn der Data Collector direkt auf dem App-V Server installiert ist, kann dieser Eintrag weggelassen werden.

Ausgabe

Der Microsoft App-V Konnektor ermittelt die folgenden Details:

- Packages
- Package Application
- Package Entitlements / AD Groups

App-V Paketdaten, SQL basierend

Systemvoraussetzungen

Keine

Konfiguration

Connector.config Attribut	Pflichtangabe	Beschreibung
uid="<Benutzer>"	Nein	Optional: Benutzer um sich auf die Datenbank zu verbinden
pwd="<Passwort>"	Nein	Passwort zu o.g. Benutzer
server="<App-V Server>"	Ja	Name des SQL Servers auf dem die Management Datenbank von App-V installiert ist.
database="App-V Management database>	Ja	Name der App-V Management Datenbank

Ausgabe

Der Microsoft App-V Konnektor ermittelt die folgenden Details:

- Packages

- Package Application
- Package Entitlements / AD Groups

App-V Nutzungsdaten, SQL basierend

Systemvoraussetzungen

Keine

Konfiguration

Connector.config Attribut	Pflichtangabe	Description
uid="<Benutzer>"	Nein	Optional: Benutzer mit dem die Verbindung auf die Datenbank hergestellt wird.
pwd="<Passwort>"	Nein	Passwort zu o.g. Benutzer
server="<App-V Server>"	Ja	Name des SQL Servers auf dem die App-V Reporting Datenbank liegt.
database="App-V Management database>	Ja	Name der App-V Reporting Datenbank
days="<Days>"	Nein	Optional: Anzahl der Tage die in der Historie zurückgegangen wird. Wird der Wert weggelassen, wird 7 Tage in der Historie zurückgegangen.

Ausgabe

Der Microsoft App-V Konnektor gibt die folgenden Details aus:

- PackageUsage

4.3.8 Hyper-V

Systemvoraussetzungen

Für die ordnungsgemäße Funktion des Konnektors, muss das Windows Feature **"Hyper-V-PowerShell"** installiert sein.

Das folgende PowerShell Kommando kann verwendet werden um die Installation des Powershell Moduls Hyper-V-PowerShell zu überprüfen und ggfs. zu installieren.

```
if (-Not (Get-Module -ListAvailable | Where-Object {($_.Name.Trim() -eq "Hyper-V")})) {
       Write-Host "Hyper-V-PowerShell not found, installing...."
       Install-WindowsFeature -Name Hyper-V-PowerShell
}
```

Ein Skript mit dem obigen Code mit dem Namen InstallWindowsFeature_Hyper-V-PowerShell.ps1 liegt im Ordner Hyper-V.

Achtung	Die Einstellungen für die PowerShell Execution Policy sind unbedingt zu beachten: PowerShell Execution Policy (siehe Seite 57)

Konfiguration

Connector.config Attribut	Pflichtangabe	Beschreibung
uid="<Benutzer>"	Nein	Optional: Benutzer der die Domain abfragen kann
pwd="<Passwort>"	Nein	Passwort des o.g. Benutzers
srv="<Hyper-V Server>"	Nein	Optional: Hyper-V server der abgefragt werden soll
file="<Dateipfad>"	Nein	Optional: Pfad der Datei die eine Liste von Hyper-V Servern enthält

Achtung	Der Konnektor benötigt die Angabe des Connector.config Attribut srv oder file.

Notiz	Die konfigurierte Datei darf nur einen Hyper-V Server pro Textzeile enthalten. Es sind keine weiteren Separatoren notwendig.

Ausgabe

Der Hyper-V Konnektor ermittelt die folgenden Details:

- DeviceRelationshipTypeID
- ChildDeviceUUID
- ParentDeviceUUID
- ParentDeviceURN
- ScanDatum

Notiz	Der Hyper-V Konnektor ermittelt nur die Host-Gast Beziehungen; Cluster Informationen werden in einem späteren Release hinzugefügt.

4.3.9 Hyper-V via Virtual Machine Manager

Systemvoraussetzungen

Damit der Konnektor arbeiten kann, muss das Modul **"virtualmachinemanager"** installiert sein, das Modul wird mit der Installation des Microsoft Virtual Machine Manager zur Verfügung gestellt.

Achtung	Dieser Konnektor funktioniert nur wenn der Data Collector auf der Maschine installiert ist, auf dem der Virtual Machine Manager ausgeführt wird.

Achtung	Die Einstellungen für die PowerShell Execution Policy sind unbedingt zu beachten: PowerShell Execution Policy (siehe Seite 57)

Konfiguration

Connector.config Attribut	Pflichtangabe	Beschreibung
uid="<Benutzer>"	Nein	Benutzer der die Domain abfragen kann.
pwd="<Passwort>"	Nein	Passwort des o.g. Benutzers
srv="<Hyper-V VMM Server>"	Ja	Hyper-V Server der den Virtual Machine Manager ausführt.

Ausgabe

Der Hyper-V VMM Konnektor ermittelt die folgende Werte:

- Hardware (auch Cluster, falls vorhanden)
  - Information der Hyper-V Cluster wenn vorhanden
- Beziehung
  - DeviceRelationshipTypeID
  - ChildDeviceUUID
  - ParentDeviceUUID
  - ParentDeviceURN
  - ScanDatum

4.3.10 Microsoft Exchange Connector (Beta)

Achtung	Dies ist ein Beta Konnektor die Ergebnisse können Unvollständig oder Fehlerhaft sein.

Systemvoraussetzungen

Für die ordnungsgemäße Funktion des Konnektors muss das Windows Feature **"RSAT-AD-PowerShell"** installiert sein.

Der folgende PowerShell Befehl kann verwendet werden um die Installation des Features RSAT-AD-PowerShell zu überprüfen und ggfs. zu installieren.

```
if (Get-WindowsFeature | Where-Object {($_.Name.Trim() -eq "RSAT-AD-PowerShell") -and ($_.Installed -eq $False)})
{
        Write-Host "RSAT-AD-PowerShell not found, installing...."
        Add-WindowsFeature  -name RSAT-AD-PowerShell
}
```

Ein Skript mit dem obigen Code mit dem Namen **InstallWindowsFeature_RSAT-AD-PowerShell.ps1** ist im ADConnector Ordner zu finden.

Achtung	Die Einstellungen für die PowerShell Execution Policy sind unbedingt zu beachten: PowerShell Execution Policy (siehe Seite 57)

Konfiguration

Connector.config Attribut	Pflichtangabe	Beschreibung
uid="<Benutzer>"	Nein	Optional: Benutzer der die Domain abfragen kann.
pwd="<Passwort>"	Nein	Passwort des o.g. Benutzers.

Ausgabe

Der Microsoft Exchange Konnektor gibt die folgenden Details aus:

- ObjectSID
- ObjectGUID
- DistinguishedName
- UUID
- Name
- Edition
- AdminDisplayVersion
- ProductID
- ExchangeVersion

4.3.11 LDAP (Beta)

Achtung Dies ist ein Beta Konnektor die Ergebnisse können Unvollständig oder Fehlerhaft sein.

Systemvoraussetzungen

Keine

Konfiguration

Connector.config Attribut	Pflichtangabe	Beschreibung
uid="<Benutzer>"	Nein	Optional: Benutzer der die Verbindung zum LDAP aufbaut.
pwd="<Passwort>"	Nein	Passwort zu o.g. Benutzer
dc="<LDAP server>"	Ja	LDAP Server der abgefragt wird.
filter="<Filter String>"	Nein	Filter um Benutzerkonten zu filtern die nicht erwünscht sind: http://www.ldapexplorer.com/en/manual/109010000-ldap-filter-syntax.htm
ou="<OU String>"	Nein	Optional: Nur Benutzer aus einer bestimmten OU zurückgeben OU=user,OU=Test,OU=domain,DC=domain,DC=com

Der LDAP Konnektor kann so konfiguriert werden, dass er nur bestimmte Attribute eines LDAP Objektes zurückliefert. Die Konfiguration dazu findet in der Datei **AttributeConfig.txt** statt, die im gleichen Pfad wie die .ps1 Datei liegt.

Die AttributeConfig.txt ist eine Datei im CSV Stil, es sind Beispiele für Windows und Novell eDirectory enthalten.

Achtung Änderungen in den Spalten SWRDAttribute und Type sind nicht erlaubt, und können zu unerwarteten Ergebnissen führen.

Wenn der Export eines bestimmten Attributes unterbunden werden soll, muss "Disabled" für diesen Wert in der Spalte "Process" eingetragen werden, zum aktivieren wird "Enabled" verwendet.

LDAPAttribute,SWRDAttribute,Type,Process

objectsid,ObjectSid,System.String,Enabled

objectguid,ObjectGUID,System.Guid,Enabled

distinguishedname,DistinguishedName,System.String,Enabled

userprincipalname,UserPrincipalName,System.String,Enabled

4.3.12 XEN Server (Beta)

Achtung	Dies ist ein Beta Konnektor die Ergebnisse können Unvollständig oder Fehlerhaft sein.

Systemvoraussetzungen

Für die ordnungsgemäße Funktion des Konnektors muss das XEN PowerShell Modul installiert sein, das Modul kann unter: http://xenserver.org/partners/developing-products-for-xenserver.html gefunden werden. Das XEN PowerShell Modul benötigt Microsoft .NET 4.5 und PowerShell v4.

Achtung	Die Einstellungen für die PowerShell Execution Policy sind unbedingt zu beachten: PowerShell Execution Policy (siehe Seite 57)

Konfiguration

Connector.config Attribut	Pflichtangabe	Beschreibung
uid="<Benutzer>"	Nein	Optional: Benutzer der die Domain abfragen kann.
pwd="<Passwort>"	Nein	Passwort des o.g. Benutzers.
srv="<XEN server>"	Ja	Name des abzufragenden XEN Servers

Ausgabe

Der XEN Konnektor ermittelt die folgenden Details:

- Hardware (des XEN Servers)
  - UUID
  - DomainName
  - HostName
  - Manufacturer
  - ScanDate
  - Model
  - ProcessorManufacturer
  - ProcessorType
  - ProcessorSpeed
  - CPUCount
  - CorePerCPU
  - CPUCoreCount
  - CPULogicalCount
  - Mac1
  - Mac2
  - Mac3
  - Mac4
  - MemoryMB
  - IPAddressV4
  - OSCaption
  - BIOSVersion

- SerialNo
- InventorySource

- Relationships
  - DeviceRelationshipTypeID
  - ChildDeviceUUID
  - ParentDeviceUUID
  - ScanDate

Notiz	Der XEN Konnektor ermittelt nur die Hardware des XEN Servers und die Host-Gast Beziehungen, Hardwaredetails der Gastmaschinen werden nicht ermittelt.

Spider/Columbus Inventory (Windows / Mac OS)

5.1 Windows

5.1.1 Systemvoraussetzungen Columbus Inventory

Die minimalen Voraussetzungen an das Betriebssystem für Columbus Inventory sind:

- Windows XP (32bit) SP3
- Windows 2003 (32/64bit) SP2

5.1.2 DSGVO / GDPR Verhalten

Ab Version 7.5.5.17 werden Felder mit personenbezogenen Daten nicht mehr automatisch erhoben, davon sind die folgenden Felder betroffen.

Inventory	Felder
HardwareScan.csv	LastLoggedOnUser LastLoggedOnSAMUser LastLoggedOnUserSID MAC1 MAC2 MAC3 MAC4 IPAddressV4 IPAddressV6
InventoryItems.csv	OS.System.RegisteredUser OS.System.Organization OS.System.ProductKey

Bei Bedarf kann die Erhebung der Daten wieder eingeschaltet werden, Details dazu sind in der Beschreibung der Konfigurationsdateien für den Inventory Agent (siehe Seite 95) und Inventory Scanner (siehe Seite 107) zu finden.

5.1.3 Columbus Inventory Agent

Der Columbus Inventory Agent ist eine Scanning Dienst der auf einer Maschine installiert wird, zusätzlich zur normalen Inventarisierung kann der Dienst auch Metering Daten erheben.

Columbus Inventory Agent Quelldateien

Der Inventory Agent kann im Installationsverzeichnis (Abbildung - Installationspfad (siehe Seite 6)) im Unterordner "ColumbusInventoryAgent", gefunden werden, er besteht aus den Dateien:

- ColumbusInventoryAgent.cfg
- ColumbusInventoryAgent.exe
- ColumbusInventoryAgentUpdater.exe
- libeay32.dll
- ssleay32.dll

(Die ColumbusInventoryAgentUpdater.exe ist für die Verteilung nicht nötig, die Datei wird an diesem Ort lediglich für eine Verwendung zur Verfügung gestellt.)

Columbus Inventory Agent Konfiguration

Der Agent wird als Dienst installiert und startet automatisch mit dem Hochfahren der Maschine.

Ein Scan findet unter den folgenden Bedingungen statt:

- der Inventory Agent wurde neu installiert und startet zum ersten Mal
- die Zeitspanne in InvScanStartPeriod ist seit dem letzten Scan abgelaufen
- die Information zum letzten Scan Zeitpunkt wurde zurückgesetzt

Die Konfiguration des Inventory Agenten wird durch die ColumbusInventoryAgent.cfg bestimmt, diese muss mindestens den Zielserver für die Übertragung der Ergebnisse enthalten.

```
[Transmitter]
InvOTB_Host=yourserver.yourdomain.local
```

Sektion	Parameter	Standardwert (falls leer)	Mögliche Werte / Beschreibung
Scanner	InvFunction	2	0 = HW, SW, Inv. Items 1 = HW, SW, Inv. Items, File Scan 2 = HW, SW, Inv. Items, File Scan, Metering
Scanner	InvDrives	Alle lokalen Laufwerke	CDE (bedeutet die Laufwerke C:, D: und E:)
Scanner	InvExtensions	.EXE	Liste der Dateiendungen, für die detaillierte Informationen während einem File Scan erhoben werden.
Scanner	InvExportPath	%ProgramData%\Columbus	Lokaler Pfad in dem die Scan Ergebnisse vorgehalten werden bevor sie übermittelt werden. Z.B. %temp% oder %_ExePath%
Scanner	InvUpdateAgent	1	Automatisches Update des Agenten 0 = Abgeschaltet 1 = Eingeschaltet

Sektion	Parameter	Standardwert (falls leer)	Mögliche Werte / Beschreibung
Scanner	InvUpdateEngine	1	Automatisches Update der Scan Signaturen (Scanner Addon DLLs) 0 = Abgeschaltet 1 = Eingeschaltet
Scanner	InvScanStartPeriod	daily	Häufigkeit des Scans daily = alle 24 h weekly = einmal pro Woche monthly = einmal pro Monat
Scanner	InvScanStartDelay	0	Startverzögerung, Scan beginnt in angegebenen Zeitraum NACH dem Start des Service n Minuten (0-100)
Transmitter	InvTransmissionMode	3	Übertragungsmethode der Ergebnisse 0 = Keine Übertragung, Offline Modus 1 = FTP 3 = OTB
Scanner	InvLastObject	0	1 = Erhebung der Benutzerinformationen • LastLoggedOnUser • LastLoggedOnSAMUser • LastLoggedOnUserSID
Scanner	InvNetwork	0	1 = Erhebung der Netzwerkinformationen • MAC1 • MAC2 • MAC3 • MAC4 • IPAddressV4 • IPAddressV6
Scanner	InvLicensee	0	1 = Erhebung der Lizenzinformationen des OS • OS.System.RegisteredUser • OS.System.Organization • OS.System.ProductKey
Transmitter	InvOTB_Host		FQDN der Maschine die die Ergebnisse entgegen nimmt.
Transmitter	InvOTB_Port	24786	Port der Maschine die die Ergebnisse entgegen nimmt.
Transmitter	InvFTP_Host		Hostname des FTP Servers
Transmitter	InvFTP_Port		Port des FTP Servers
Transmitter	InvFTP_User		FTP-Server Authentifizierung, Benutzer (Falls leer, wird Anonymous verwendet)
Transmitter	InvFTP_Password		FTP-Server Authentifizierung, Passwort (Verschlüsselung mit cryptit.exe)
DirectoryFilter	InvDirectoryFilter001 ... InvDirectoryFilter999		Filter für Verzeichnisse die vom Scan ausgeschlossen werden sollen Akzeptiert Windows Variablen und feste Pfade z.B. %windir%* oder D:\Data*

Standardfilter

Der Agent wird mit einigen Standardfiltern vorbelegt:

```
[DirectoryFilter]
InvDirectoryFilter000=*\microsoft system center 2012\dpm\dpm\volumes\*
InvDirectoryFilter001=%windir%\$*_$\*
InvDirectoryFilter002=%windir%\*\$*_$\*
InvDirectoryFilter003=%windir%\Installer\*
InvDirectoryFilter004=%windir%\system32\ccm\cache\*
InvDirectoryFilter005=%windir%\WinSxS\*
InvDirectoryFilter006=%windir%\ServicePackFiles\i386\*
InvDirectoryFilter007=%ProgramData%\appv\*
InvDirectoryFilter008=%ProgramData%\app-v\*
InvDirectoryFilter009=%APPDATA%\*
InvDirectoryFilter010=%LOCALAPPDATA%\*
InvDirectoryFilter011=*\AppData\LocalLow\*
```

Columbus Inventory Agent Installation

Es wird empfohlen den Inventory Agent im regulären Programmverzeichnis z.B. "C:\Program Files (x86)\Columbus\InventoryAgent". zu installieren.

Der ColumbusInventoryAgent.exe unterstützt die folgenden Kommandozeilenbefehle:

Befehl	Funktion
/Install	Installiert den Agent als Dienst
/Uninstall	Entfernt den Dienst
/Silent	Stille Operation für die Verwendung in Skripten.

Beispiele

Installation des Inventory Agents als Dienst im stillen Modus:

```
C:\Program Files (x86)\Columbus\InventoryAgent\ColumbusInventoryAgent.exe /Install /Silent
```

Deinstallation des Dienstes

```
C:\Program Files (x86)\Columbus\InventoryAgent\ColumbusInventoryAgent.exe /Uninstall
```

Nach dem ersten Start des Dienstes werden die Informationen aus der ColumbusInventoryAgent.cfg eingelesen und in der Registry abgelegt. Im Anschluss wird die Datei gelöscht.

Columbus Inventory Agent Aktualisierung

Wenn die automatische Aktualisierung aktiviert ist (Standardeinstellung, Autoupdate aktiviert), (siehe Columbus Inventory Agent Configuration) prüft der Agent alle 24h nach einem Update auf Server.

Die Dateien für das automatische Update werden im "Updates_Agent" Ordner hinterlegt. Das Verzeichnis ist dasselbe in das der Agent seine Inventarinformationen überträgt. Der Ordner "Updates_Agent" muss eine Zip Datei mit dem Namen "Updates_Agent.Zip". enthalten. Innerhalb dieser Zip Datei können/müssen die folgenden Dateien enthalten sein.

- ColumbusInventoryAgentUpdater.exe (zwingend)
- ColumbusInventoryAgent.exe (optional)
- ColumbusInventoryAgent.cfg (optional)

Wenn entweder die "ColumbusInventoryAgent.exe" oder "ColumbusInventoryAgent.cfg" Datei in der Zip Datei existieren, wird der Agent ein Update durch diese Datei(en) versuchen. Im Falle der ColumbusInventoryAgent.cfg wird die (ggfs. geänderte) Konfiguration eingelesen.

Während des Updates des Data Collectors, wird die beschriebene Zip Datei automatisch durch das Setup aktualisiert.

Columbus Inventory Agent Scanzeitpunkt zurücksetzen

Gelegentlich kann es vorkommen, dass der Zeitpunkt des letzten Scans zurückgesetzt werden muss um erneut einen Scan auszuführen.

Das wird erreicht indem der folgende Key gelöscht wird:

```
Key: HKEY_LOCAL_MACHINE\SOFTWARE\<Wow6432Node>\BrainWare\Columbus\7\InvAgent
Value: LastRun
```

Columbus Inventory Agent Metering

Der Inventory Agent erfasst keine Meteringdaten

Falls der Inventory Agent keine Meteringdaten erfasst sollte die Konfiguration auf der Maschine überprüft werden

Die Einstellung wird mit dem folgenden Registry Key gemacht:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\BrainWare\Columbus\7\InvAgent\Config
```

Der REG_SZ Wert "InvFunction" muss existieren, sollte der Wert einen anderen Eintrag als "2" enthalten, ist das Metering nicht aktiv.

5.1.4 Columbus Inventory Agent MSI

Der Inventory Agent ist zusätzlich als MSI Paket verfügbar.

Columbus Inventory Agent MSI Installation

Obwohl das ColumbusInventoryAgent.msi eine grafische Benutzeroberfläche zur Verfügung stellt, ist das MSI zur Installation im Unbeaufsichtigtem Modus vorgesehen. Es wird empfohlen das MSI mit einem Standard Softwareverteilungssystem wie z.B. Columbus (ggfs. auch SCCM, LANDESK, Altiris etc.) oder durch ein Group Policy Object (GPO) zu verteilen.

Das ColumbusInventoryAgent.msi bietet folgende Public Properties:

Name	Values	Description
INVOTB_PORT	1685	Port zu dem der Agent die Ergebnisse übermittelt.

Name	Values	Description
INVOTB_HOST		OTB Server der die Scan Ergebnisse entgegen nimmt.
AUTOSTARTSERVICE	0 | 1	Inventory Agent Dienst nach der Installation automatisch starten. 1 = automatisch starten (Standardeinstellung) 0 = nicht automatisch starten
SCANNERFUNCTION	0 = HW, SW, Inv. Items 1 = HW, SW, Inv. Items, File Scan 2 = HW, SW, Inv. Items, File Scan, Metering	Einstellung der Scan Funktion. Default Wert falls nichts anderes angegeben wird ist: 2
INVUPDATEENGINE	0 | 1	Automatisches Update der Scanner Add-on DLLs Standardwert: 1
INVUPDATEAGENT	0 | 1	Automatisches Update der Agent Komponenten Standardwert: 1
INVSCANSTARTPERIOD	daily | weekly | monthly	Häufigkeit des Scans Standardwert: daily
INVSCANSTARTDELAY	0 - 100	Zufällige Verzögerung nach Start des Service bis zum Beginn des Scans in Minuten Standardwert: 0 (keine Verzögerung)
INVDRIVES	CDE	Lokale Laufwerke die gescannt werden sollen. Wenn leer, werden alle lokalen Laufwerke gescannt.

Beispiele für die Installation

Installation des Inventory Agent, automatischer Start des Dienstes, nur Inventory, im Stillen Modus:

```
msiexec /i "ColumbusInventoryAgent.msi" /qn INVOTB_HOST="<FQDN of Data Collector machine>" INVOTB_PORT="24999" SCANNERFUNCTION=1
/L*V InventoryAgentInstaller.log
```

Installation des Inventory Agent, automatischer Start des Dienstes, Inventory und Metering, im Stillen Modus:

```
msiexec /i "ColumbusInventoryAgent.msi" /qn INVOTB_HOST="<FQDN of Data Collector machine>" INVOTB_PORT="24999" /L*V
InventoryAgentInstaller.log
```

Weitere Informationen für die Steuerung des MSI Installation können durch die Eingabe von "msiexec.exe /?" in einer Kommandozeile angezeigt werden.

Achtung	Um das Metering zu aktivieren, muss entweder SCANNERFUNCTION aus dem MSI Aufruf weggelassen werden, oder es muss SCANNERFUNCTION=2 verwendet werden, alle anderen Werte deaktivieren das Metering.

Columbus Inventory Agent MSI Quelldatei

Das MSI befindet sich im Installationsverzeichnis (Abbildung - Installationspfad (siehe Seite 6)) im Unterordner "ColumbusInventoryAgent-MSI".

Verteilung per GPO (Schritt für Schritt)

Dies ist eine kurze Schritt für Schritt Anleitung wie das MSI mittels einer GPO verteilt werden kann.

Notiz	Sämtliche Pfade und Konfigurationen müssen an die gegebene Umgebung angepasst werden.

Erstellen einer MST Datei für das MSI

1. Herunterladen des Installers für das Windows Software Development Kit (SDK) for Windows 8.1 von https://msdn.microsoft.com/en-us/windows/desktop/bg162891.aspx und ausführen der Installation.
   Im Feature Dialog "Select the features you want to install" Auswahl der "MSI Tools"

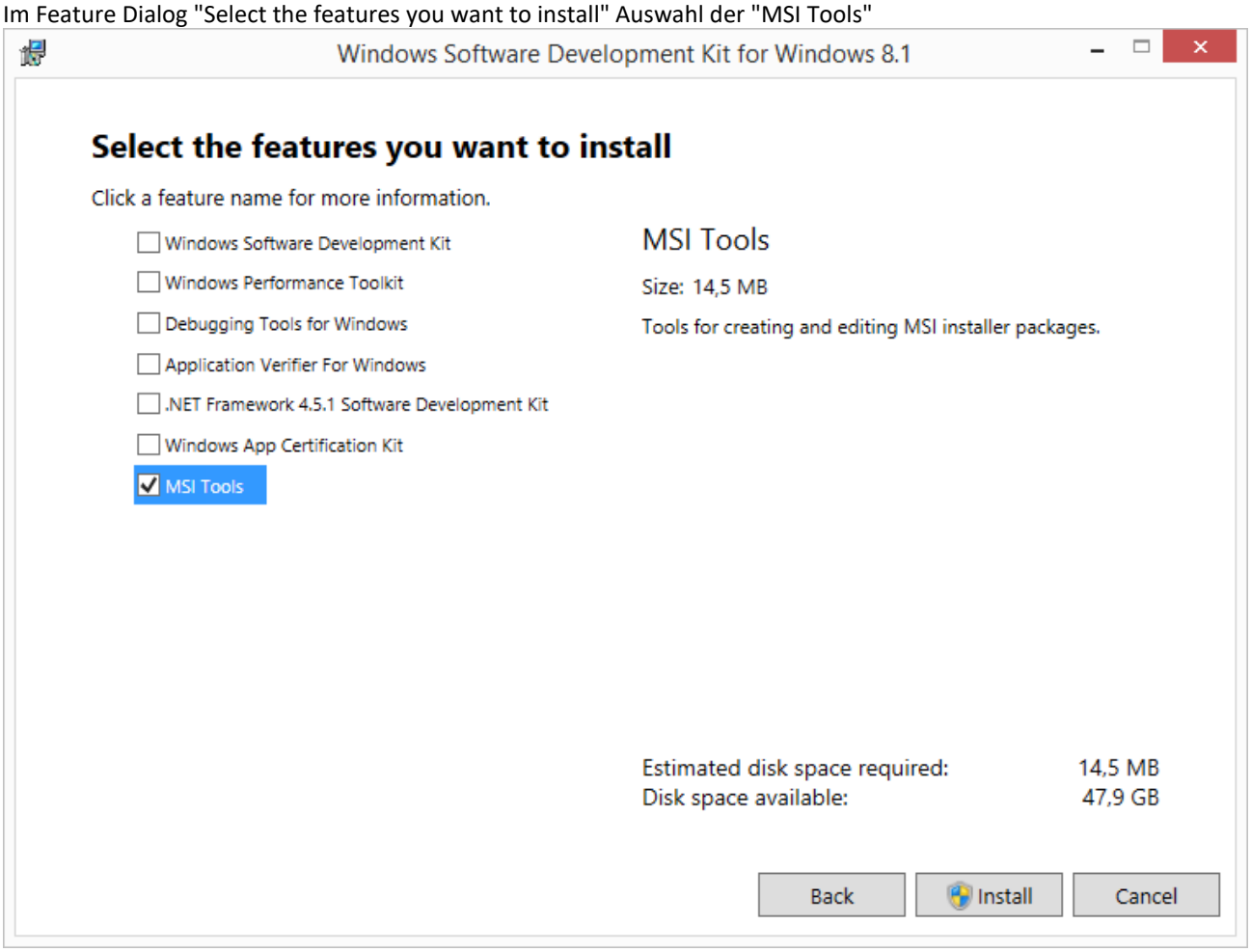


Abbildung - Select Features to install

2. Nachdem die Installation durchgeführt wurde, das ORCA*.msi aus C:\Program Files (x86)\Windows Kits\8.1\bin\x86 ausführen und Installieren.
3. Orca öffnen und das ColumbusInventoryAgent.msi laden
   "Transform" und "New Transform" wählen.

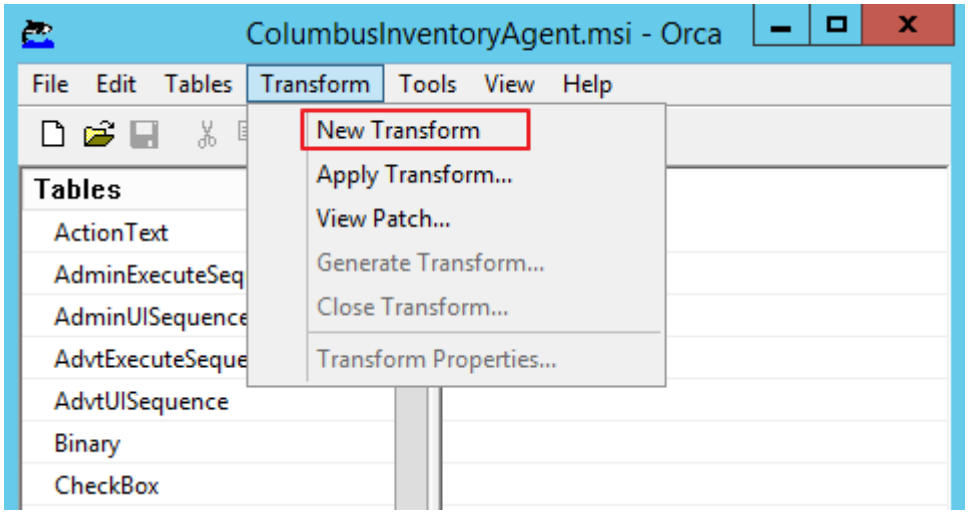


Abbildung - New Transform

4. Die Tabelle "Property" öffnen und die Einstellungen für
   - INVOTB_PORT
   - INVOTB_SERVER
   - AUTOSTARTSERVICE
   - SCANNERFUNCTION

   setzen.

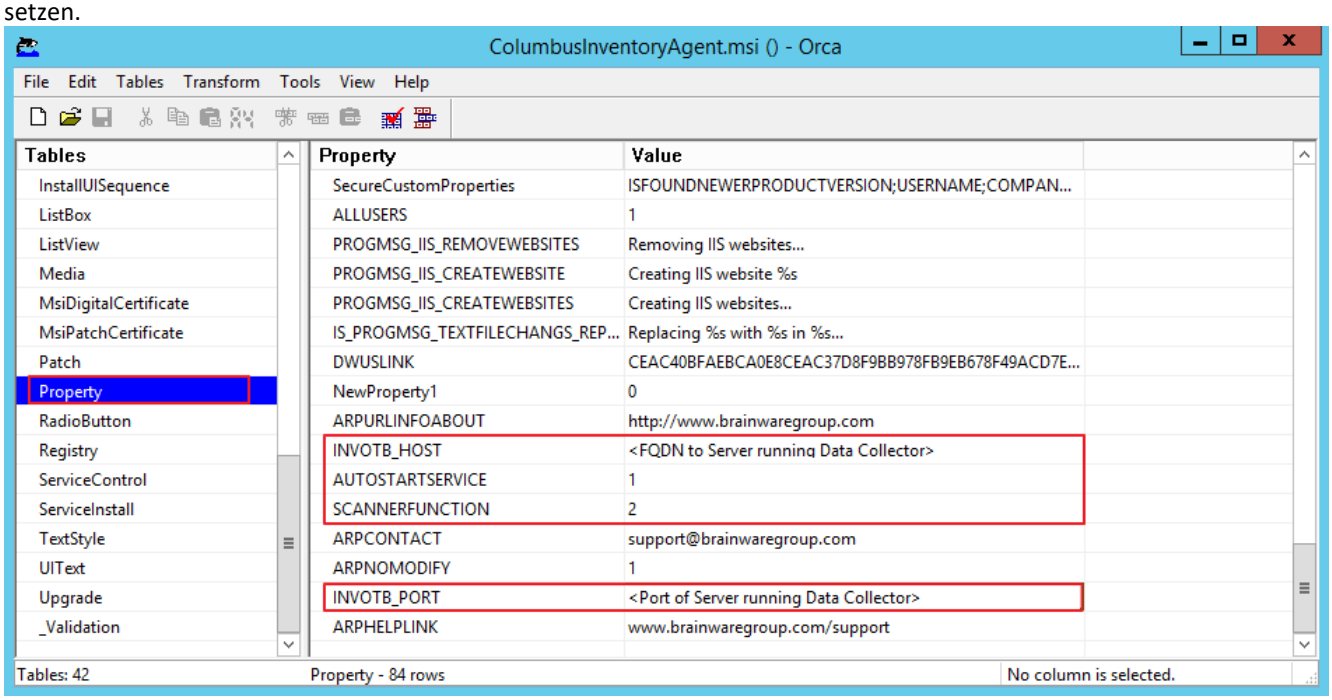


Abbildung - Property table

Details zu den benötigten Einstellungen wurden in den vorherigen Kapiteln beschrieben

5. "Transform" > "Generate Transform" wählen und dann den Pfad wählen in dem die MST Datei gesichert werden soll. Jetzt kann diese MST Datei zusammen mit dem Agent MSI für eine Installation per GPO verwendet werden.

Erstellen der GPO

1. Kopieren von ColumbusInventoryAgent.msi/.mst auf eine Netzwerkfreigabe. Die Berechtigungen müssen so gesetzt sein, dass alle Benutzer und Computer **Lesend** auf die Freigabe zugreifen können.

2. In "Group Policy Management", den Container wählen auf dem die Verteilung stattfinden soll (eine Site, eine Domain, oder eine Organisationseinheit (OU)), per Rechtsklick "Create a GPO in this domain, and Link it here..." wählen.

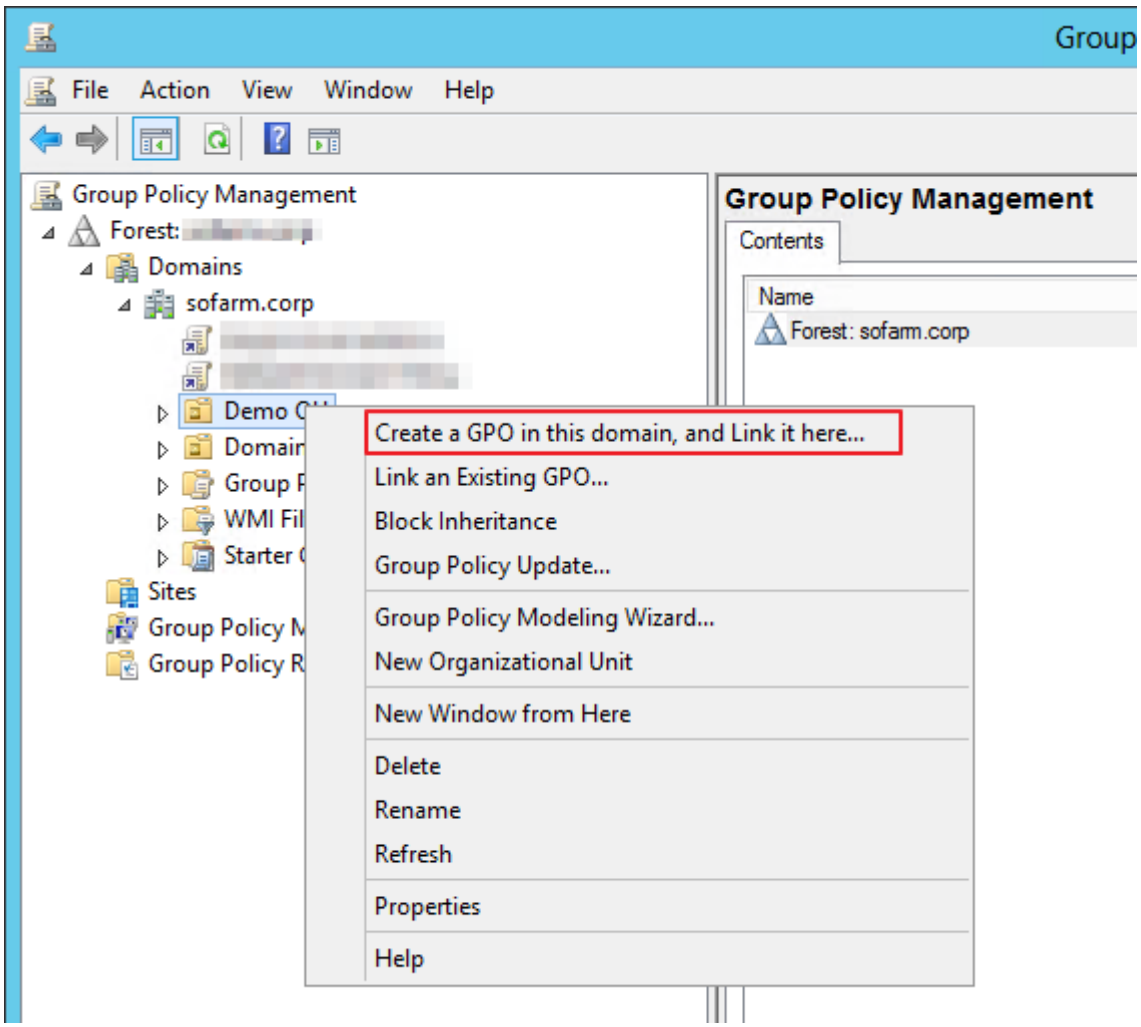


Abbildung - Create GPO...

3. Der GPO einen Namen geben, in diesem Beispiel ist das "Deploy Columbus Inventory Agent MSI"

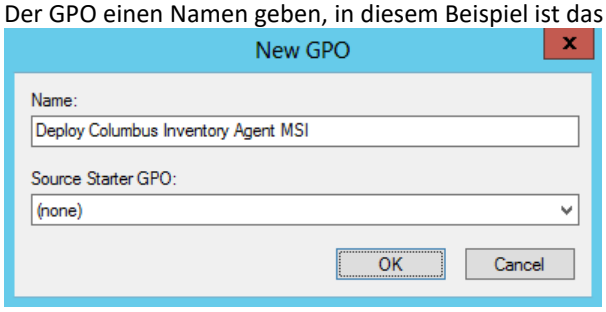


Abbildung - New GPO

4. Rückkehr zum "Group Policy Management" und editieren der neu angelegten GPO.

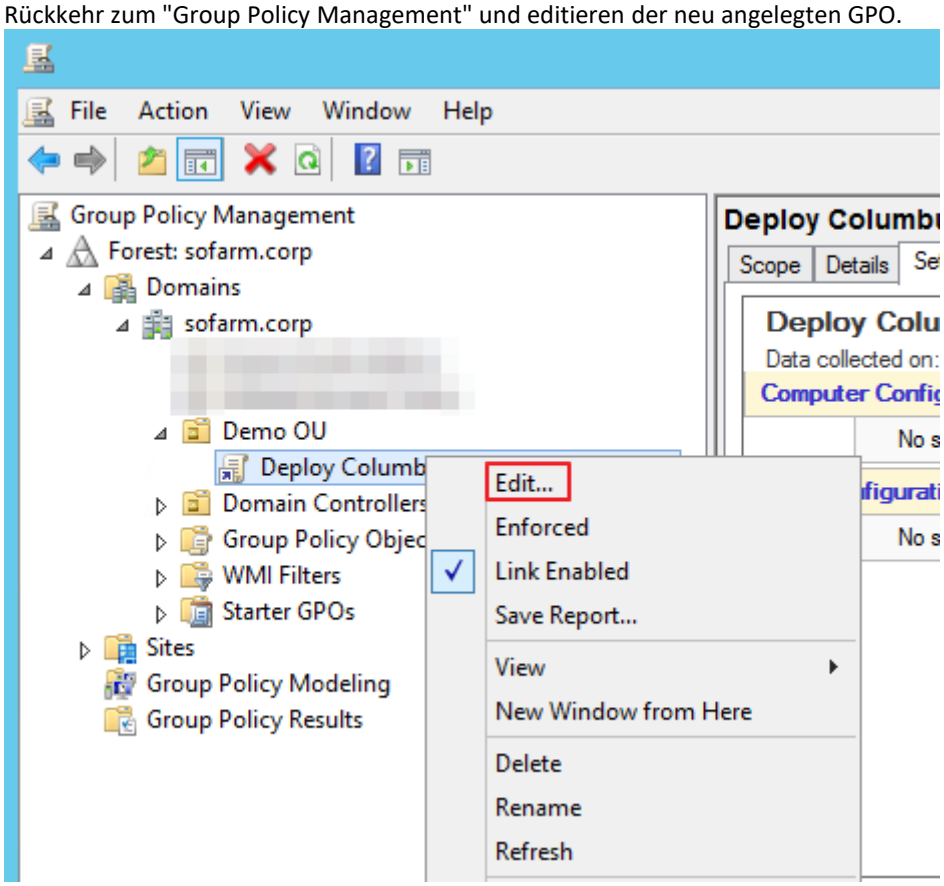


Abbildung - Edit GPO

5. Wenn die Anwendung per Benutzer verteilt werden soll, Auswahl von **User Configuration\Software Settings** im **Group Policy Management Editor**, Rechts-Klick **Software Installation**, Auswahl von **New**, und dann **Package** auswählen.
Wenn die Anwendung pro Computer Konto verteilt werden soll, Auswahl von **Computer Configuration\Software Settings** in der **GPO**, Rechts-Klick **Software Installation**, Auswahl von **New**, und dann **Package** auswählen.
Siehe hier::

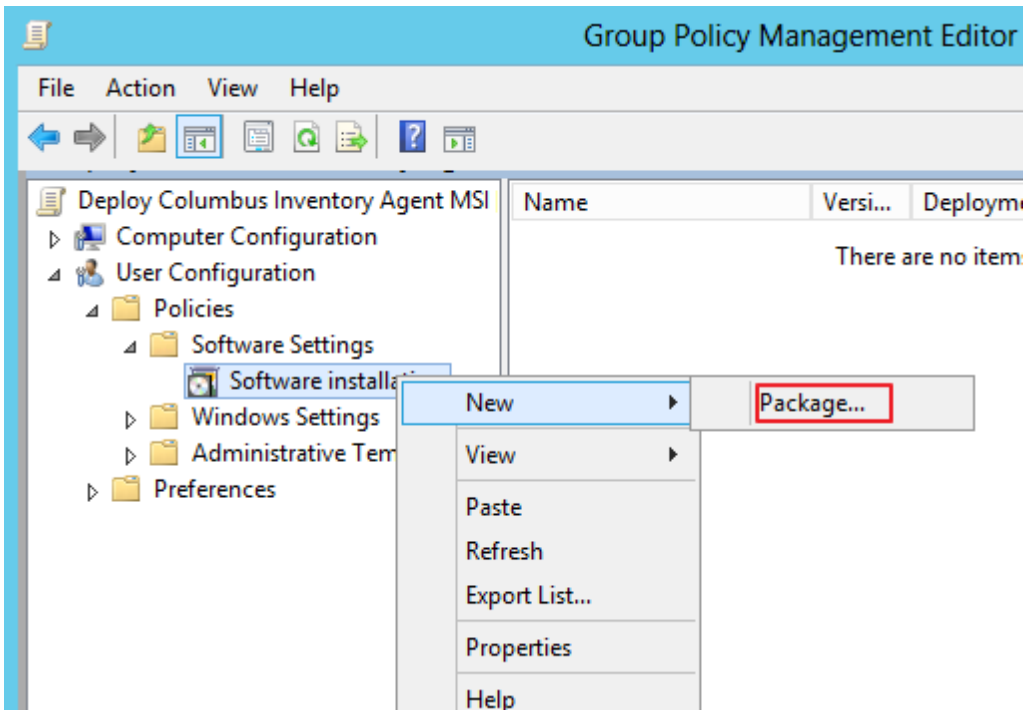


Abbildung - New Package

6. MSI Paket auswählen und dann **Advanced** als Verteilmethode auswählen (siehe unten)

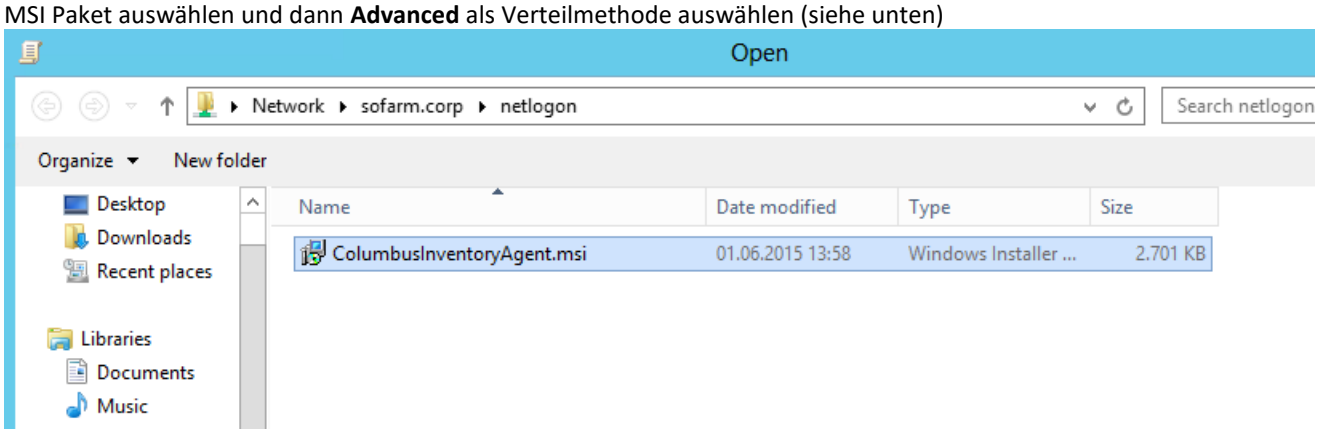


Abbildung - Choose MSI

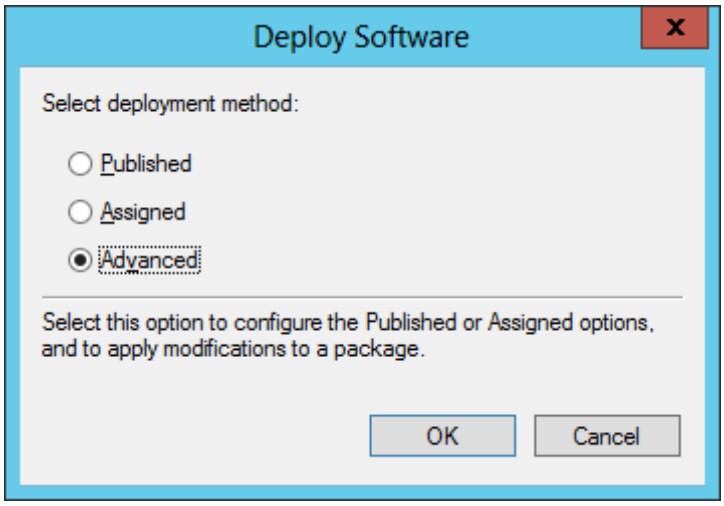


Abbildung - Deployment Method

7. Auf dem **Deployment** Reiter, den Deployment type und die Deployment wie unten zu sehen markieren (die Deployment Type/Options Einstellungen können ggfs. abweichen). In diesem Beispiel wird **Assigned** als deployment type verwendet..

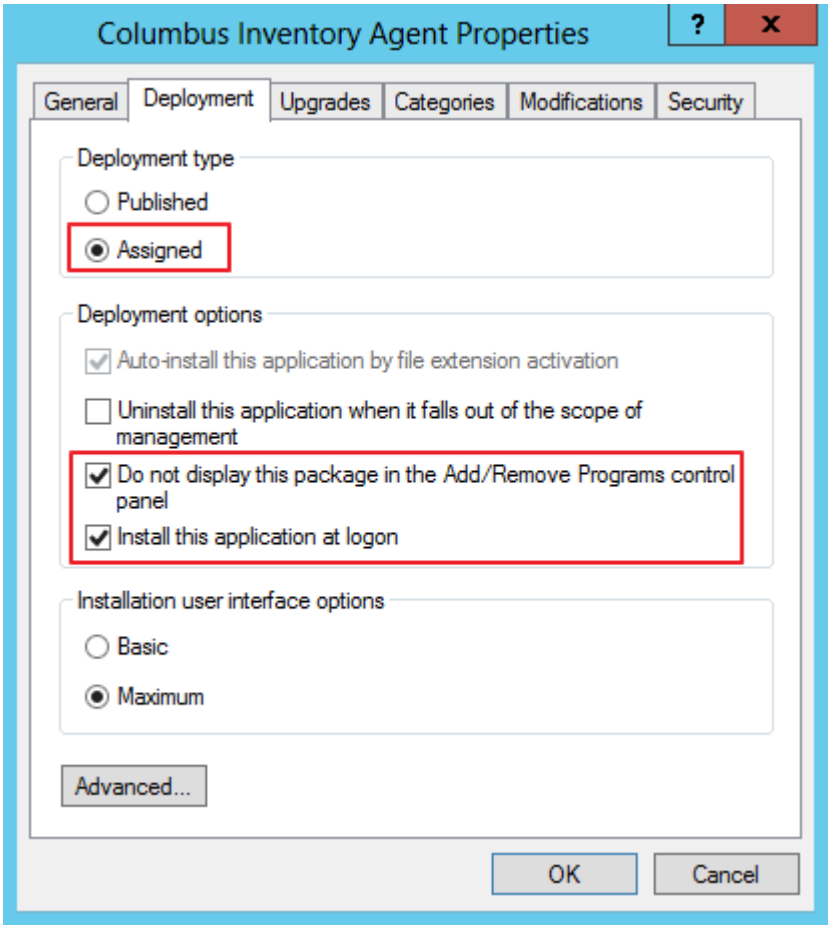


Abbildung - Deployment

8. Auf dem **Modifications** Reiter, die MST Datei auswählen (die die Installation anpasst):

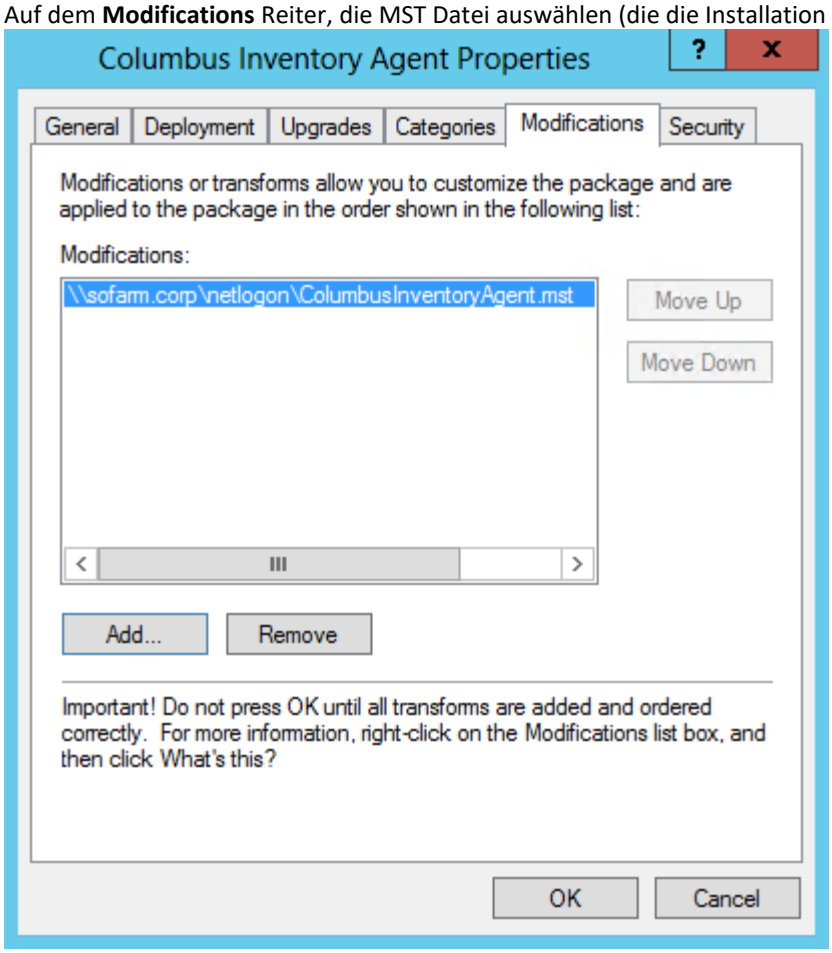


Abbildung - Modifications

9. Mit **OK** die Einrichtung bestätigen.

Achtung	Dies ist nur ein Beispiel, abhängig von den Vorgaben in der jeweiligen Organisation weicht das Beispiel ggfs. stark vom gelebten Prozess ab.

5.1.5 Columbus Inventory Scanner

.

Columbus Inventory Scanner Quelldateien

Der Inventory Scanner liegt im Installationsverzeichnis (Abbildung - Installationspfad (siehe Seite 6)) im Unterordner "ColumbusInventoryScanner", das Verzeichnis enthält die folgenden Dateien:

- ColumbusInventoryScanner.cfg
- ColumbusInventoryScanner.exe
- libeay32.dll
- ssleay32.dll
- StartReset.cmd (sollte nicht verteilt werden, wird der Scanner mit dieser Datei gestartet, wird automatisch das letzte Scan Datum zurückgesetzt.)

Columbus Inventory Scanner Konfiguration

Der Scanner wird durch das Ausführen der ColumbusInventoryScanner.exe gestartet, die Konfiguration wird durch Einträge in der ColumbusInventoryScanner.cfg bestimmt.

```
[Scanner]
InvExtensions=.EXE

[Transmitter]
InvOTB_Host=hostname.domain.suffix
InvOTB_Port=24786
```

Nach der Installation des Data Collectors ist der Scanner bereits vorkonfiguriert und bereit zur Verwendung.

Sektion	Parameter	Standardwert (falls leer)	Mögliche Werte / Beschreibung
Scanner	InvFunction	2	0 = HW, SW, Inv. Items 1 = HW, SW, Inv. Items, File Scan 2 = HW, SW, Inv. Items, File Scan, Metering
Scanner	InvDrives	Alle lokalen Laufwerke	CDE (bedeutet die Laufwerke C:, D: und E:)
Scanner	InvExtensions	.EXE	Liste der Dateiendungen, für die detaillierte Informationen während einem File Scan erhoben werden.
Scanner	InvExportPath	%ProgramData%\Columbus	Lokaler Pfad in dem die Scan Ergebnisse vorgehalten werden bevor sie übermittelt werden. Z.B. %temp% oder %_ExePath%
Scanner	InvUpdateEngine	1	Automatisches Update der Scan Signaturen (Scanner Addon DLLs) 0 = Abgeschaltet 1 = Eingeschaltet
Scanner	InvScanStartPeriod	daily	Häufigkeit des Scans daily = alle 24 h weekly = einmal pro Woche monthly = einmal pro Monat
Scanner	InvScanStartDelay	0	Startverzögerung, Scan beginnt in angegebenen Zeitraum NACH dem Start des Service n Minuten (0-100)
Scanner	InvLastObject	0	1 = Erhebung der Benutzerinformationen • LastLoggedOnUser • LastLoggedOnSAMUser • LastLoggedOnUserSID
Scanner	InvNetwork	0	1 = Erhebung der Netzwerkinformationen • MAC1 • MAC2 • MAC3 • MAC4 • IPAddressV4 • IPAddressV6
Scanner	InvLicensee	0	1 = Erhebung der Lizenzinformationen des OS • OS.System.RegisteredUser • OS.System.Organization

Sektion	Parameter	Standardwert (falls leer)	Mögliche Werte / Beschreibung
			• OS.System.ProductKey
Transmitter	InvTransmissionMode	3	Übertragungsmethode der Ergebnisse 0 = Keine Übertragung, Offline Modus 1 = FTP 3 = OTB
Transmitter	InvOTB_Host		FQDN der Maschine die die Ergebnisse entgegen nimmt.
Transmitter	InvOTB_Port	24786	Port der Maschine die die Ergebnisse entgegen nimmt.
Transmitter	InvFTP_Host		Hostname des FTP Servers
Transmitter	InvFTP_Port		Port des FTP Servers
Transmitter	InvFTP_User		FTP-Server Authentifizierung, Benutzer (Falls leer, wird Anonymous verwendet)
Transmitter	InvFTP_Password		FTP-Server Authentifizierung, Passwort (Verschlüsselung mit cryptit.exe)
DirectoryFilter	InvDirectoryFilter001 ... InvDirectoryFilter999		Filter für Verzeichnisse die vom Scan ausgeschlossen werden sollen Akzeptiert Windows Variablen und feste Pfade z.B. %windir%* oder D:\Data*

Standardfilter

Der Scanner wird mit einigen Standardfiltern vorbelegt:

```
[DirectoryFilter]
InvDirectoryFilter000=*\microsoft system center 2012\dpm\dpm\volumes\*
InvDirectoryFilter001=%windir%\$*_$\*
InvDirectoryFilter002=%windir%\*\$*_$\*
InvDirectoryFilter003=%windir%\Installer\*
InvDirectoryFilter004=%windir%\system32\ccm\cache\*
InvDirectoryFilter005=%windir%\WinSxS\*
InvDirectoryFilter006=%windir%\ServicePackFiles\i386\*
InvDirectoryFilter007=%ProgramData%\appv\*
InvDirectoryFilter008=%ProgramData%\app-v\*
InvDirectoryFilter009=%APPDATA%\*
InvDirectoryFilter010=%LOCALAPPDATA%\*
InvDirectoryFilter011=*\AppData\LocalLow\*
```

Columbus Inventory Scanner Ausführung

Zur Ausführung des Columbus Inventory Scanners benötigt man lediglich Lesezugriff auf die Dateien die in Inventory Scanner Location genannt sind, die ColumbusInventoryScanner.exe ist auszuführen.

Anmeldeskript

Der Inventory Scanner kann durch unterschiedliche Automatismen ausgeführt werden; eine davon ist ein Logon Skript. Wenn die Ausführung durch ein Login Skript erfolgt, sollten die Dateien des Scanners auf einer allgemein verfügbaren

Freigabe abgelegt werden, das alle Maschinen/Benutzer erreichen können, z.B. die NETLOGON Freigabe. Zur Ausführung muss dann nur noch die ColumbusInventoryScanner.exe ins das Logon Skript eingebunden werden.

Beispiel für eine Einbindung ins Logon Skript:

```
Start "\\domain.local\Netlogon\InventoryScanner\ColumbusInventoryScanner.exe"
```

Wichtig	Der Inventory Scanner sollte so ausgeführt werden, dass er die weitere Verarbeitung des Logon Skripts nicht behindert. Sollte dies nicht geschehen, kann es sein das die Ausführung für die Dauer des Scans blockiert wird, dies führt ggfs. zu verlängerten Anmeldezeiten.

Software Verteilung

Als Software Verteilung kann eine beliebige Software zum Einsatz kommen. Es müssen lediglich die unten genannten Dateien in ein Paket verpackt und ausgeführt werden. Alternativ kann auch durch eine Software Verteilung ein Aufruf von einer Netzwerkfreigabe erfolgen.

- ColumbusInventoryScanner.exe
- ColumbusInventoryScanner.cfg

GPO

Der Mechanismus zur Ausführung des Inventory Scanners ist der gleiche der bereits im Kapitel für den Agenten beschrieben ist. Ein Beispiel für die GPO Einstellung ist im Screenshot zu sehen.

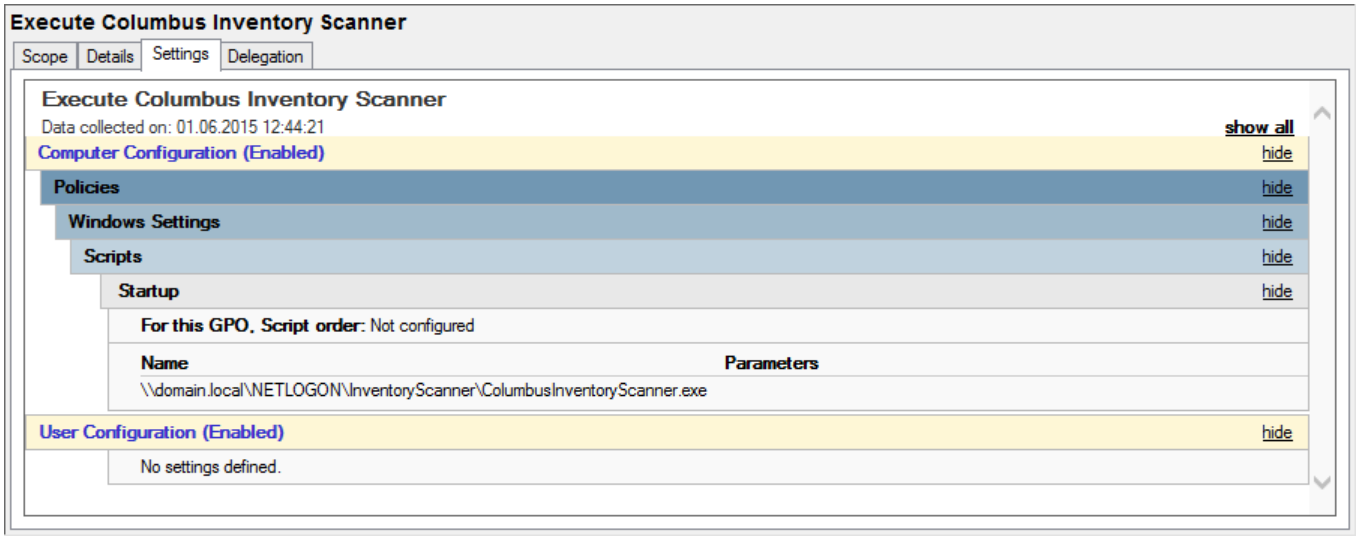


Abbildung - GPO

Notiz	Group Policies werden synchron ausgeführt, d.h. die Weiterverarbeitung ist blockiert solange der Scanner arbeitet. Abhängig von der Hardware der Maschine (Größe und Anzahl Festplatten) sowie der generellen Leistung kann dies ein paar Minuten dauern.

Ein Weg diesen Bottleneck zu umgehen, ist die Verwendung des Tools PSEXEC von Sysinternals.

Die Psexec.exe kann im gleichen Verzeichnis wie die der Inventory Scanner angelegt und in einem Batch File namens ColumbusInventoryScanner.cmd aufgerufen werden. Die Batchdatei hat dann Beispielsweise den folgenden Inhalt:

```
psexec /accepteula -d %~dp0.\ColumbusInventoryScanner.exe
```

Dieses Vorgehen startet den Prozess asynchron, das Tool ist kostenlos bei Microsoft verfügbar unter: http://technet.microsoft.com/en-us/sysinternals/bb897553.aspx

Scan vom USB Stick

In Umgebungen ohne Netzwerkverbindung kann es nötig sein einen Scan durchzuführen ohne das Ergebnis über das Netzwerk zu übermitteln. Hier kann ein USB Stick zum Einsatz kommen.

Damit dies funktioniert muss der Inventory Scanner so konfiguriert werden, das er die Scan Ergebnisse nicht übermittelt und am besten noch auf dem USB Stick ablegt. Die folgenden Einträge in die ColumbusInventoryScanner.cfg machen dies möglich:

```
[Scanner]
InvExportPath="%_ExePath%"
[Transmitter]
InvTransmissionMode=0
```

Informationen zur Einbindung der Addon DLLs werden im Kapitel Advanced Inventory with Scanner Add-on DLLs beschrieben.

Übermitteln der Scan Ergebnisse

Nachdem die Ergebnisse auf dem USB Stick gesammelt wurden, müssen Sie in den "ScanResults" Ordner auf der Maschine auf der der Data Collector installiert ist kopiert werden. Der Ordner "ScanResults" ist ein Unterordner des Datenordners der während der Installation des Data Collectors angeben wurde.

Der Pfad kann auch aus der Datei SpiderDataCollector.cfg Sektion "[OTBServer]" Variable "DataDirectory" ermittelt werden. Die SpiderDataCollector.cfg befindet sich im Installationsverzeichnis des Data Collectors.

Achtung	Das Setup erstellt einen Ordner mit einem für den USB Scan vorkonfigurierten Scanner. Es befindet sich im Installationsverzeichnis des Data Collectors und heißt: "ColumbusInventoryScanner-USB"

5.1.6 Columbus Inventory Scanner Scanzeitpunkt zurücksetzen

Gelegentlich kann es vorkommen, dass der Zeitpunkt des letzten Scans zurückgesetzt werden muss um erneut einen Scan auszuführen.

Das wird erreicht indem der folgende Key gelöscht wird:

Benutzer: System:

```
Key: HKEY_USERS\S-1-5-18\Software\BrainWare\Columbus\7\InvScanner
Value: LastRun
```

Angemeldeter Benutzer:

```
Key: HKEY_CURRENT_USER\Software\BrainWare\Columbus\7\InvScanner
Value: LastRun
```

5.1.7 Erhobene Hardware Informationen

Element	Beschreibung	Beispiel

Element	Beschreibung	Beispiel
DomainName	Voll qualifizierter Domänenname	stark.industries.local
HostName	Hostname	WRK-P-TOST001
Manufacturer	Hersteller der Hardware	Dell Inc.
Model	Computer Modell	OptiPlex 7010
MAC1	1. MAC Adresse	F8-B1-56-A3-BE-2F
MAC2	2. MAC Adresse	
MAC3	3. MAC Adresse	
MAC4	4. MAC Adresse	
Serial	Seriennummer	50U8AA2
OSClass	Server, Workstation etc.	Client
DeviceChassis	Notebook, Server etc.	Mini Tower
ProcessorManufacturer	Hersteller des Prozessors	Intel
ProcessorType	Prozessortyp	Core i7-3770
ProcessorSpeed	Prozessorgeschwindigkeit	3400
CPUCount	Anzahl der physikalischen CPUs	1
CPUCoreCount	Anzahl der Cores (Summe aller Cores, aller Prozessoren)	4
CPULogicalCount	Anzahl logischer CPUS (Summe aller Cores, aller Prozessoren)	8
UUID	UUID (wohlformatiert)	C1FB8C42-E7F7-422E-9211-757E3BFD82F5
InventorySource	Name und Version des Inventartools	ColumbusInventoryAgent.exe 7.4.0.131
ScanDate	Zeitpunkt an dem der Scan stattgefunden hat.	2014-03-31T10:03:32
DiskTotalMB	Summe des Speicherplatzes aller im System verbauten Festplatten	238472
DiskFreeMB	Summe des freien Speicherplatzes aller im System verbauten Festplatten.	189312
GraphicAdapter	Name der Grafikkarte	AMD Radeon HD 7470
GraphicMemory	Speichergröße der Grafikkarte in MB	1024
MemoryMB	Größe des Systemspeichers in MB	16338
IPAddressV4	Aktuelle IPv4 Adresse des Systems	10.10.20.30
IPAddressV6	Aktuelle IPv6 Adresse des Systems	fe80::d8a9:dd4c:619b:ef5
CPUArchitecture	CPU Architektur	amd64
OSCaption	Name des Betriebssystems	Microsoft Windows 8.1 Enterprise
DomainNetBIOS	NetBIOS Domänenname	STARKINDUSTRIES
LastLoggedOnUser	Zuletzt angemeldeter Benutzer	STARKINDUSTRIES\Tony.Stark
BIOSVendor	Hersteller des BIOS	Dell Inc.
BIOSVersion	Version des BIOS	A16
BIOSDate	Datum des BIOS	09.09.2013
URN	<für zukünftige Verwendung>	
Class	<für zukünftige Verwendung>	
ComputerHomePath	<für zukünftige Verwendung>	

Element	Beschreibung	Beispiel
LastLoggedOnSAMUser	SAM Account Name des angemeldeten Benutzers	STARKINDUSTRIES\Tony.Stark
LastLoggedOnUserSID	SID des angemeldeten Benutzers	S-1-5-21-3427917592-4004333369-2915694803-2802

5.1.8 Unterschiede Scanner / Agent

Die folgende Tabelle zeigt die wichtigsten Unterschiede zwischen Inventory Scanner und Inventory Agent.

Merkmal	Inventory Scanner	Inventory Agent
Installierbar als Dienst		X
Ausführen durch Login Skript	X	
Automatische Aktualisierung Scanner		X
Automatische Aktualisierung Addon DLLs	X	X
Hardware, Software, Datei Scan	X	X
Software Metering		X
Multi-user support		X
Einsatz auf Terminal Servern		X
Start ohne Benutzer möglich		X
Einsatz bei reisenden Benutzern		X
Offline Benutzung	X	X

5.1.9 Erweiterte Inventarisierung mit den Scanner Add-on DLLs

Für die erweiterte Erkennung von Softwareprodukten (z.B. Editionserkennung von SQL Server, Erkennung embedded OS) können Agent und Scanner DLLs verwenden die zusätzliche Erkennungen integrieren.

Die DLLs sind .Net basierend und verfügbar für den Einsatz unter .NET2 und .NET4

Sowohl Inventory Scanner als auch Inventory Agent, versuchen vor dem Scan die DLLs vom Data Collector zu aktualisieren, wenn die Konfiguration dies nicht untersagt.

Die DLLs (ScannerAddon.Net2.dll und ScannerAddon.Net4.dll) müssen für die Scanning Komponenten im Ordner "Files_Scanner" zur Verfügung gestellt werden. Dieser Ordner ist ein Unterordner des ScanResults Ordners in dem der Data Collector die empfangenen Zip Dateien ablegt. Der Ort des Ordners wird während des Setups abgefragt und in die SpiderDataCollector.cfg, geschrieben (Sektion [OTBServer] Variable "DataDirectory").

Während des Updates des Data Collectors, werden die DLLs automatisch aktualisiert.

Achtung	Beim Einsatz der standalone Version des Inventory Scanners (z.B. auf einem USB Stick) müssen die DLLs im gleichen Ordner wie die ColumbusInventoryScanner.exe gelegt werden.

5.1.10 Zusätzliche Werte aus der Registry erfassen

Wenn zusätzliche Informationen über die Inventory Komponenten ermittelt werden sollen, so können diese Informationen in der Registry abgelegt werden und werden dann durch die Inventory Komponenten erfasst.

```
Key: HKEY_LOCAL_MACHINE\SOFTWARE\<Wow6432Node>\Brainware\Columbus\7\ExternalInventoryData
Value: LastScheduledActionCompleted
```

Unterstützte Typen

- REG_SZ
- REG_DWORD
- REG_QWORD

Beispiel:

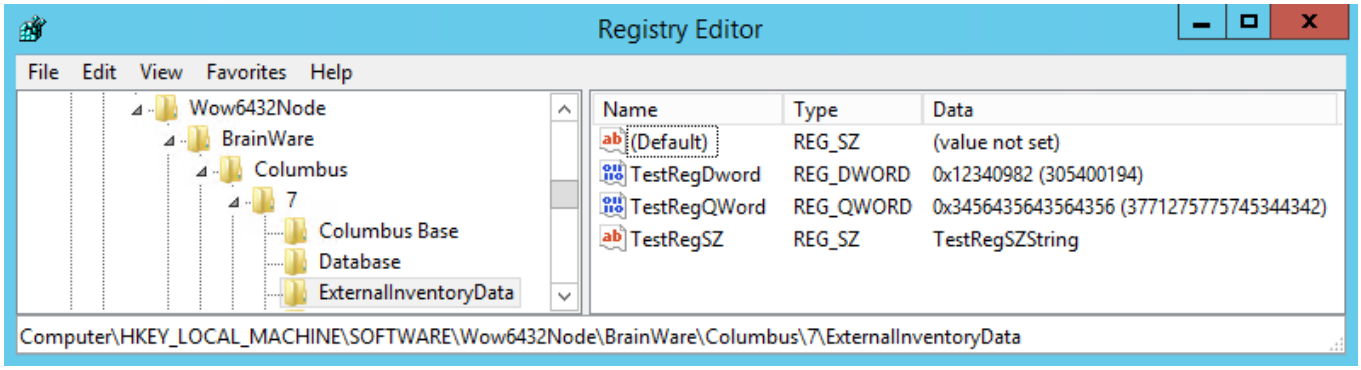


Abbildung - Registry Editor

5.1.11 SSL verschlüsselte Übertragung

Die Kommunikation des Spider Data Collectors kann per SSL verschlüsselt werden (RSA 2048Bit).

Nach der Installation des Data Collectors, liegen die benötigten Dateien in: %ProgramData%\Columbus

Die Verwendung von SSL kann durch die folgenden Registry Einträge ermöglicht und erzwungen werden.

```
Key: HKEY_LOCAL_MACHINE\SOFTWARE\<Wow6432Node>\BrainWare\Columbus\7\
"OTBEncryptionUseSSL"="1"
"OTBEncryptionForceUseSSL"="0"
```

5.2 Mac OS

5.2.1 Columbus Inventory Scanner Quelldateien

Der Columbus Inventory Scanner (CIS) liegt im Installationsverzeichnis (Abbildung - Installationspfad (siehe Seite 6)) im Unterordner "cis" das Verzeichnis enthält die folgenden Dateien:

- cis.prv
- ColumbusInventoryScanner.cfg
- ColumbusInventoryScanner.tar
- setup.sh

5.2.2 Columbus Inventory Scanner Konfiguration

Vor dem ersten Ausführen des CIS, sollte die Konfigurationsdatei angepasst werden. Wichtig ist hier hauptsächlich der Bereich „[Transmitter]“. Mit „InvTransmissionMode“ kann festgelegt werden, wie mit dem Columbus Server kommuniziert wird (1=sftp, 2=ssh, 0=keine Kommunikation)

Außerdem sollte noch festgelegt werden mit welchem System kommuniziert wird. Hierfür sind die Einträge Host, Port, User und Key zu ändern. Sollten die Einträge leer sein, wird der Default Wert verwendet (sofern vorhanden).

Soll der SSH/SFTP Key ausgetauscht werden, muss dieser im Ordner „etc/.ssh“ ersetzt werden. Ein neuer Schlüssel kann nur über die automatische Generierung SFTP Server Konfiguration (siehe Seite 19) erstellt werden.

Ein Beispiel für die Kommunikation über SFTP mit dem Server „sdc-cis“ auf Port „22“, dem User „cis“ und der Key Authentifizierung sähe so aus:

```
InvTransmissionMode=1
InvTransmissionHost1=sdc-cis
InvTransmissionPort = 22
InvTransmissionUser = cis
InvTransmissionKey = /Pfad/zu/privateKey
```

Achtung:	Die genannten Einstellungen sind nach der Installation bereits vorkonfiguriert, ebenso wurde ein neuer Schlüssel generiert. Bitte die Dateien verwenden, die im gleichen Verzeichnis wie die setup.sh liegen

Die meisten Einträge der Konfigurationsdatei können auch als Runtime Argument an das Binary übergeben werden. Diese Argumente haben Priorität über die Einträge der Konfigurationsdatei. Für eine Liste der Runtime Argumente kann das entsprechende Binary aus dem „/bin“ Ordner mit dem Argument „--full-help“ ausgeführt werden.

Im Folgenden werden die einzelnen Punkte der „ColumbusInventoryScanner.cfg" – Konfigurationsdatei erklärt. Die Datei liegt im Installationsordner im Unterverzeichnis „etc/"

Sollte die Datei leer oder nicht vorhanden sein, werden die Standardwerte genutzt.

Sektion	Parameter	Standardwert (falls leer)	Mögliche Werte /Beschreibung
Scanner	InvUpdateScanner	1	Update des Binaries 0 = kein Update

Sektion	Parameter	Standardwert (falls leer)	Mögliche Werte /Beschreibung
			1 = Update
Scanner	InvUpdatePackage	1	Update der Abfrage Pakete 0 = kein Update 1 = Update
Scanner	InvScanStartTime	22:00	Uhrzeit der Ausführung hh:mm
Scanner	InvScanStartDelay	0	Verzögerung der Ausführung um n Minuten
Scanner	InvScanStartPeriod	daily	Ausführhäufigkeit daily = täglich weekly = wöchentlich
Scanner	InvPackageExec	$OS	Name des Pakets welches Ausgeführt wird z.B. „macos“
Transmitter	InvTransmissionMode	1	Art der Datenübertragung 0 = keine Übertragung (offline) 1 = sftp 2 = scp
Transmitter	InvTransmissionHost1		Hostname/IP des Zielservers (leer = offline)
Transmitter	InvTransmissionHost2		Optional. Alternativer Zielserver (Hostname/IP)
Transmitter	InvTransmissionHost3		Optional. Alternativer Zielserver (Hostname/IP)
Transmitter	InvTransmissionPort	22	Port des Transmission Hosts
Transmitter	InvTransmissionUser	cis	Benutzername des Transmission Hosts
Transmitter	InvTransmissionKey	etc/.ssh/cis.prv	Private Key für Authentifizierung (ohne Passwort)
Transmitter	InvTransmissionTimeout	60	Zeit für Datenübertragung je Host je Verbindungsversuch in Sekunden
Transmitter	InvTransmissionRetry	3	Anzahl Verbindungsversuche je Host
DirectoryFilter	InvStorage	10	Anzahl der Ergebnispakete die in InvOutput vorgehalten werden
Runtime	InvLogLevel	2	Art der Log Meldungen 0 – 10 0 = nur Fehler 10 = möglichst viel Output
Runtime	InvLogDir	log/	Speicherort für Logfiles
Runtime	InvLogFileName	ColumbusInventoryScanner.log	Name des Logfiles
Runtime	InvLogOutput	0	1 = Log Meldungen zusätzlich auf Stdout 0 = Keine Log Meldungen auf Stdout
Runtime	InvLogSyslog	0	0 = Log Meldungen nur in Logfile 1 = Log Meldungen in syslog 2 = Log Meldungen in Logfile und syslog
Runtime	InvHostname	$HOSTNAME	Hostname des Systems. Unter diesem Namen werden die Ergebnisfiles in InvOutput gespeichert

Sektion	Parameter	Standardwert (falls leer)	Mögliche Werte /Beschreibung
Runtime	InvPlatform	$OS	Art der Befehle die aus dem Paket ausgeführt werden z.B. „macos“
Runtime	InvOutput	output/	Ablageort für Ergebnisdaten
Runtime	InvTimeout	60	Zeit je Befehl aus dem ausgeführten Paket in Sekunden

5.2.3 Columbus Inventory Scanner Installation

Der Columbus Inventory Scanner wird mit dem „setup.sh" Skript installiert. Der Aufruf muss als administrativer Benutzer erfolgen.

Wichtig:	Die Datei setup.sh muss vor der Installation ausführbar gemacht werden.

Ein Beispielaufruf für die Installation mit den vorkonfigurierten Einstellungen sieht folgendermaßen aus:

```
sudo ./setup.sh -c -k --headless
```

Mit diesem Aufruf wird das tar-Archiv in den Standardordner „/opt/ColumbusInventoryScanner/" entpackt. Das Skript fragt vor Ausführung, ob die Einstellungen so in Ordnung sind. Während der Installation informiert es über den Installationsfortschritt.

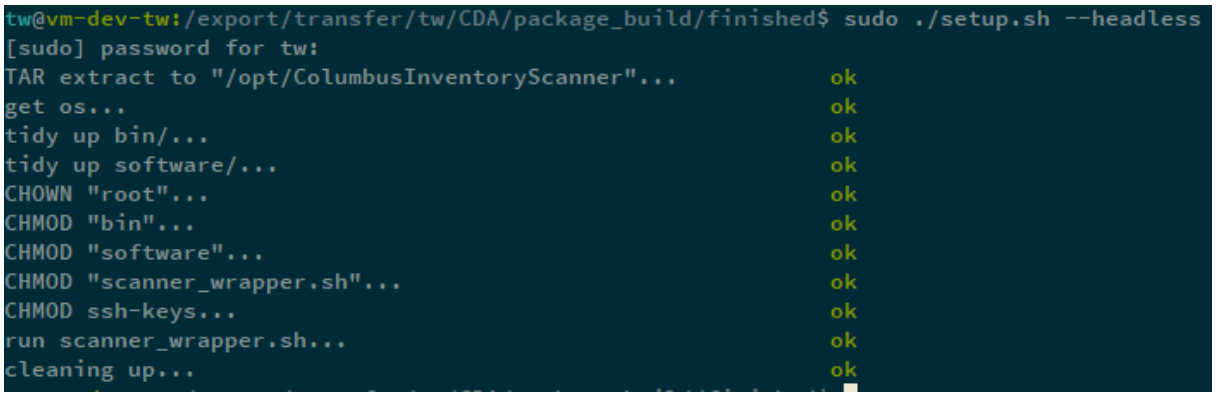


Abbildung - Parameter der setup.sh

Option	Parameter	Funktion
-a |--archive	<DATEI>	Angabe von Pfad und Dateinamen zum CIS-Archive, wenn nur ein CIS[...].tar.gz gefunden wird, nimmt das Script dieses.
-i | --installdir	<PFAD>	Installationsverzeichnis, der Standardwert ist /opt/ColumbusInventoryScanner
-c |--config	[<DATEI>]	Pfad und Name zur ColumbusInventoryScanner.cfg, die für die Konfiguration verwendet werden soll. Wenn keine Datei angegeben wird, wird die ColumbusInventoryScanner.cfg verwendet, die im gleiche Pfad wie die setup.sh liegt.
-k | --key	[<DATEI>]	Pfad und Dateiname zur cis.prv mit der die mitgelieferte Datei überschrieben wird. Wenn keine Datei angegeben wird, wird die cis.prv verwendet, die im gleichen Verzeichnis wie die setup.sh liegt.
--headless		keine Aufforderung zur Bestätigung an den Benutzer während der Installation
-n | --no-run		Scanner nach Installation nicht ausführen
-h |--help		Anzeige des Hilfetextes

Über das „setup.sh" Skript können noch weitere Optionen mitgegeben werden (s. Abbildung 2: Installationsoptionen). So kann Beispielsweise der Installationsordner mit -i „/anderer/Pfad" geändert werden.

```
RELEASE: 1.0.2

usage: ./setup.sh  [OPTIONS]
    OPTIONS:
    -a|--archive <FILE>     - Provide the path/name of the CIS archive
                              if only one CIS[..].tar.gz is found; script can default to this one
    -i|--installdir <PATH>  - Installationdirectory (defaults to /opt/ColumbusInventoryScanner)
    -c|--config [<FILE>]    - define ColumbusInventoryScanner.cfg file to overwrite existing file
                              if [<FILE>] is empty ./ColumbusInventoryScanner.cfg will be used
    -k|--key [<FILE>]       - define key-file to overwrite existing cis.prv file
                              if [<FILE>] is empty ./cis.prv will be used
    --headless              - no user confirmation
    -n|--no-run             - don't run the scanner after setup
    -h|--help               - this message
```

Abbildung - Installationsbeispiel

Notiz:	mit dem Argument „--headless“ kann eine automatisierte Installation vorgenommen werden. Hierbei muss beachtet werden, dass sich **nur ein** ColumbusInventoryScanner.tar Archiv im gleichen Ordner befindet. Dieses wird installiert und anschließend wird das Binary ausgeführt.

Wichtig:	Es bietet sich an bei der Columbus Inventory Scanner Installation die im Verzeichnis "cis" abgelegte Konfigurations- und Schlüsseldatei zu verwenden. Dazu einfach das „setup.sh“ Skript mit den Argumenten „–c“ und „-k“ aufrufen. Die mitgelieferte Konfigurationsdatei und der SSH Key werden dann mit dem „ColumbusInventoryScanner.cfg“ und „cis.prv“ aus dem aktuellen Ordner ersetzt. [Alternativ ist z.B. der Aufruf „-c /Pfad/zu/neuer/ConfigDatei“ möglich. In diesem Fall ersetzt „ConfigDatei“ die ColumbusInventoryScanner.cfg Datei des Installationsarchivs

Im Installationsordner befinden sich nun (bzw. nach dem ersten Aufruf des scanner_wrapper.sh Skripts) folgende Elemente:

Element	Typ	Funktion
bin/	Ordner	Hier befinden sich die CIS Binaries für UNIX Systeme
etc/	Ordner	Hier befindet sich die Konfigurationsdatei und der SSH Schlüssel
software/	Ordner	Hier befindet sich Hilfssoftware (ssh) und Bibliotheken (ssl)
scanner_wrapper.sh	Skript	Skript zum Ausführen des richtigen Binaries

5.2.4 Columbus Inventory Scanner Ausführung

Standardmäßig wird das Binary über einen Root-Cronjob entsprechend der Konfigurationsdatei gestartet (täglich 22 Uhr).

Das Binary kann mit folgendem Kommando auch manuell ausgeführt werden:

```
sudo .bin/ColumbusInventoryScanner_Darwin_x86_64 –p <PaketName>
```

Der Name muss einem Paket aus dem „package/“ Ordner oder dem „package.tar“ File entsprechen (z.B. „macos“). Sofern die SFTP/SCP Verbindung erfolgreich ist, wird das package.tar vom Server geladen und entpackt. Sollte das runterladen nicht funktionieren, wird das lokale Paket ausgeführt (sofern vorhanden).

Sollte kein Name für ein Paket mitgegeben werden, wird (sofern vorhanden) das angegebene Paket aus der Konfigurationsdatei ausgeführt. Standardmäßig (=leerer Eintrag) wird das Paket, welches dem OS-Namen entspricht (z.B. „macos“) ausgeführt.

Für eine einfachere Handhabung kann auch das Skript „scanner_wrapper.sh“ ausgeführt werden:

```
sudo ./scanner_wrapper.sh
```

Dies bestimmt Automatisch $OS und $Architecture und führt das entsprechende Binary mit den Standardeinstellungen auf. Das Wrapper Skript informiert auf Stdout über seinen Fortschritt

Ablauf

Das Binary durchläuft folgende Stufen

- Lesen der bisherigen Konfigurationsdatei
- Übertragen einer neuen Konfigurationsdatei vom Server
- Neu einlesen der Konfigurationsdatei
- Cron Job anlegen/erneuern
- (eventuell) Updates Binaries & Pakete
- Paket <PaketName> bzw. $OS ausführen
- Ergebnisse verpacken und an Server schicken
- Aufräumen

Bei Erfolgreicher Ausführung ist der Exit Code „0“. Der Verlauf kann im log File im „log/“ Ordner nachvollzogen werden. Auch stehen verschiedene Log-Optionen zur Verfügung, z.B. Wird durch „—LogOutput“ der Verlauf zusätzlich auf Stdout ausgegeben.

Data Center Inventory (Linux / Unix)

Zur Inventarisierung unterschiedlicher Server-Plattformen (Unix/Linux/etc.) stehen Inventarisierungsagenten zur Verfügung, die direkt an die Spider Data Center Appliance angebunden werden können. Diese liefern nicht an den installierten Spider Data Collector.

6.1 Voraussetzungen

In den nachfolgenden Abschnitten wird erklärt, welche Voraussetzungen für die Verwendung des Data Center Inventory gegeben sein müssen.

6.1.1 Begriffsdefinition

Die Scan Engine der Spider Data Center Plattform ermöglicht die Datenabfrage von Serversystemen durch den Spider Data Center Server. Die Datenabfrage kann durch verschiedene Mechanismen erfolgen. In diesem Kapitel wird ausschließlich die Datenabfrage durch den Einsatz des Spider Data Center Agenten beschrieben.

Die Kommunikation zwischen dem Spider Data Center Server und den Spider Data Center Agenten erfolgt über TCP/IP und zwei für das Spider Data Center Protokoll reservierte Ports.

6.1.2 Netzwerk-Ports

Für die Kommunikation des Spider Data Center Systems mit dem Spider Data Center Agenten auf den abzufragenden Serversystemen müssen die **Ports 9616** und **9617** bidirektional freigeschaltet sein. Dies gilt sowohl für Firewall-Systeme im Netz als auch für Firewalls auf den Zielsystemen.

6.1.3 Serversysteme

Zur Installation der Agentenpakete auf den Serversystemen werden Administratorrechte auf diesen Server-systemen benötigt. Lokale Firewalls müssen die **Ports 9616** und **9617** bidirektional freigegeben haben zur Kommunikation mit dem Spider Data Center Server.

6.1.4 UUID-Generator

Der UUID Generator gibt zwei UUIDv4 pro System aus: eine „Generated" und eine „Machine" UUID. Die „Generated" UUID wird vom Programm zufällig erstellt; die „Machine" UUID wird vom System abgefragt.

Sollte keine „Machine" UUID ermittelbar sein, wird „Machine: 00000000-0000-0000-0000-000000000000" ausgegeben.

Um zu verhindern, dass mit jedem Aufruf eine neue UUID generiert wird, wird die generierte UUID in eine Datei „eRunbook.uuid" gespeichert. Vor jedem Generierungslauf wird geprüft, ob diese vorhanden ist, die Neugenerierung nur ausgelöst, wenn keine Datei existiert.

Die Default Speicherorte der Datei „eRunbook.uuid";

- Unix/Linux: /var/eRunbook/
- Windows: %AppData%\eRunbook\
  entspricht %SystemDrive%\ProgramData\eRunbook\" (Windows® 7 und neuer) bzw. "%SystemDrive%\Documents and Settings\All Users\Application Data\eRunbook\" Windows® XP/Server2003).

Mögliche Konfigurationseinstellungen in der spezifischen eRunbook.conf (<agent|scriptmodule>/etc/eRunbook.conf):

```
create_uuid_file=<yes|no> Erstellen der „eRunbook_uuid" Datei, default ist "yes"
dir_uuid_file_win=
dir_uuid_file_unix=
```

Das Programm wird in Binary Form für Linux (x86, x64), Windows (x86), Solaris (x86, Sparc), HPUX, AIX und MacOS geliefert.

6.2 Oracle Datenbanken

Die Agenten ermitteln den Lizenzstatus der Oracle Datenbanken auf den abgefragten Serversystemen. Hierzu müssen neben Systeminformationen auch Informationen aus den Oracle Datenbanken abgefragt werden.

Hinweise:	Die Agenten lesen keine applikationsspezifischen Daten aus. Die Agenten lesen keine Kundendaten aus.

Die Abfrage der Datenbanken ist out-of-the-Box so konfiguriert, dass kein besonderer Benutzer in den abzufragenden Datenbanken angelegt werden muss, um alle notwendigen Informationen zu erfassen.

Auf den Systemen werden alle laufenden Instanzen erkannt. Nach einem Wechsel in den Kontext des Prozesseigentümers erfolgt der lesende Zugriff auf die Datenbank mit dieser UID. Alternativ kann auch ein neuer Datenbankbenutzer angelegt oder ein vorhandener Benutzer verwendet werden. Diese Datenbankbenutzer benötigen ausschließlich lesenden Zugriff.

Zur Anlage und Rechtevergabe dieser Benutzer werden Grant-Skripte bereitgestellt.

Der Datenbankbenutzer kann auf allen Serversystemen für alle dort laufenden Oracle Datenbanken identisch sein.

6.2.1 Ausführung der Grant-Skripte

Der Aufruf des Grant-Skriptes **MUSS** als SYSDBA (oder vergleichbare Rolle) mit dem Recht zum Anlegen von Benutzern erfolgen! Die Grant-Skripte müssen einmalig für jede abzufragende Datenbankinstanz ausgeführt werden.

Der Aufruf des Grant-Skriptes lautet:

```
@novaratio_grantscript.sql <user> <password> <tablespace> <ORACLE_SID>
```

Die Parameter beim Aufruf des Grant-Skriptes bedeuten:

- <user> ist ein neuer Benutzername.
  Wenn er bereits existiert, wird eine entsprechende Meldung ausgegeben.
- <password> ist das Kennwort, mit dem der neue Benutzer angemeldet werden kann.
  Das Kennwort folgt den allgemeinen Regeln der vorgegebenen Datenbankrichtlinien.
- <tablespace> ist ein existierender Tabellenraum.
  Wenn dieser fehlt, wird eine Fehlermeldung ausgegeben und der Benutzer nicht angelegt.
- <SID> ist der ORACLE_SID der Datenbank, wo der Benutzer angelegt werden soll.
  Falls er nicht existiert, wird eine Fehlermeldung ausgegeben und der Benutzer nicht angelegt.

Besonderheit für Oracle 12c mit Pluggable Databases (PDB):

Das Grant-Skript **MUSS zwingend in der CDB$ROOT** ausgeführt werden. Es legt den übergebenen Benutzer sowohl automatisch in der CDB$ROOT als auch in allen zu der CDB generierten PDBs an.

6.2.2 Hinterlegung der Anmeldedaten

Wird die Datenabfrage der Serversysteme über einen bestimmten Datenbankbenutzer ausgeführt, so müssen Benutzername und Kennwort hinterlegt werden.

Diese Anmeldedaten müssen vom Administrator auf den jeweiligen Serversystemen in einer entsprechenden Anmeldedatendatei hinterlegt werden. Diese Datei kann mehrere Anmeldedaten, d.h. gültige Kombinationen von Benutzername und Kennwort, enthalten und kann identisch auf mehreren Serversystemen eingesetzt werden.

Die Einträge in der Anmeldedatendatei sind zeilenorientiert **mit einem Tabulator** als Trennzeichen zwischen Benutzername und Kennwort:

```
User1   password1
User2   password2
```

Kommentarzeilen sind nicht zulässig.

Unter Windows wird die Anmeldedatendatei in folgendem Pfad nach der Installation des Agenten erwartet:

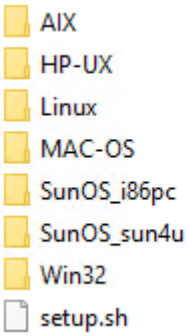
```
%Program Files (x86)%\eRunbook\product\agent\tools\login
```

Für Unix/Linux wird die Anmeldedatendatei nach der Agenteninstallation unter dem folgenden Pfad abgelegt:

```
/opt/eRunbook/product/agent/tools/login
```

6.3 Installation der Agenten

Die Installationsdateien für die jeweiligen Betriebssysteme liegen im Installationsverzeichnis (Abbildung - Installationspfad (siehe Seite 6)) im Unterverzeichnis „Inventory for Data Center“. Dort befinden sich folgende weitere Unterverzeichnisse:



6.3.1 Linux

Für die Installation unter Linux stehen sowohl RPM als auch DEB Pakete zur Verfügung.

RPM Pakete

Die Installation der signierten RPM Pakete erfordert root Rechte. Die Pakete lassen sich nur installieren, wenn die Signatur erkannt wurde. Um die Signatur zu erkennen, muss der Key vor der Installation importiert werden.

Hierzu z.B.:

```
rpm –import <PFAD>/signatur.key
```

ausführen. Danach kann das Paket installiert werden. Die Signatur lässt sich manuell mit

```
rpm --checksig <Paketname>
```

prüfen.

Um den Signaturcheck auszuschalten, kann die Option --nosignature des rpm verwendet werden:

```
rpm -Uv --nosignature <Paketname>
```

DEB Pakete

Die Installation der signierten DEB Pakete erfordert root Rechte.

```
dpkg -i <Paketname>
```

Hier kann die Signatur nur überprüft werden, wenn das Paket "debsigs" installiert wurde. Die Signatur kann dann wie folgt überprüft werden:

```
debsig-verify <Pfad/Paketname>
```

6.3.2 Solaris, HPUX, AIX

Die Installationsdateien müssen in einer bestimmten, mit ausgelieferten Struktur auf dem Zielsystem abgelegt werden. Die Installation des UNIX Agenten wird als root-Benutzer mit dem folgenden Befehl gestartet:

```
./setup.sh --agent
```

Der Erfolg der Installation kann mit dem folgenden Befehl überprüft werden:

```
#ps –aef | grep eRunbook_agent
```

6.3.3 Mac OS

Hinweis:	Für Mac OS kann alternativ zum Columbus Inventory Scanner, dieser Agent gewählt werden, der direkt an die Spider Data Center Appliance liefert.

Die Installationsdateien müssen in einer bestimmten, mit ausgelieferten Struktur auf dem Zielsystem abgelegt werden. Die Installation des Mac OS Agenten wird als root-Benutzer mit dem folgenden Befehl gestartet:

```
./setup.sh --agent
```

Der Erfolg der Installation kann mit dem folgenden Befehl überprüft werden:

```
#ps –aef | grep eRunbook_agent
```

Es muss darauf geachtet werden, dass das setup.sh Skript durch den root-Benutzer ausgeführt wird. Der admin-Benutzer hat nicht die nötigen Rechte.

6.3.4 Windows

Hinweis:	Sofern Oracle Datenbanken auf Windows Server genutzt werden, muss dieser Agent verwendet werden.

Die Installation des Agenten unter Windows erfolgt aus einer DOS-Shell mit Administratorrechten mit folgendem Befehl:

```
msiexec /i eRunbookAgent-<VERSION>.msi /qn
```

Der Agent ist nach der Installation automatisch als Dienst gestartet.

6.4 VMware vCenter

Zur Berechnung der Oracle Lizenznutzung im Zusammenhang mit VCenter werden u.a. zusätzlich Hardwareinformationen von den eingesetzten Hosts benötigt.

Die Hardwareinformationen der ESXi-Hosts werden über das verwaltende VCenter abgefragt. Hierzu muss auf den vCenter Servern das VMware vSphere PowerCLI eingerichtet sein. Zur Abfrage wird ein bereits vorhandener Nutzer mit Leserechten benötigt.

Die Abfrage wird durch den auf dem VCenter Server befindlichen Windows Agenten durchgeführt.

Hinterlegung Anmeldedaten

Die Anmeldedaten werden in der Datei vmlogin auf dem jeweiligen vCenterServer hinterlegt:

```
%Program Files (x86)%\eRunbook\product\agent\tools
```

Die Einträge in der Anmeldedatendatei sind zeilenorientiert mit einem Tabulator als Trennzeichen zwischen Benutzername und Kennwort:

```
User    password
```

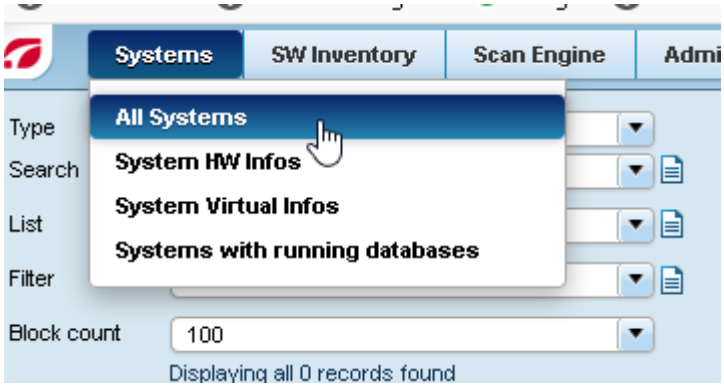
6.5 Agenten in der Spider Data Center Appliance einrichten

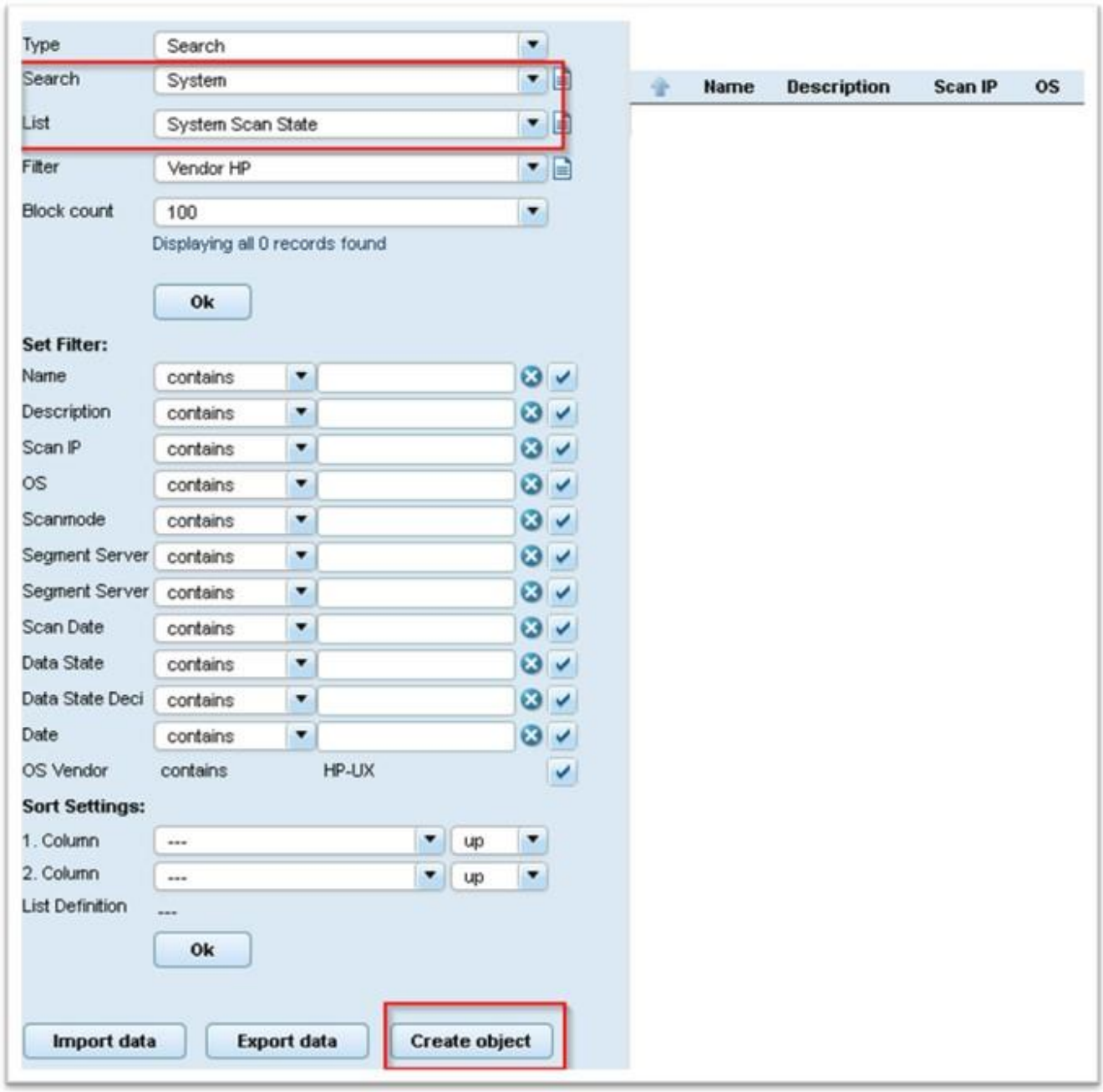
Es gibt zwei Möglichkeiten ein System in der Spider Data Center Appliance Instanz anzulegen:

1. Anlegen mit dem Editor
2. Anlegen durch Import

6.5.1 Anlegen mit dem Editor

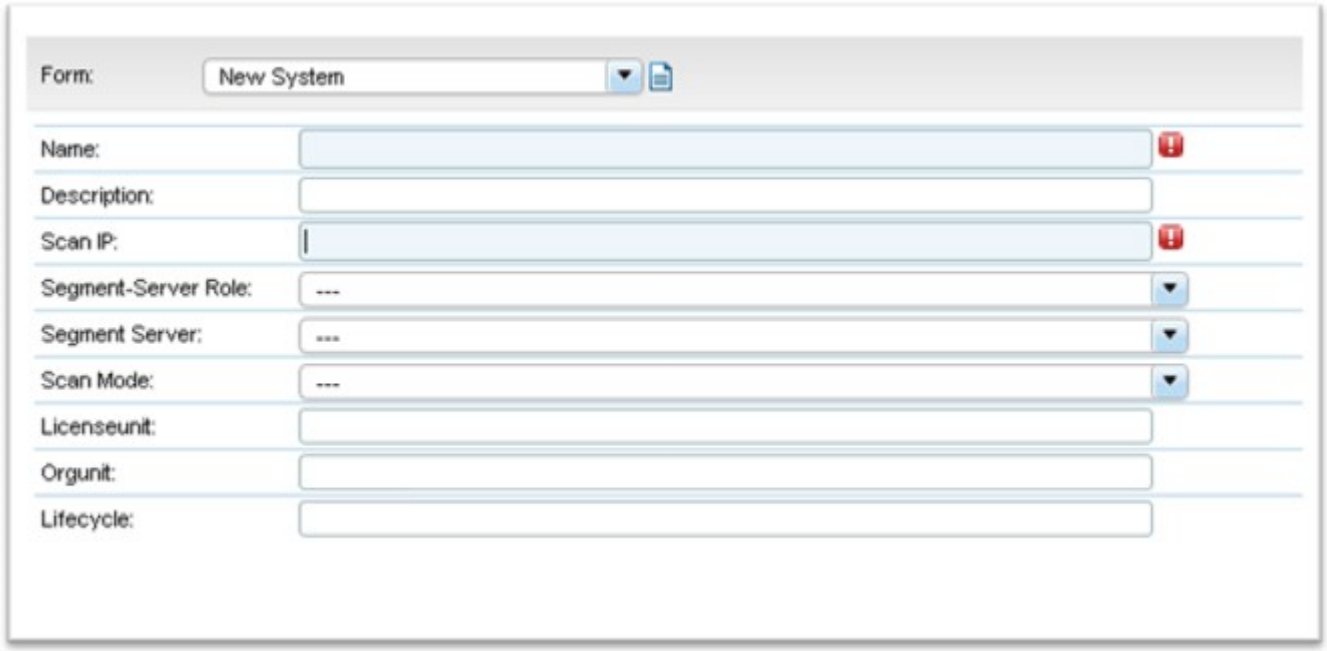
Über den Menüpunkt **Systems > All Systems** kann die Liste der eingerichteten Systeme angezeigt werden, welche zunächst leer ist:




Type
Search
Search
System
List
System Scan State
Filter
Vendor HP
Block count
100
Displaying all 0 records found
Ok
Set Filter:
Name
contains
Description
contains
Scan IP
contains
OS
contains
Scanmode
contains
Segment Server
contains
Segment Server
contains
Scan Date
contains
Data State
contains
Data State Deci
contains
Date
contains
OS Vendor
contains
HP-UX
Sort Settings:
1. Column
---
up
2. Column
---
up
List Definition

Ok
Import data
Export data
Create object
Name
Description
Scan IP
OS

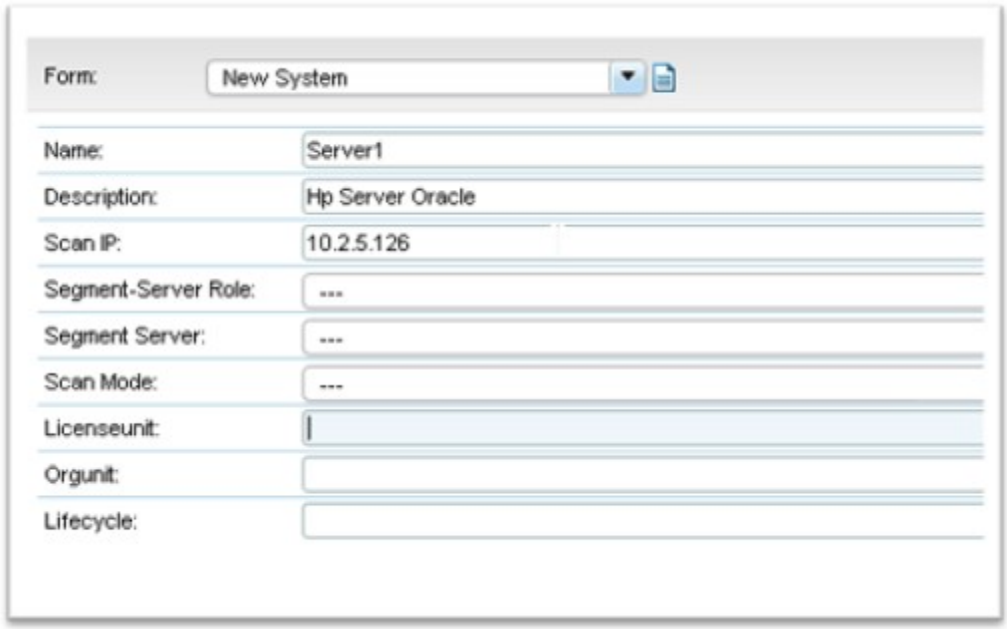
Mit der unten rechts befindlichen Schaltfläche „Create object“ kann der Editor zur Erfassung von neuen Systemen geöffnet werden.



Die Pflichtfelder "Name" und "IP" (IPv4) werden durch ein rotes Symbol rechts neben dem Eingabefeld hervorgehoben. Weitere Felder müssen nicht befüllt werden, allerdings wird dringend empfohlen das "Description" Feld nicht leer zu lassen.

Hinweise:	Der Systemname (Name) soll der Hostname des Systems sein. Anstelle der IP kann auch ein * eingetragen werden, wenn die Systemnamen per DNS auf der Appliance aufgelöst werden können.

Beispiel:



6.5.2 Import von größeren Mengen von Systemen

Um mehrere Systeme gleichzeitig zu registrieren können Listen im Excel-Format importiert werden.

Hinweis:	Es werden die Formate XLS und CSV unterstütz, aber nicht XLSX!

Die erste Zeile der Excel-Tabelle **muss** die Spaltenüberschriften enthalten:

- Name
- Description
- Scan IP

Die zweite Zeile der Excel-Tabelle **muss** die folgenden Werte enthalten:

- $attrib-ute:system:class_system_field_name
- $attrib-ute:system:class_system_field_description
- $attrib-ute:system:class_system_field_scan_ip

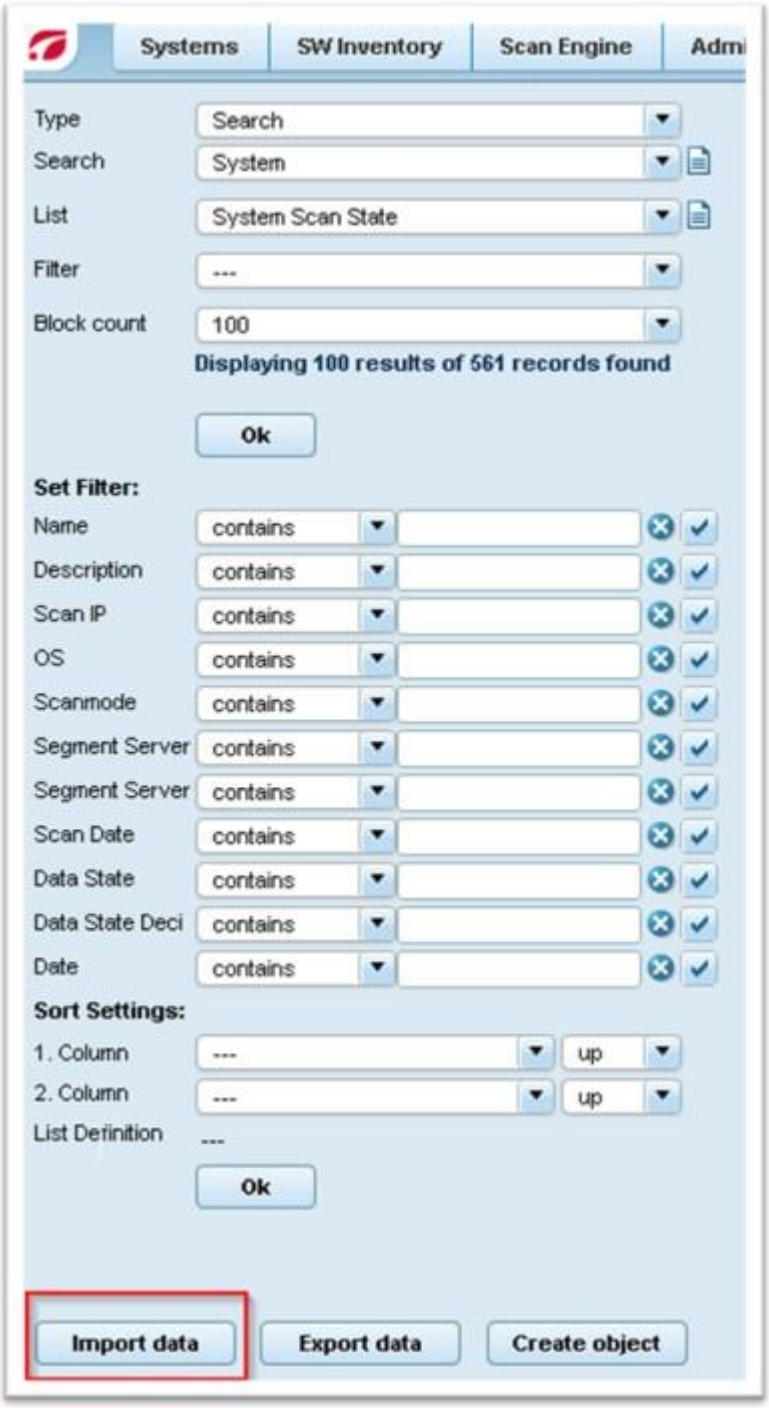
Alle weiteren Zeilen enthalten die Werte der zu importierenden Systeme.

Beispiel:

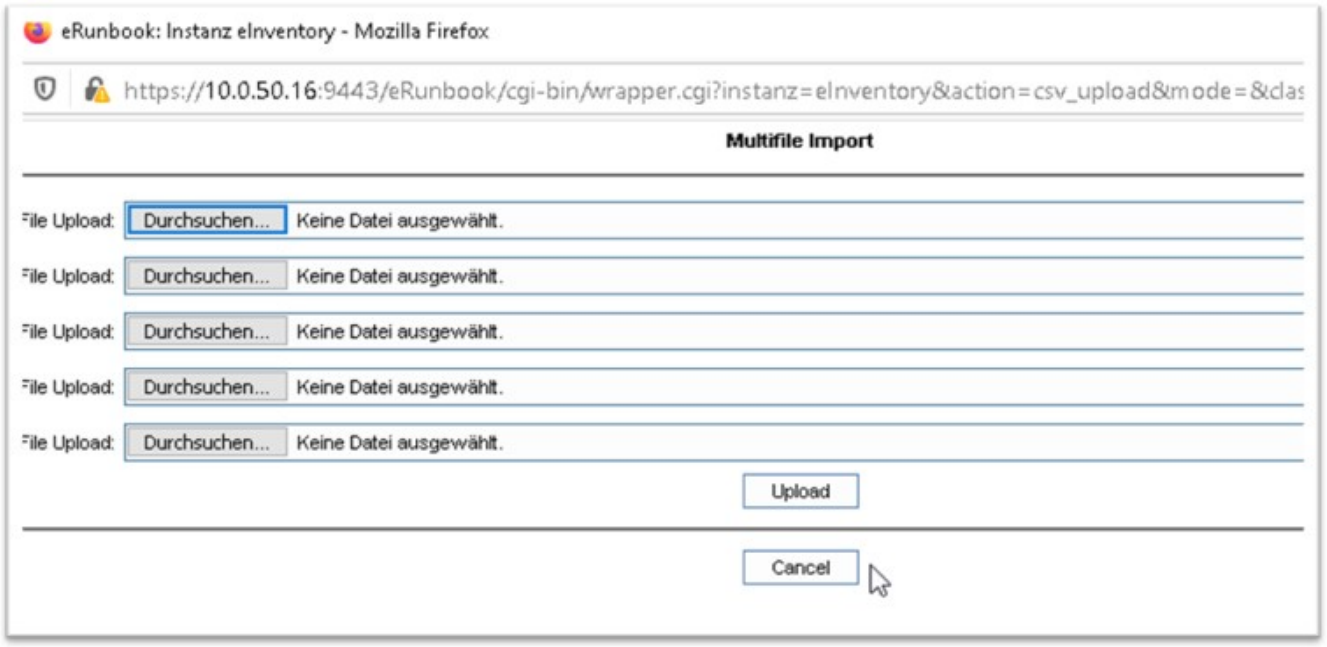
Name	**Description**	**Scan IP**
$at-trib-ute:system:class_system_field_name	$at-trib-ute:system:class_system_field_description	$at-trib-ute:system:class_system_field_scan_ip
jktest	ORA Test	10.0.100.92
jktest2	ORA Test 2	10.0.100.93

Hinweis:	Die beiden ersten Zeilen sind die Steuerzeilen für die korrekte Ablage der Daten in der Appliance. Sie dürfen NICHT verändert werden!

Der Datei-Upload und sowie der anschließende Import werden durch die Schaltfläche „Import data“ erreicht:

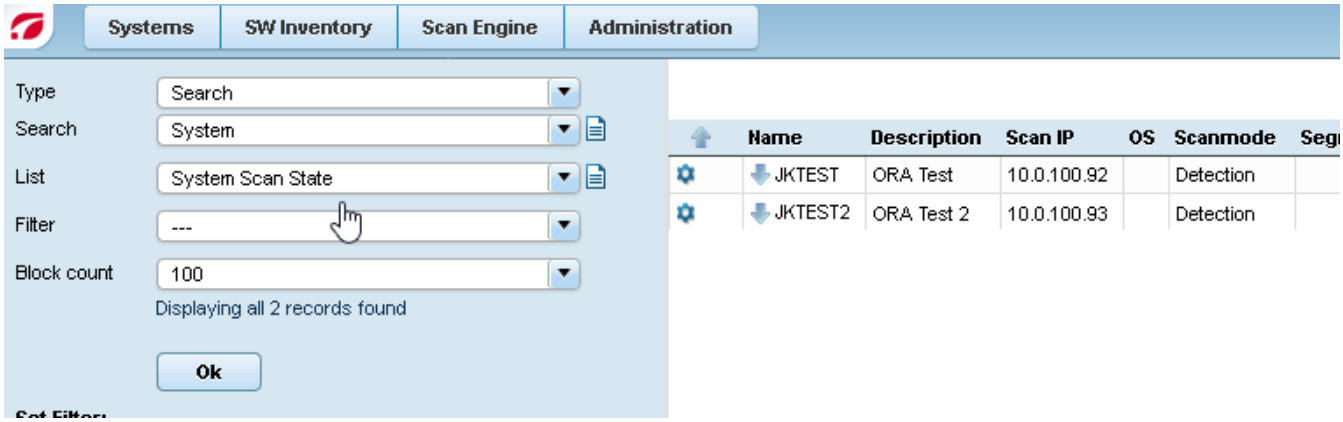


Zunächst muss der Ablageort der Importdatei im Dateisystem angegeben werden. Es können bis zu 5 Dateien gleichzeitig hochgeladen werden.



Durch Klick auf die Schaltfläche „Upload“ wird der Import gestartet. Der Dialog schließt sich nach dem Import automatisch, kann aber auch vorzeitig geschlossen werden.

Nach dem Import sind die Systeme in der Appliance eingerichtet:



6.6 Deinstallation der Agenten

Sollte ein Agent wieder deinstalliert werden müssen, sind hier je nach Betriebssystem unterschiedliche Vorgehensweisen beschrieben.

6.6.1 Deinstallation auf Linux, HPUX, AIX, MacOS

Die folgenden Befehle sind als root User auszuführen.

Vor der Deinstallation kann überprüft werden, ob der Agent überhaupt installiert ist und als Prozess läuft. Die Dateien des Agenten liegen standardmäßig unter "/opt/eRunbook".

Der Prozess lässt sich mit folgendem Kommando ermitteln:

```
ps -aef | grep eRunbook_agent
```

Bei der Deinstallation muss zunächst der Prozess mit folgendem Kommando gestoppt werden:

```
/etc/init.d/eRunbook stop
```

Der eRunbook Prozess sollte danach nicht mehr auftauchen. Zur Kontrolle kann folgender Befehl erneut ausgeführt werden:

```
ps -aef | grep eRunbook_agent
```

Anschließend kann die eRunbook Datei unter "/etc/init.d" gelöscht werden:

```
rm /etc/init.d/eRunbook
```

Ebenfalls kann nun der Agentenordner gelöscht werden, der standardmäßig unter "/opt/eRunbook" liegt:

```
rm –r /opt/eRunbook
```

Der Agent ist damit vollständig deinstalliert.

6.6.2 Deinstallation der RPM Pakete

Die folgenden Befehle sind als root-Benutzer auszuführen.

Zuerst sollte überprüft werden, ob der Agent als RPM Paket installiert wurde:

```
rpm –qa | grep eRunbook
```

Wird dieser hierbei aufgelistet, kann man das RPM Paket wie folgt deinstallieren:

```
rpm –e eRunbook-Agent
```

Der Prozess wird dabei automatisch beendet. Der eRunbook Ordner wird hierbei allerdings nur mit einem Zeitstempel verschoben. Dieser Ordner muss manuell entfernt werden.

```
rm –r /opt/eRunbook_<Zeitstempel>
```

Auch das eRunbook Startskript unter "/etc/init.d" muss im Anschluss von Hand gelöscht werden.

```
rm /etc/init.d/eRunbook
```

6.6.3 Deinstallation der DEB Pakete

Die folgenden Befehle sind als root-Benutzer auszuführen.

Sollte der Agent als .DEB Paket installiert worden sein, kann man sich den genauen Paketnamen mit folgendem Befehl anzeigen lassen:

```
dpkg –l | grep erunbook
```

Anschließend wird dieses Paket mit folgendem Befehl deinstalliert:

```
dpkg –r erunbook-agent
```

Der Prozess wird dabei automatisch beendet und der eRunbook Ordner gelöscht.

6.6.4 Deinstallation auf Windows

Der Agent lässt sich auf Windows über die Systemsteuerung entfernen. Über die Menüführung **Systemsteuerung > Programme > Programme und Features** erscheint die Liste der installierten Programme. Dort sucht man nach **eRunbookAgentStandard** und initiiert per Rechtsklick die Deinstallation.

Erfüllung der DSGVO/GDPR Anforderungen in Spider Produkten

Was ist DSGVO/GDPR und wer ist betroffen?

Ab dem 25. Mai 2018 gilt europaweit die neue Datenschutz-Grundverordnung (abgekürzt als GDPR oder DSGVO). Diese neue Verordnung kommt überall zur Anwendung, wo persönliche Daten von Menschen eingegeben, verarbeitet und gespeichert werden.

Die Verordnung muss bis zum 25. Mai 2018 umgesetzt werden. GDPR betrifft nicht nur Unternehmen innerhalb der EU, sondern auch Unternehmen, die Geschäftsbeziehungen zur EU unterhalten oder Daten von EU-Bürgern verarbeiten. Ebenso betroffen sind Softwareprodukte, die personenbezogene Daten verarbeiten.

Zur Sicherstellung der fristgerechten Umsetzung dieser umfangreichen Herausforderungen bereiten wir uns seit vielen Monaten intensiv auf die neue Datenschutz-Grundverordnung vor. Besondere Schwerpunkte liegen in der Sicherstellung der Betroffenenrechte, der Auftragsverarbeitung, den technischen und organisatorischen Maßnahmen, sowie der Fähigkeit, erforderliche Dokumentations-, Melde und Rechenschaftspflichtigen ordnungsgemäß zu erfüllen.

Bis einschließlich 24. Mai 2018 handeln und agieren wir nach dem noch geltenden Bundesdatenschutzgesetz. Auch die Spider Produkte müssen bis zu diesem Zeitpunkt die aktuellen gesetzlichen Vorgaben erfüllen. Mit Wirkung ab dem 25. Mai 2018 werden wir als Unternehmen, wie auch unsere Produkte, nach neuem Recht konform handeln.

7.1 Konnektoren mit personenbezogenen Daten

In den folgenden Kapiteln werden die Konnektoren aufgeführt, die personenbezogene Daten erheben.

Ab dem Release 1.1804 werden die Konnektoren nur noch mit einer DSGVO konformen Basiskonfiguration ausgeliefert, an manchen Stellen sind die erhobenen Daten für die Weiterverarbeitung zwingend notwendig, andere sind reine Komforteinstellungen und werden als solche gekennzeichnet aber im Standard nicht mehr aktiviert.

7.1.1 API basierende Konnektoren

VMware vCenter / ESX Server

Der Konnektor erhebt keinerlei personenbezogene Daten

Spider Data Center Inventory

Der Konnektor selbst erhebt keine personenbezogenen Daten, die Daten die aus dem Data Center Inventory übertragen werden, können aber sehr wohl solche Daten enthalten. Die Daten erhalten ggfs. IP Adressen, diese sind aber aufgrund der Einschränkung, dass das Data Center Inventory nur für Server verwendet wird, nicht einer einzelnen Person zuzuordnen.

Active Directory

Folgende personenbezogene Daten werden beim Export von Benutzerobjekten erhoben:

AD Attribut	Inhalt	Verwendung	Ausgabe im Standard aktiviert

AD Attribut	Inhalt	Verwendung	Ausgabe im Standard aktiviert
DistinguishedName	Eindeutiger RDN (relative distinguished name) der den Benutzer im Active Directory eindeutig identifiziert	Ermittlung eines Benutzers	Ja
UserPrincipalName	Kennung mit der im Kerberos-Authentifizierungs-System die Anmeldung des Benutzers realisiert werden kann.	Ermittlung eines Benutzers	Ja
EmailAddress	E-Mail Adresse des Benutzers	Die E-Mail Adresse wird z.B. zum Versenden von Benachrichtigungen verwendet.	Ja
GivenName	Vorname des Benutzers	Vorname zur Anzeige	Ja
Surname	Nachname des Benutzers	Nachname zur Anzeige	Ja
DisplayName	Anzeigename des Benutzers im Active Directory	Anzeigename zur Anzeige	Ja
ObjectGUID	Eindeutige ID die den Benutzer im Active Directory Identifiziert	Ermittlung eines Benutzers	Ja
ObjectSid	Eindeutige SID, die den Benutzer im Active Directory identifiziert.	Ermittlung eines Benutzers	Ja
UserAccountControl	Das Active Directory Attribut userAccountControl beinhaltet eine Reihe von Flags, die wichtige Grundeigenschaften eines Benutzerobjektes festlegen und mit denen der Status des betreffenden User Accounts abgefragt werden kann.	Wird verwendet um festzustellen ob ein Konto aktiv ist oder nicht.	Ja
SamAccountType	Dieses Attribut enthält Informationen zu jedem Kontotypobjekt	Wird verwendet um den Typen des Kontos, z.B. Benutzer, Gruppe, etc. festzustellen.	Ja
SamAccountName	Der Anmeldename wird verwendet, um Clients und Server zu unterstützen, die frühere Versionen des Betriebssystems ausführen		Nein
telephoneNumber	Das AD-Attribut Telefonnummer kann die primäre Telefonnummer enthalten, unter der der Benutzer bei der Arbeit verfügbar ist.		Nein
homePhone	Das Active Directory-Attribut homePhone kann die private Telefonnummer des Benutzers enthalten.		Nein
mobile	Das Active Directory-Attribut mobile kann die Mobiltelefonnummer des Benutzers enthalten.		Nein
Company	Firmenname des Benutzers		Nein
employeeID	Die ID eines Benutzers		Nein
Department	Im Active Directory Attribut Department kann die Bezeichnung der Abteilung oder eine Teams für den betreffenden Benutzers eingetragen werden.		Nein

AD Attribut	Inhalt	Verwendung	Ausgabe im Standard aktiviert
physicalDeliveryOfficeName	Enthält den Bürostandort am Geschäftssitz des Benutzers.		Nein
Title	Enthält den Job Titel des Benutzers. Diese Eigenschaft wird häufig verwendet, um den formellen Berufsbezeichner anzugeben.		Nein
co	Landesangabe für die Adresse des Benutzers		Nein
StreetAddress	Straßenangabe des Benutzers		Nein
l	Stadt der Benutzeradresse		Nein
st	Staat/Kanton/Bundesland		Nein
PostalCode	Postleitzahl		Nein
facsimileTelephoneNumber	Faxnummer		Nein
c	Land		Nein
cn	Der Name, der ein Objekt darstellt. Wird zum Ausführen von Suchen verwendet.		Nein

Microsoft Azure

Folgende personenbezogene Daten beim Export von Benutzerobjekten erhoben:

AD Attribut	Inhalt	Verwendung	Ausgabe im Standard aktiviert
OnPremisesSecurityIdentifier	SID die bei der Synchronisation mit dem lokalen Active Directory übertragen wird.	Ermittlung eines Benutzers	Ja
UserPrincipalName	Kennung mit der im Kerberos-Authentifizierungs-System die Anmeldung des Benutzers realisiert werden kann.	Ermittlung eines Benutzers	Ja
Mail	E-Mail Adresse des Benutzers	Ermittlung eines Benutzers	Ja
GivenName	Vorname des Benutzers	Vorname zur Anzeige	Ja
Surname	Nachname des Benutzers	Nachname zur Anzeige	Ja
DisplayName	Anzeigename des Benutzers im Active Directory	Anzeigename zur Anzeige	Ja
AccountEnabled	Enthält die Information ob das Konto aktiviert ist.		Ja
CompanyName	Firmenname des Benutzers		Nein
Country	Land des Benutzers, Teil der Adresse		Nein
CreationType	Gibt an, ob das Benutzerkonto ein lokales Konto für einen Azure Active Directory-B2C-Mandanten ist		Nein
DeletionTimestamp	Der Zeitpunkt, zu dem das Verzeichnisobjekt gelöscht wurde.		Nein

AD Attribut	Inhalt	Verwendung	Ausgabe im Standard aktiviert
DirSyncEnabled	Angabe ob der Benutzer aus dem lokalen AD synchronisiert wird.		Nein
TelephoneNumber	Telefonnummer		Nein
ImmutableId	ID des Benutzers		Nein
IsCompromised	Flag ob ein Konto als Sicherheitsrisiko gesehen wird.		Nein
LastDirSyncTime	Letzter Sync mit dem AD		Nein
MailNickName	MailNickName des Benutzers		Nein
Mobile	Kann die Mobiltelefonnummer der Benutzers enthalten.		Nein
Department	Abteilung des Benutzers		Nein
ObjectType	Typ das AzureAD Objektes		Nein
PasswordPolicies	Password Policies die dem Benutzer zugewiesen sind.		Nein
PhysicalDeliveryOfficeName	Enthält den Bürostandort am Geschäftssitz des Benutzers.		Nein
JobTitle	Enthält den Job Titel des Benutzers. Diese Eigenschaft wird häufig verwendet, um den formellen Berufsbezeichner anzugeben.		Nein
PreferredLanguage	Bevorzugte Sprache des Benutzers		Nein
RefreshTokensValidFromDateTime	Gültigkeitszeitraum des Validierungstokens		Nein
ShowInAddressList	Flag ob der Benutzer in der Adressliste angezeigt wird		Nein
StreetAddress	Straßenangabe des Benutzers		Nein
City	Stadt		Nein
State	Statt/Kanton/Bundesland		Nein
PostalCode	Postleitzahl		Nein
FacsimileTelephoneNumber	Faxnummer		Nein
UsageLocation	UsageLocation des Benutzers		Nein
UserType	Benutzer Typ		Nein

Folgende personenbezogene Daten werden beim Export der Tenant Details erhoben:

Tenant Attribut	Inhalt	Verwendung	Ausgabe im Standard aktiviert
DisplayName	Anzeigename des Tenants	Zur Anzeige	Ja
ObjectId	Eindeutige Kennung des Tenants	Ermittlung eines Benutzers	Ja
DirSyncEnabled	Angabe ob die Daten aus einem Active Directory synchronisiert wurden.		Ja
PreferredLanguage	Standardsprache des Tenant		Ja
ObjectType	Objekttyp des Tenants.		Nein
PostalCode	Postleitzahl		Nein
CountryLetterCode	Ländercode		Nein
City	Stadt		Nein
State	Staat/Kanton/Bundesland		Nein
Country	Land		Nein
TelephoneNumber	Telefonnummer		Nein
Street	Adresse		Nein

Adobe Online

Folgende personenbezogene Daten werden beim Export von Benutzerobjekten erhoben:

Adobe Attribut	Inhalt	Verwendung	Ausgabe im Standard aktiviert
email	E-Mail Adresse des Benutzers	Ermittlung eines Benutzers	Ja
username	Benutzername	Ermittlung eines Benutzers	Ja
firstName	Vorname des Benutzers	Angabe zur ggfs. nötigen manuellen Zuweisung	Ja
lastName	Nachname des Benutzers	Angabe zur ggfs. nötigen manuellen Zuweisung	Ja
status	Status des Benutzers		Ja
domain	Domain des Benutzers		Ja
countryCode	Land des Benutzers		Ja
userType	Typ des Benutzers		Ja

*Der Adobe Konnektor bietet hier keine Konfigurationsmöglichkeit.

Microsoft Application Virtualization (App-V)

Folgende personenbezogene Daten werden beim Export der Application Usage erhoben:

App-V Attribut	Inhalt	Verwendung
UserName	Anmeldename des Benutzers der ein bestimmtes Paket ausgeführt hat.	Ermittlung eines Benutzers

7.1.2 Datenbankbasierende Konnektoren

Für die datenbankbasierenden Konnektoren wird in der Regel immer die gleiche Menge an Parametern ausgegeben, die, die personenbezogene Daten beinhalten, werden hier aufgeführt:

Feld	Inhalt	Verwendung
Hostname	Eindeutiger Name der Maschine, kann in Abhängigkeit zu den zugrundeliegenden Firmenrichtlinien zur Benennung von Computern den Namen des Anwenders enthalten.	Identifizierung des Computers
MAC1 ... MAC4	MAC Adressen der Maschine	Kein technischer Verwendungszweck, daher ist die Ausgabe seit Release 1.1805 standardmäßig abgeschaltet*
IPAddressV4	IP v4 Adresse der Maschine	Kein technischer Verwendungszweck, daher ist die Ausgabe seit Release 1.1805 standardmäßig abgeschaltet*
IPAddressV6	IP v6 Adresse der Maschine	Kein technischer Verwendungszweck, daher ist die Ausgabe seit Release 1.1805 standardmäßig abgeschaltet*
LastLoggedOnUser	Name des zuletzt an der Maschine angemeldeten Benutzers	Ermittlung eines Benutzers

* Details zum Ein-/Ausschalten der Ausgabe sind im Kapitel Ausgabe von MAC und IP Informationen unterdrücken (siehe Seite 38) zu finden.

7.2 Inventory Komponenten

7.2.1 Windows

Ab Version 7.5.5.17 werden Felder mit personenbezogenen Daten nicht mehr automatisch erhoben, davon sind die folgenden Felder betroffen.

Inventory	Felder

Inventory	Felder
HardwareScan.csv	LastLoggedOnUser LastLoggedOnSAMUser LastLoggedOnUserSID MAC1 MAC2 MAC3 MAC4 IPAddressV4 IPAddressV6
InventoryItems.csv	OS.System.RegisteredUser OS.System.Organization OS.System.ProductKey

Bei Bedarf kann die Erhebung der Daten wieder eingeschaltet werden, Details dazu sind in der Beschreibung der Konfigurationsdateien für den Inventory Agent (siehe Seite 95) und Inventory Scanner (siehe Seite 107) zu finden.

7.3 Ablageorte von personenbezogenen Daten

Die von den Konnektoren und Inventory Komponenten erstellten und verarbeiteten Dateien befinden sich in dem im Setup angegebenen Ordner.

Die genauen Ablageorte können aus der Konfigurationsdatei (siehe Seite 17) des Data Collectors (SpiderDataCollector.cfg) ermittelt werden.

Sektion	Variable	Verwendung
OTBServer	DataDirectory	Ablage der Dateien die durch die Inventory-Komponenten erzeugt werden.
General	DataDirectory	Bereitstellung der Dateien die zur Recognition übertragen werden.

7.4 Sicherer Datentransport

Die Übertragung der Daten kann mit einem SSL Zertifikat abgesichert werden.

Für die Bereitstellung ist das jeweilige übergeordnete System zuständig, d.h. für die Inventory Komponenten kann die Verschlüsselung über den Data Collector bereitgestellt werden.

SSL verschlüsselte Übertragung (siehe Seite 113)

Die Sicherheit der Übertragung vom Data Collector zur Recognition stellt die Recognition zur Verfügung, Details dazu sind in der Technischen Referenz beschrieben.

FAQ

8.1 TCP/IP Socket basierende Kommunikation (OTB)

Der Spider Data Collector verwendet einen proprietären, TCP/IP basierenden, Kommunikations-Backbone der Object Transfer Bus (OTB) genannt wird, für den größten Teil der Client/Server Kommunikation

Einige der wichtigsten Punkte in der Implementierung von OTB sind:

- Konfigurierbare, auf einen Port beschränkte Übertragung
- Limitierung von Bandweite und/oder Datenvolumen, sowohl auf Server als auch auf Client Seite
- Kundenspezifische Verschlüsselung
- Blockbasierte Komprimierung
- Blockbasiertes Streaming

Im Ergebnis ergibt das eine Port-basierte Kommunikationsfähigkeit über all unsere Softwareplattformen mit der Fähigkeit, einfach in einem Netzwerk verwaltet zu werden. OTB kann die absolute Bandbreite des Dienstes unabhängig von der Anzahl der Knoten, dem Umfang und der Art der Daten verwalten. OTB unterstützt auch softwaregesteuerte Checkpoint-Neustarts und Übertragungswiederholung, sowie die Anpassung an spezifische Verschlüsselungsanforderungen für Regierungs-, Militär- und Finanzinstitutionen.

8.2 Ablageorte der Logdateien

Bei der Inanspruchnahme des Supports von Seiten der Flexera, müssen die folgenden Logdateien zur Verfügung gestellt werden.

Programm	Programmdatei	Logdatei(en)	Ablageort(e)
Data Collector	SpiderDataCollector.exe	Brainware.log Brainware_0.log Brainware_1.log Brainware_2.log Brainware_3.log Brainware_4.log	%windir% %ProgramData%\Columbus
Inventory Scanner	ColumbusInventoryScanner.exe		
Inventory Agent	ColumbusInventoryAgent.exe		
DSDC	DSDC.exe	DSDC-<Timestamp>.sil	%ProgramData%\brainwaregroup\DSDC\Logs
PowerShell Scripts	<PowerShellScriptName>.ps1	<PowerShellScript-Name><Timestamp>.sil	%ProgramData%\brainwaregroup\<ConnectorName>\Logs

Notiz Die Datei brainware.log wird regelmäßig geleert, vorherige Versionen werden als Brainware_0.log, Brainware_1.log usw. abgelegt.

Notiz Die *.sil Logdateien haben kein für Menschen lesbares Format.

8.3 Datenfluss

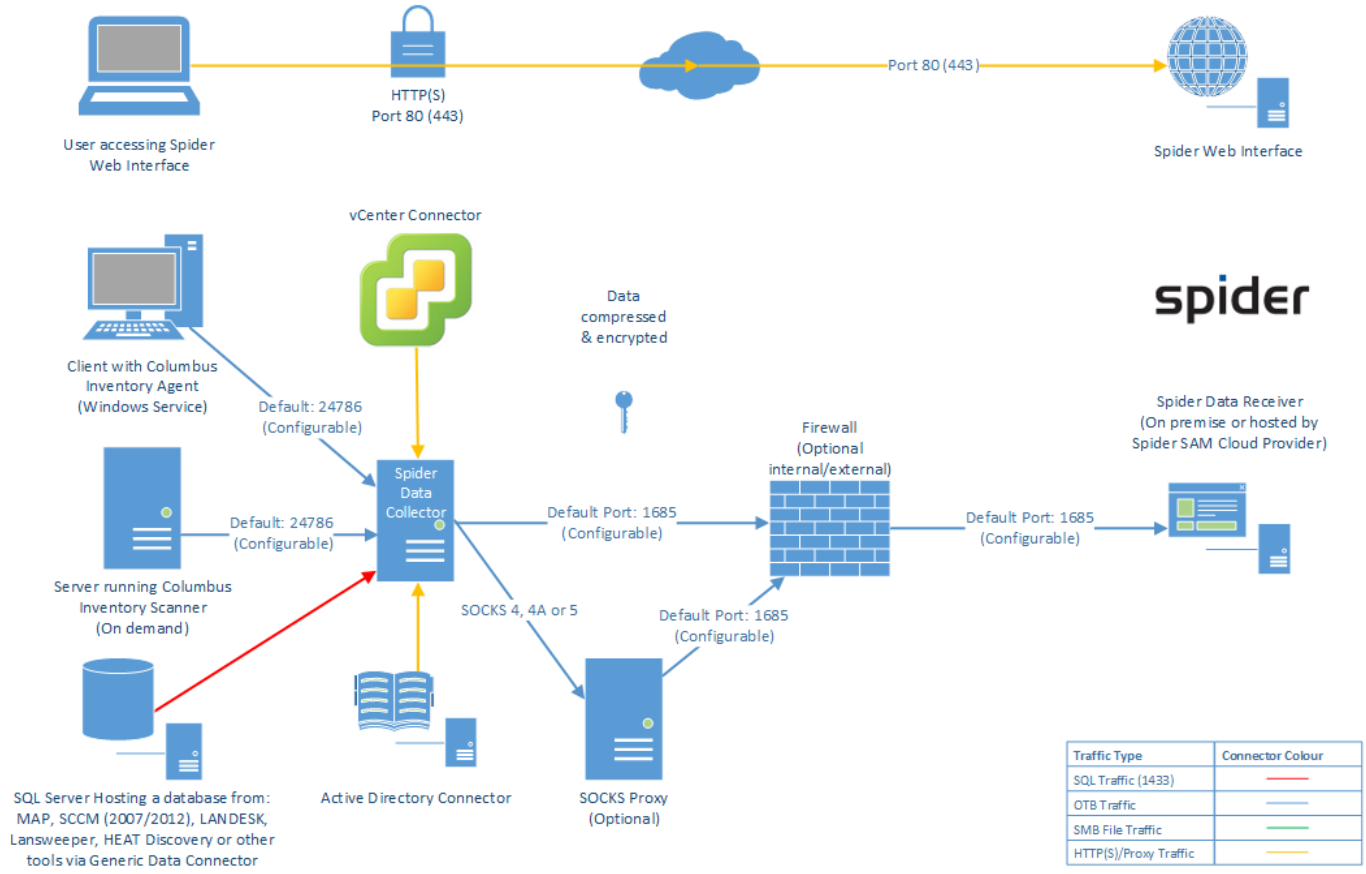


Abbildung - Datenfluss

Anhang

9.1 PowerShell Modul - bwgTools

Notiz:	Das bwgTools Modul wurde mit dem Release 1712.2, veröffentlicht, vorher hatte es den Namen bwgLogging

Das Logging und von den Konnektoren gemeinsam Benutzte Funktionen werden durch das Modul, das in den von Microsoft empfohlenen Pfaden hinterlegt wird, realisiert.

Dies sind die folgenden Pfade:

für x86: %ProgramFiles(x86)%\WindowsPowerShell\Modules

für x64: %ProgramFiles%\WindowsPowerShell\Modules

Da der Data Collector Dienst ein 32bit Dienst ist, wird er die x86 PowerShell Installation verwenden, wenn die Skripte von Hand ausgeführt werden (z.B. während Tests) wird das normalerweise im x64 Kontext einer Maschine ausgeführt. Aus diesem Grund wird das Modul in beiden oben genannten Pfaden abgelegt.

Obwohl diese Pfade von Microsoft vorgegeben sind, sind die Referenzen darauf erst ab PowerShell v4 automatisch verfügbar.

Im Falle einer älteren Version von PowerShell, müssen diese Pfade der (PowerShell) Umgebungsvariable "PSModulePath" hinzugefügt werden.

Der Inhalt der Variable kann wie folgt in einer PowerShell Konsole angezeigt werden:

```
$env:PSModulePath
```

Wenn die oben genannten Pfade nicht in der Variable enthalten sind, können sie mit dem folgenden Befehl ergänzt werden:

```
$p = [Environment]::GetEnvironmentVariable("PSModulePath", "Machine")
$newPSModulePath = Join-Path $env:ProgramFiles "WindowsPowerShell\Modules"
$p
$newPSModulePath

if($p -match  [regex]::Escape($newPSModulePath))
{
   Write-Host "Found" -ForegroundColor Green
}
else
{
   Write-Host "Not Found" -ForegroundColor Red
   $p += ";$($newPSModulePath)"
   $p
   #[Environment]::SetEnvironmentVariable("PSModulePath",$p, "Machine")
}
```

Achtung	Das o.g. Skript muss sowohl in der x86 als auch der x64 Windows PowerShell Konsole ausgeführt werden, ansonsten kann es sein, das eine der Architekturen das Modul nicht findet Die PowerShell-Konsolen sind unter: x86: %windir%\SysWOW64\WindowsPowerShell\v1.0\PowerShell.exe x64: %windir%\System32\WindowsPowerShell\v1.0\PowerShell.exe zu finden.

9.2 Stored Procedures des generischen Konnektors

Eine Zip Datei mit Vorlagen für die Stored Procedures kann hier bezogen werden:
https://docs.flexera.com/Spider64/GenericDataConnectorTemplates.zip

Achtung	Über SIDs Einige der folgenden Stored Procedures verwenden sogenannte SIDs (Security Identifiers) um die Identifizierung von Konten, Gruppen oder Gruppenmitgliedschaften zu realisieren. Damit alle Daten korrekt importiert werden können ist es notwendig das eine korrekt formatierte SID beim Export der Daten verwendet wird. Eine SID sieht z.B. wie folgt aus: S-1-5-21-1004336348-1177238915-682003330-512, der blau markierte Teil ist der Domain Identifier, der grüne Teil der relative Identifier. Sollte es nötig sein, diesen Teil zu anonymisieren aber trotzdem alle Funktionen beizubehalten, darf nur der blaue Teil verändert werden. Dieser muss über alle SIDs hinweg gleich gehalten werden, der grüne Teil muss auch bei mehrfachen Exporten für jedes Objekt immer gleichbleiben. Es muss also sichergestellt sein, das Hans Mustermann mit der SID S-1-5-21-1111111111-2222222222-333333333-1000 bei jedem weiteren Export ebenfalls die genannte SID erhält. Gleiches gilt ebenfalls für alle anderen Objekte die eine SID verwenden. Details zu SIDs werden hier beschrieben: https://technet.microsoft.com/de-de/library/cc778824(v=ws.10).aspx

9.2.1 dbo.swrGetWorkList

Diese Stored Procedure ermittelt eine Liste von Maschinen, die dann von den nachfolgenden Stored Procedures verarbeitet werden.

Spalte	Datentyp	Beschreibung	Pflichtangabe
Identifier	Variabel	Abhängig von den ermittelten Daten, kann das Format dieser Spalte variieren. Der Wert der in dieser Spalte ausgegeben wird, wird der Variable @identifier an die nachfolgenden Stored Procedures übergeben.	Ja
UUID	uniqueidentifier / GUID	Eine GUID zur eindeutigen Erkennung der Maschine	Nein
Urn	nvarchar(100)	URN der Maschine, in der Regel NULL	Nein
Domain	nvarchar(100)	DNS Domänenname der Maschine	Nein
DomainNameNetBIOS	nvarchar(100)	NETBIOS Domänenname der Maschine	Nein
Hostname	nvarchar(100)	Hostname der Maschine	Ja

Es wird in der folgenden Reihenfolge beim Import geprüft ob ein Gerät bereits existiert.

1. UUID
2. URN
3. DomainName + Hostname
4. DomainNameNetBIOS + Hostname

9.2.2 dbo.swrGetHardwareScan

Diese Prozedur ermittelt die Hardwaredaten einer Maschine

Parameter: @identifier (Format wie durch dbo.swrGetWorkList definiert)

Spalte	Datentyp	Beschreibung	Pflichtangabe
ScanDate	datetime	Datum und Uhrzeit des Scans	Ja
Manufacturer	nvarchar(256)	Hersteller der Maschine	Ja
Model	nvarchar(100)	Modell der Maschine	Ja
MAC1	nvarchar(100)	1. MAC Adresse	Nein
MAC2	nvarchar(100)	2. MAC Adresse	Nein
MAC3	nvarchar(100)	3. MAC Adresse	Nein
MAC4	nvarchar(100)	4. MAC Adresse	Nein
Serial	nvarchar(100)	Seriennummer	Nein
DeviceChassis	nvarchar(100)	Wird für die Erkennung des Gerätetypen verwendet. Es darf entweder DeviceChassis ODER ChassisType angegeben sein. Eine Liste mit gültigen Werten ist in der unteren Tabelle zu finden.	Nein
ChassisType	int	Wird für die Erkennung des Gerätetypen verwendet. Es darf entweder DeviceChassis ODER ChassisType angegeben sein. Eine Liste mit gültigen Werten ist in der unteren Tabelle zu finden.	Nein
ProcessorManufacturer	nvarchar(100)	Hersteller des Prozessors	Nein
ProcessorType	nvarchar(256)	Name des Prozessors	Nein
ProcessorSpeed	int	Geschwindigkeit des Prozessors	Nein
CPUCount	int	Anzahl der physikalischen Prozessoren	Nein
CPUCoreCount	int	Anzahl der Cores (Summe aller Cores, aller Prozessoren)	Nein
CorePerCPU	int	Anzahl der Cores eines einzelnen Prozessors	Nein
CPULogicalCount	int	Anzahl logischer CPUS (Summe aller Cores, aller Prozessoren)	Nein
DiskTotalMB	int	Summe des Speicherplatzes aller im System verbauten Festplatten	Nein
DiskFreeMB	int	Summe des freien Speicherplatzes aller im System verbauten Festplatten	Nein
GraphicAdapter	nvarchar(100)	Name der Grafikkarte	Nein
GraphicMemoryMB	int	Hauptspeicher der Grafikkarte in MB	Nein
MemoryMB	int	Hauptspeicher der Maschine in MB	Nein
IPAddressV4	nvarchar(15)	IPv4 Adresse der Maschine	Nein

Spalte	Datentyp	Beschreibung	Pflichtangabe
IPAddressV6	nvarchar(50)	IPv6 Adresse der Maschine	Nein
CPUArchitecture	nvarchar(100)	CPU Architektur, z.B. amd64, x86, Itanium	Nein
OSCaption	nvarchar(100)	Name des Betriebssystems	Nein
LastLoggedOnUser	nvarchar(100)	Benutzer der zuletzt an der Maschine angemeldet war im Format Domäne\Benutzername	Nein
BIOSVendor	nvarchar(100)	Hersteller des BIOS	Nein
BIOSVersion	nvarchar(100)	Version des BIOs	Nein
BIOSDate	datetime	Datum des BIOS	Nein
InventorySource	nvarchar(100)	Name und Version der Inventarquelle	Nein
Class	nvarchar(100)	Assettyp, diese Einstellung setzt die automatische Erkennung über das Device Chassis außer Kraft. Mögliche Werte: Cluster Desktop Mobile Device Laptop Unknown Printer Router Server Switch Tablet Thin Client Virtual Client Virtual Server Network Device	Nein
LegalEntity	nvarchar(500)	Pfad der Geschäftseinheit	Nein
Parameter01	nvarchar(100)	Zusätzlicher Parameter 1	Nein
Parameter02	nvarchar(100)	Zusätzlicher Parameter 2	Nein
Parameter03	nvarchar(100)	Zusätzlicher Parameter 3	Nein
Parameter04	nvarchar(100)	Zusätzlicher Parameter 4	Nein
Parameter05	nvarchar(100)	Zusätzlicher Parameter 5	Nein

Gültige Werte für DeviceChassis and ChassisType

ChassisType	DeviceChassis	ChassisType	DeviceChassis
1	Other	15	Space-Saving
2	Unknown	16	Lunch Box

ChassisType	DeviceChassis	ChassisType	DeviceChassis
3	Desktop	17	Main System Chassis
4	Low Profile Desktop	18	Expansion Chassis
5	Pizza Box	19	Sub Chassis
6	Mini Tower	20	Bus Expansion Chassis
7	Tower	21	Peripheral Chassis
8	Portable	22	Storage Chassis
9	Laptop	23	Rack Mount Chassis
10	Notebook	24	Sealed-Case PC
11	Hand Held	99	Virtual
12	Docking Station	100	Thin Client
13	All in One	105	Mobile Device
14	Sub Notebook	110	AzureVM

9.2.3 dbo.swrGetFileScan

Diese Prozedur liefert die auf einer Maschine erkannten Dateien.

Parameter: @identifier (Format wie durch dbo.swrGetWorkList vorgegeben)

Spalte	Datentyp	Beschreibung	Pflichtangabe
Manufacturer	nvarchar(256)	Dateihersteller	Ja
ProductName	nvarchar(256)	Produktname aus Datei	Ja
ProductVersion	nvarchar(256)	Produktversion aus Datei	Ja
FileName	nvarchar(256)	Name der Datei	Ja
FileDescription	nvarchar(256)	Beschreibung aus Datei	Ja
FileVersion	nvarchar(256)	Version der Datei	Ja
FileSize	bigint	Größe der Datei	Ja
FilePath	nvarchar(512)	Pfad der Datei	Ja

9.2.4 dbo.swrGetSoftwareScan

Diese Prozedur gibt die auf der Maschine erkannte Software aus.

Parameter: @identifier (Format wie durch dbo.swrGetWorkList vorgegeben)

Spalte	Datentyp	Beschreibung	Pflichtangabe
Manufacturer	nvarchar(256)	Hersteller des Programms	Ja
SoftwareName	nvarchar(256)	Name des installierten Programms	Ja
SoftwareVersion	nvarchar(256)	Version des installierten Programms	Ja
LicenceRequirement	decimal(18,4)	Benötigte Lizenzmenge	Nein
SerialNo	nvarchar(100)	Seriennummer	Nein
InstallDate	date	Installationsdatum	Nein

Betriebssysteme

Das Betriebssystem ist in den Programmen einer Maschine nicht aufgeführt, daher muss dies als zusätzliche Information ausgegeben werden. Eine Liste der Betriebssysteme ist in der folgenden Tabelle ersichtlich:

SoftwareName	SoftwareVersion	Manufacturer
Microsoft Windows 2000 Advanced Server	5.0	Microsoft Corporation
Microsoft Windows 2000 Professional	5.0	Microsoft Corporation
Microsoft Windows 2000 Professionnel	5.0	Microsoft Corporation
Microsoft Windows 2000 Server	5.0	Microsoft Corporation
Microsoft Windows XP Professional	5.1	Microsoft Corporation
Microsoft Windows XP Professionnel	5.1	Microsoft Corporation
Microsoft Windows Server 2003	5.2	Microsoft Corporation
Microsoft Windows 7 Enterprise	6.1	Microsoft Corporation
Microsoft Windows 7 Enterprise K	6.1	Microsoft Corporation
Microsoft Windows 7 Enterprise N	6.1	Microsoft Corporation
Microsoft Windows 7 Professional	6.1	Microsoft Corporation
Microsoft Windows 7 Professional N	6.1	Microsoft Corporation
Microsoft Windows 7 Professionnel	6.1	Microsoft Corporation
Microsoft Windows 7 Ultimate	6.1	Microsoft Corporation
Microsoft Windows Server 2008 R2 Datacenter	6.1	Microsoft Corporation
Microsoft Windows Server 2008 R2 Enterprise	6.1	Microsoft Corporation
Microsoft Windows Server 2008 R2 Foundation	6.1	Microsoft Corporation
Microsoft Windows Server 2008 R2 Standard	6.1	Microsoft Corporation

SoftwareName	SoftwareVersion	Manufacturer
Microsoft Windows 8 Enterprise	6.2	Microsoft Corporation
Microsoft Windows 8 Pro	6.2	Microsoft Corporation
Microsoft Windows Server 2012 Datacenter	6.2	Microsoft Corporation
Microsoft Windows Server 2012 Standard	6.2	Microsoft Corporation
Microsoft Windows 8.1 Enterprise	6.3	Microsoft Corporation
Microsoft Windows 8.1 Pro	6.3	Microsoft Corporation
Microsoft Windows Server 2012 R2 Standard	6.3	Microsoft Corporation
Microsoft Windows 10 Enterprise	10.0	Microsoft Corporation
Microsoft Windows 10 Enterprise 2015 LTSB	10.0	Microsoft Corporation
Microsoft Windows 10 Enterprise Edition	10.0	Microsoft Corporation
Microsoft Windows 10 Home	10.0	Microsoft Corporation
Microsoft Windows 10 Home K	10.0	Microsoft Corporation
Microsoft Windows 10 Professional	10.0	Microsoft Corporation
Microsoft Windows 10 Professionnel	10.0	Microsoft Corporation
Microsoft Windows 10 Pro	10.0	Microsoft Corporation
Microsoft Windows 10 Pro N	10.0	Microsoft Corporation
Microsoft Windows Server 2016 Datacenter	10.0	Microsoft Corporation
Microsoft Windows Server 2016 Datacenter Edition	10.0	Microsoft Corporation
Microsoft Windows Server 2016 Standard	10.0	Microsoft Corporation
Microsoft Windows Server 2016 Standard Edition	10.0	Microsoft Corporation

SQL Server Editionserkennung

Zur besseren Erkennung der SQL Server Edition können dem Software Scan spezielle Einträge hinzugefügt werden. Diese müssen den nachstehenden Regeln entsprechen.

Spalte	Wert	Beschreibung
Manufacturer	Microsoft Corporation	Dieser Wert ist fix und darf nicht anders angegeben werden.
SoftwareName	*Microsoft SQL Server <XXXX> <YYYY>	Der SoftwareName muss(!) mit einem Sternchen beginnen, <XXXX> ist die 4-stellige Jahreszahl der SQL Server Version und <YYYY> ist die Edition

Spalte	Wert	Beschreibung
SoftwareVersion	<MajorVersion>.<MinorVersion>	Major und Minor Version des SQL Servers, durch einen Punkt "." getrennt und ohne führende Nullen

Beispiele

SoftwareName	Version
*Microsoft SQL Server 2005 Express Edition	9.3
*Microsoft SQL Server 2008 Developer Edition	10.3
*Microsoft SQL Server 2008 Developer Edition	10.4
*Microsoft SQL Server 2008 Express Edition	10.3
*Microsoft SQL Server 2008 R2 Developer Edition (64-bit)	10.51
*Microsoft SQL Server 2012 Developer Edition	11.1
*Microsoft SQL Server 2012 Developer Edition	11.3
*Microsoft SQL Server 2012 Developer Edition (64-bit)	11.1
*Microsoft SQL Server 2012 Enterprise Edition	11.1
*Microsoft SQL Server 2012 Enterprise Edition: Core-based Licensing	11.0
*Microsoft SQL Server 2012 Enterprise Edition: Core-based Licensing (64-bit)	11.0
*Microsoft SQL Server 2012 Express Edition	11.0
*Microsoft SQL Server 2012 Express Edition	11.3
*Microsoft SQL Server 2012 Express Edition (64-bit)	11.0
*Microsoft SQL Server 2014 Developer Edition	12.0
*Microsoft SQL Server 2014 Developer Edition	12.2
*Microsoft SQL Server 2014 Express Edition	12.0
*Microsoft SQL Server 2014 Express Edition	12.2
*Microsoft SQL Server 2014 Standard Edition	12.0
*Microsoft SQL Server 2014 Standard Edition	12.1
*Microsoft SQL Server 2014 Standard Edition	12.2
*Microsoft SQL Server 2016 Developer Edition	13.0
*Microsoft SQL Server 2016 Enterprise Edition: Core-based Licensing	13.0
*Microsoft SQL Server 2016 Express Edition	13.0
*Microsoft SQL Server 2016 Express Edition	13.1

9.2.5 dbo.swrGetDeviceRelationship

Diese Prozedur liefert Host-Gast Beziehungen zwischen Geräten zurück.

Parameter: keiner

Spalte	Datentyp	Beschreibung	Pflichtangabe
ChildDeviceUUID	uniqueidentifier	UUID des Gastes	Nein
ChildDeviceUrn	nvarchar(100)	ORN des Gastes	Nein
ChildDeviceDomainName	nvarchar(100)	Domänenname des Gastes	Nein
ChildDeviceHostName	nvarchar(100)	Hostname des Gastes	Nein
ChildDeviceDomainNetBIOS	nvarchar(100)	NetBIOS Domänenname des Gastes	Nein
ParentDeviceUUID	uniqueidentifier	UUID des Hosts	Nein
ParentDeviceUrn	nvarchar(100)	URN des Hosts	Nein
ParentDeviceDomainName	nvarchar(100)	Domänenname des Hosts	Nein
ParentDeviceHostName	nvarchar(100)	Hostname des Hosts	Nein
ParentDeviceDomainNetBIOS	nvarchar(100)	NetBIOS Domänenname des Hosts	Nein
DeviceRelationshipTypeID	int	Typ der Beziehung 1: Guest-Host 2: Host-Cluster	Nein
ScanDate	datetime	Scandatum	Ja

9.2.6 dbo.swrGetADUserObject

Diese Prozedur gibt Benutzerobjekte zurück.

Parameter: keiner

Spalte	Datentyp	Pflichtangabe
ObjectGUID	uniqueidentifier	Eine dieser Spalte wird zur Identifikation benötigt..
ObjectSid	nvarchar(184)	
DistinguishedName	nvarchar(255)	
UserPrincipalName	nvarchar(1024)	
EmailAddress	nvarchar(254)	Nein
SamAccountName	nvarchar(20)	Nein (aber empfohlen)
NetbiosDomainName	nvarchar(16)	nein (aber empfohlen)

Spalte	Datentyp	Pflichtangabe
Firstname	nvarchar(100)	Nein
Lastname	nvarchar(100)	Nein
DisplayName	nvarchar(256)	Nein
PhoneNo	nvarchar(100)	Nein
PrivatePhoneNo	nvarchar(100)	Nein
MobilePhoneNo	nvarchar(100)	Nein
Company	nvarchar(64)	Nein
StaffNo	nvarchar(100)	Nein
Department	nvarchar(100)	Nein
PhysicalDeliveryOfficeName	nvarchar(128)	Nein
JobTitle	nvarchar(100)	Nein
Country	nvarchar(100)	Nein
StreetAddress	nvarchar(1024)	Nein
Location	nvarchar(100)	Nein
State	nvarchar(128)	Nein
PostalCode	nvarchar(100)	Nein
FaxNo	nvarchar(100)	Nein
CountryCode	nvarchar(2)	Nein
Name	nvarchar(255)	Nein
UserAccountControl	int	Nein
SamAccountType	int	Nein

Achtung Damit die Erkennung und Zuweisung korrekt funktioniert ist es nötig das SAMAccountName und NetBiosDomainName angeben werden!

9.2.7 dbo.swrGetADGroupObject

Diese Prozedur gibt Gruppenobjekte zurück.

Spalte	Datentyp	Pflichtangabe
ObjectGUID	uniqueidentifier	Ja

Spalte	Datentyp	Pflichtangabe
ObjectSid	nvarchar(184)	Ja
DistinguishedName	nvarchar(256)	Ja
Name	nvarchar(256)	Ja
SamAccountName	nvarchar(256)	Ja

Achtung Diese Prozedur funktioniert nur, wenn gleichzeitig auch dbo.swrGetADGroupMember verwendet wird.

9.2.8 dbo.swrGetADGroupMember

Diese Prozedur gibt die Gruppenmitglieder zurück.

Spalte	Datentyp	Pflichtangabe
GroupObjectSID	nvarchar(184)	Ja
GroupObjectGUID	uniqueidentifier	Ja
GroupDistinguishedName	nvarchar(256)	Ja
MemberObjectGUID	uniqueidentifier	Ja
MemberObjectSID	nvarchar(184)	Ja
MemberName	nvarchar(256)	Ja
MemberDistinguishedName	nvarchar(256)	Ja
MemberSamAccountName	nvarchar(256)	Ja
MemberObjectClass	nvarchar(256)	Ja, entweder "User", "Group" oder "Computer"

Achtung Die Daten dieser Prozedur können nur dann korrekt verarbeitet werden, wenn dbo.swrGetADGroup auch verwendet wird.

Die verwendeten Benutzer- und Computerkonten müssen ebenfalls exportiert worden sein.

9.2.9 dbo.swrGetSwidScan

Diese Prozedur liefert die SWID Tags einer Maschine.

Parameter: @identifier (Format wie durch dbo.swrGetWorkList definiert)

Spalte	Datentyp	Pflichtangabe
SWCreatorName	nvarchar(256)	Ja
SWCreatorRegID	uniqueidentifier	Ja
Product_title	nvarchar(256)	Ja

Spalte	Datentyp	Pflichtangabe
Product_version	nvarchar(256)	Ja
SWLicensorName	nvarchar(256)	Ja
SWLicensorRegID	nvarchar(256)	Nein
SoftwareUnique	nvarchar(256)	Ja
SoftwareRegID	nvarchar(256)	Nein
TAGCreatorName	nvarchar(256)	Nein
TAGCreatorRegID	nvarchar(256)	Nein
LicenseActivation	nvarchar(256)	Nein
LicenseChannel	nvarchar(256)	Ja
LicenseCustomer	nvarchar(256)	Nein
SerialNumber	nvarchar(256)	Ja

.

9.3 Inventarisierung mittels MAP Toolkit

Das Microsoft Application and Planning Toolkit (MAP) ist ein Inventory Tool das ohne auf den zu inventarisierenden Maschinen zu installierende Software auskommt.

Die Informationen die mittels MAP erhoben werden, können auch durch den Data Collector verwendet werden.

Ressourcen:

Produktseite: http://technet.microsoft.com/en-us/solutionaccelerators/dd537566

Download: http://www.microsoft.com/en-us/download/details.aspx?&id=7826

Systemvoraussetzungen: https://www.microsoft.com/en-us/download/details.aspx?&id=7826

9.3.1 Datenbank

Im Standardfall wird das MAP Toolkit eine SQL Server 2012 Express LocalDB während des Setups installierten. Alternativ kann auch eine existierende Installation (auf der Maschine auf der MAP installiert wird) von SQL Server 2008, SQL Server 2008 R2, oder SQL Server 2012 verwendet werden Damit das MAP Setup den alternativen Datenbankserver akzeptiert, MUSS dort eine Instanz mit dem Namen "MAPS" existieren bevor das MAP Toolkit installiert wird.

Notiz	Das MAP Toolkit setzt voraus, das die Kollation der Datenbank Engine auf "SQL_Latin1_General_CP1_CI_AS" gesetzt ist.

Instance Name: MAPS

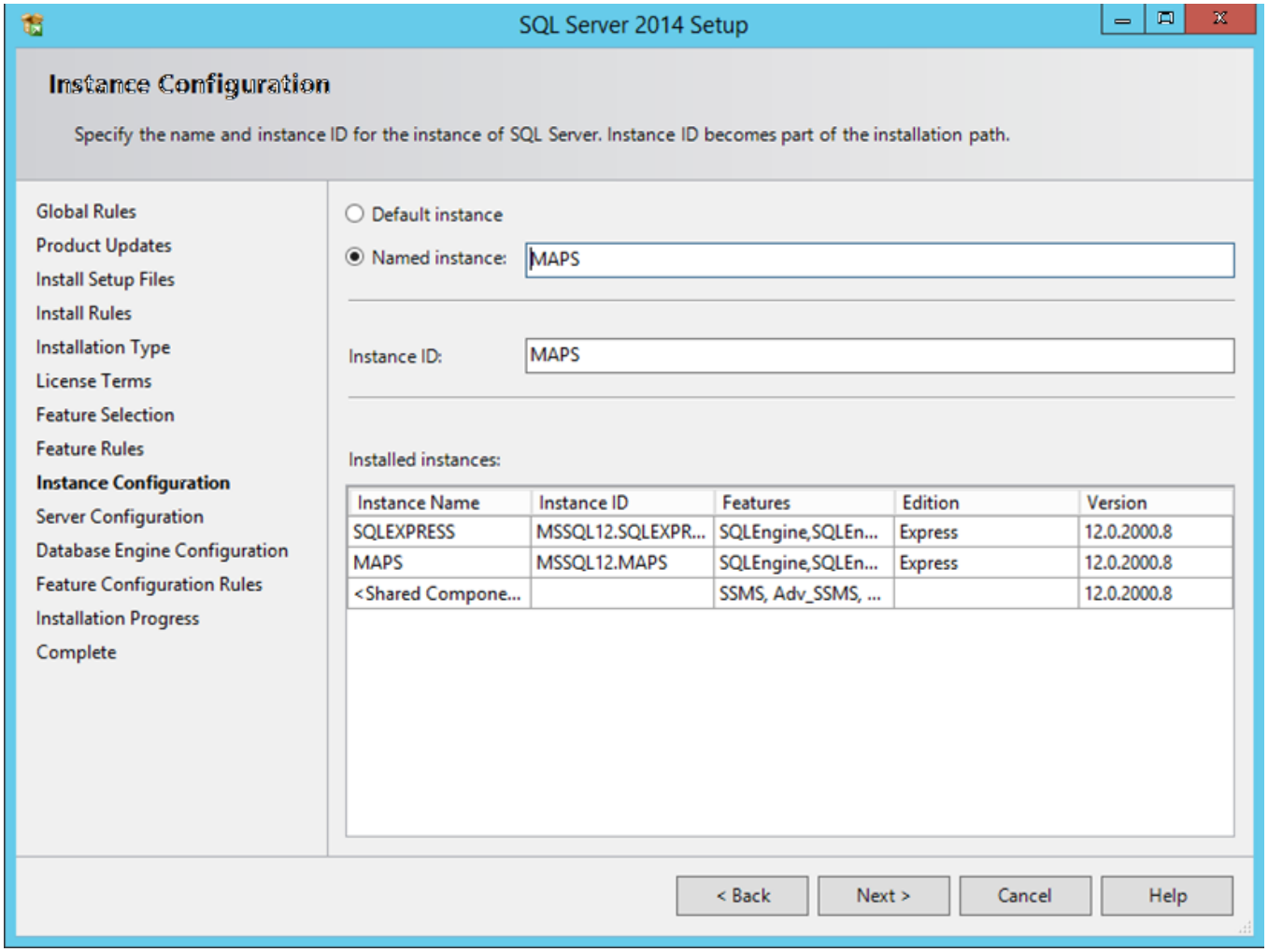


Abbildung - Instance Name

SQL Server Collation: SQL_Latin1_General_CP1_CI_AS

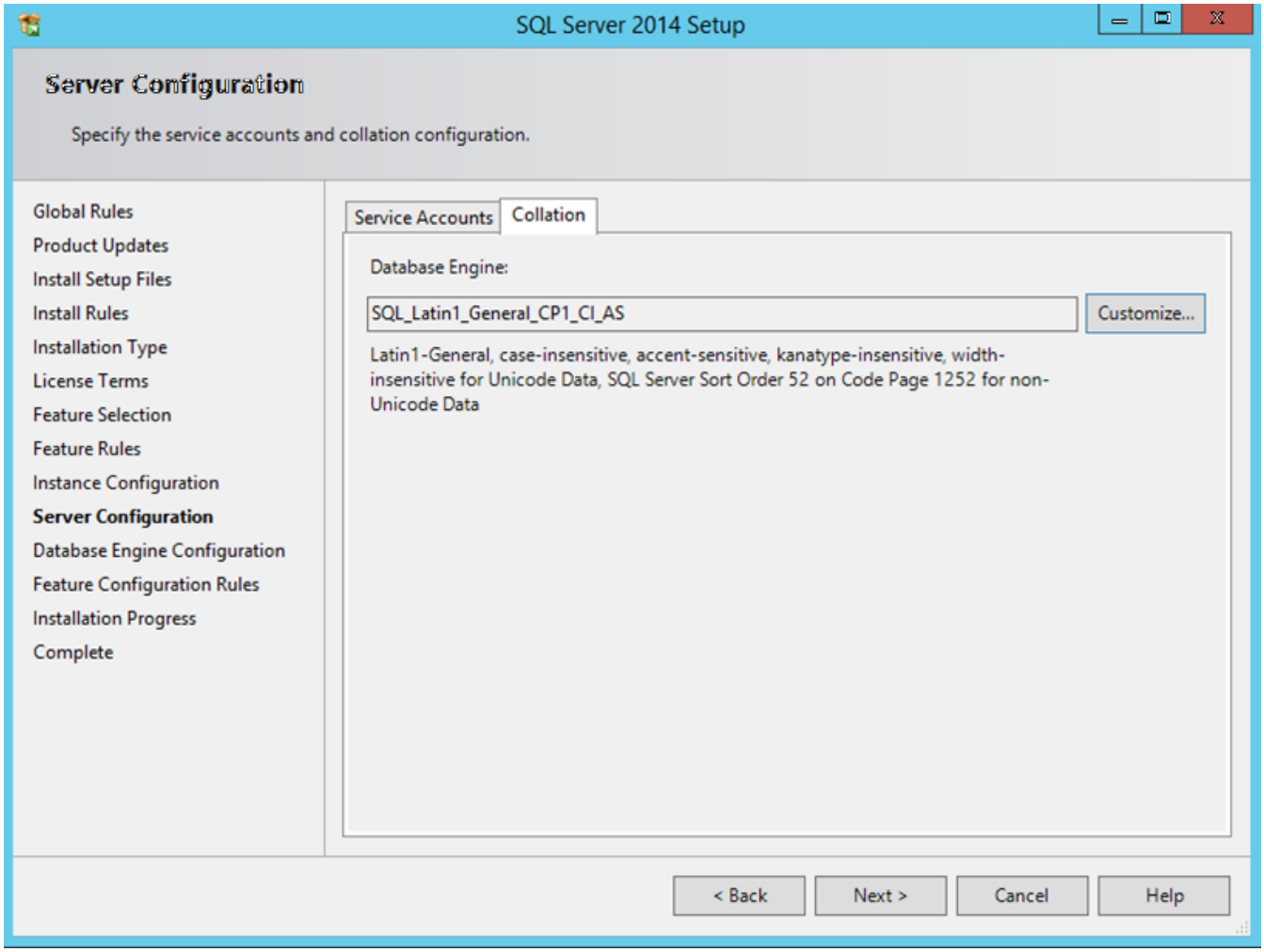
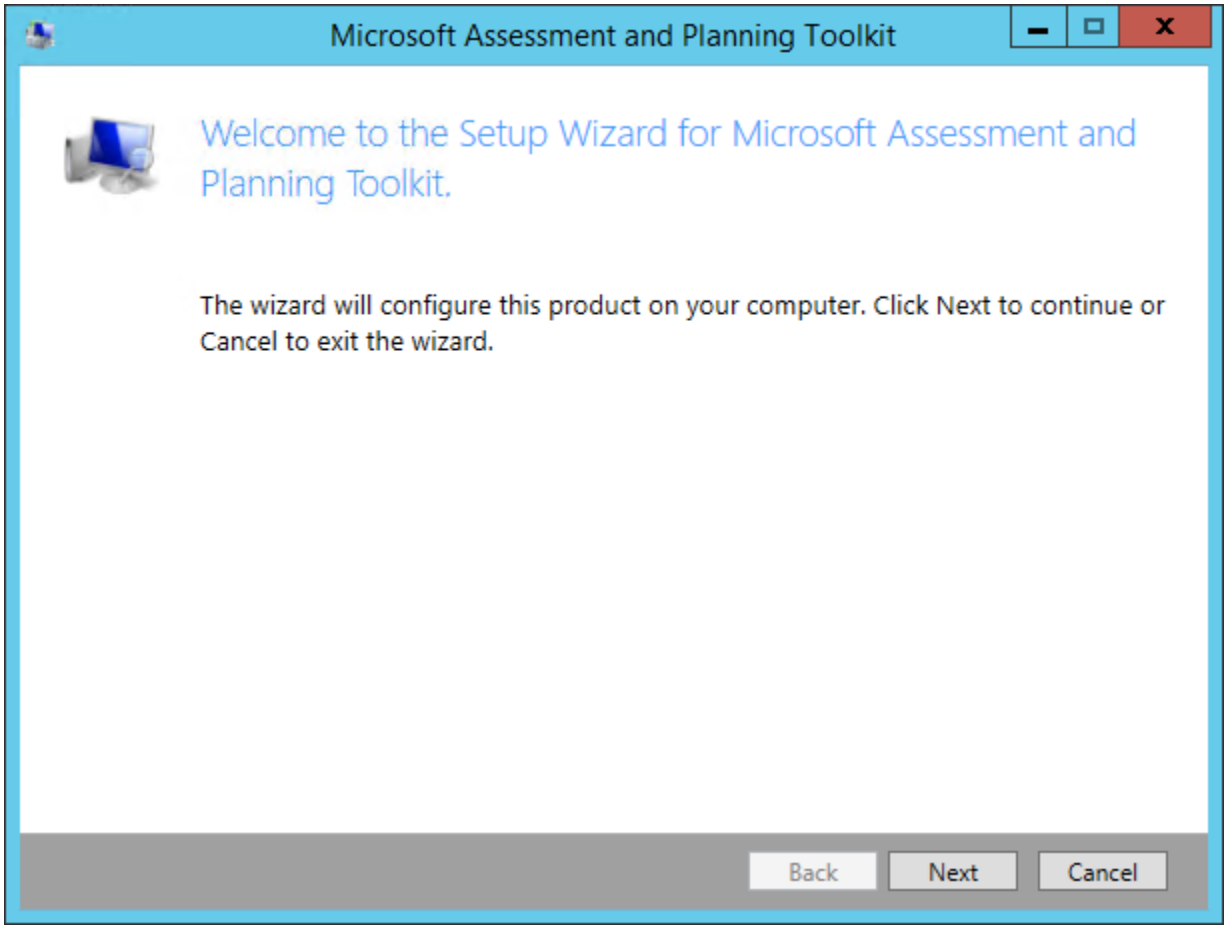


Abbildung - SQL Server Collation

Die Verwendung eines SQL Servers wie zuvor beschrieben bring einige Vorteile mit sich, der SQL Server läuft als Dienst und es können einfach zusätzliche Benutzer für den Zugriff eingerichtet werden. Auch muss dann der Data Collector beim Export nicht dafür sorgen das der entsprechende SQL Server (LocalDB) vor dem Export gestartet wird

9.3.2 Installation

Ausführen der "MapSetup.exe"



1. Next betätigen um zu beginnen.
2. Der erste Schritt prüft ob die Systemvoraussetzungen erfüllt sind, diese müssen ggfs. korrigiert werden bevor mit dem Setup fortgefahren werden kann.
3. Lizenzvereinbarung akzeptieren und auf Next klicken.
4. Ändern/Akzeptieren des Installationspfades, Next
5. Angaben zum Customer Experience Improvement Program machen, Next.
6. Install.

Wenn die Installation durchgelaufen ist, mit Finish das Setup verlassen.

9.3.3 Konfiguration

Wenn das MAP Toolkit zum ersten Mal ausgeführt wird, muss eine Datenbank für die Ablage der Daten erstellt werden. Nach der Vergabe eines Namens wie z.B. "MAPDB", OK klicken und die Datenbank erstellen.

Verwendung von LocalDB

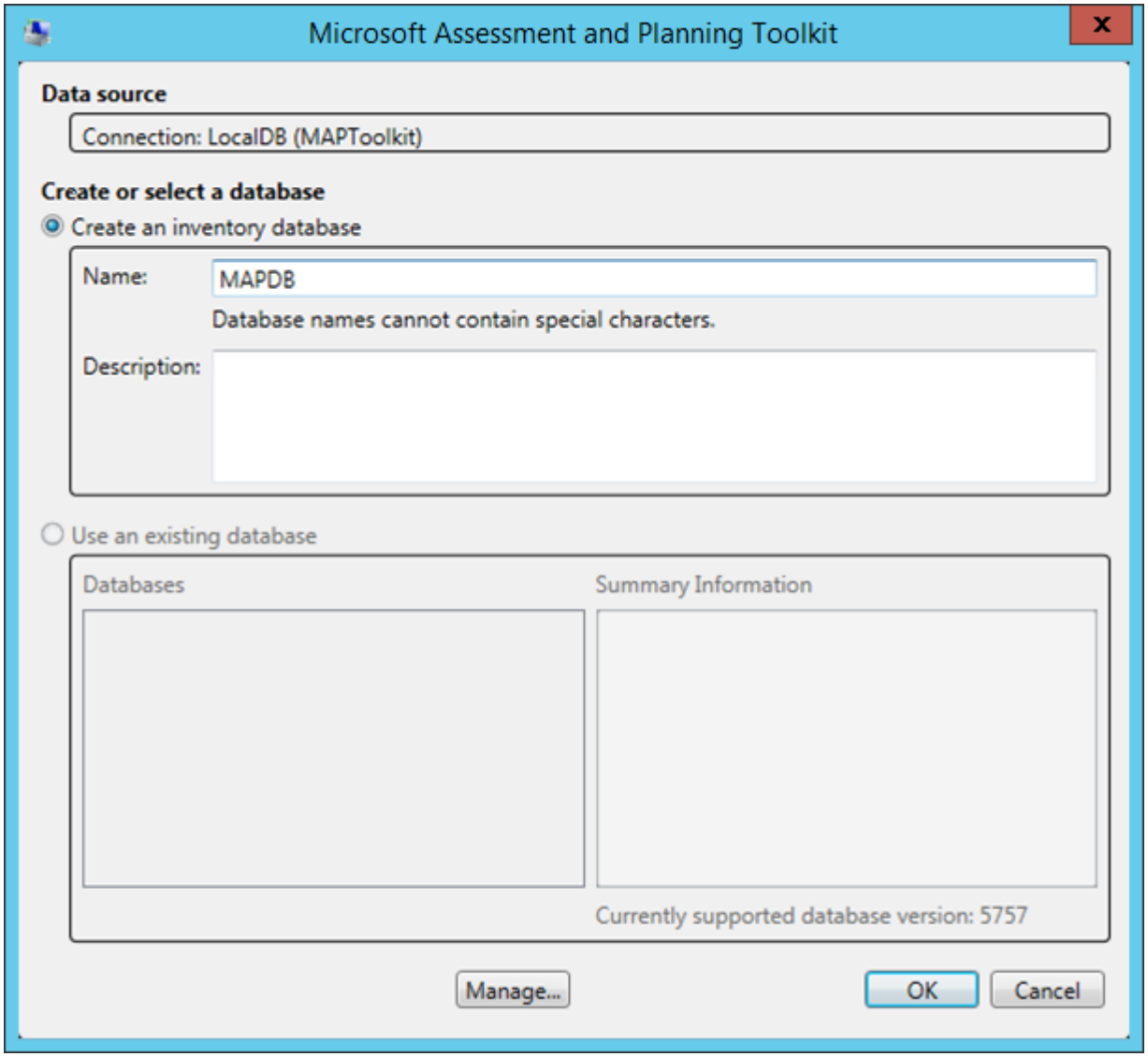


Abbildung - LocalDB

Alternative: Verwendung eines SQL Servers mit MAPS Instanz

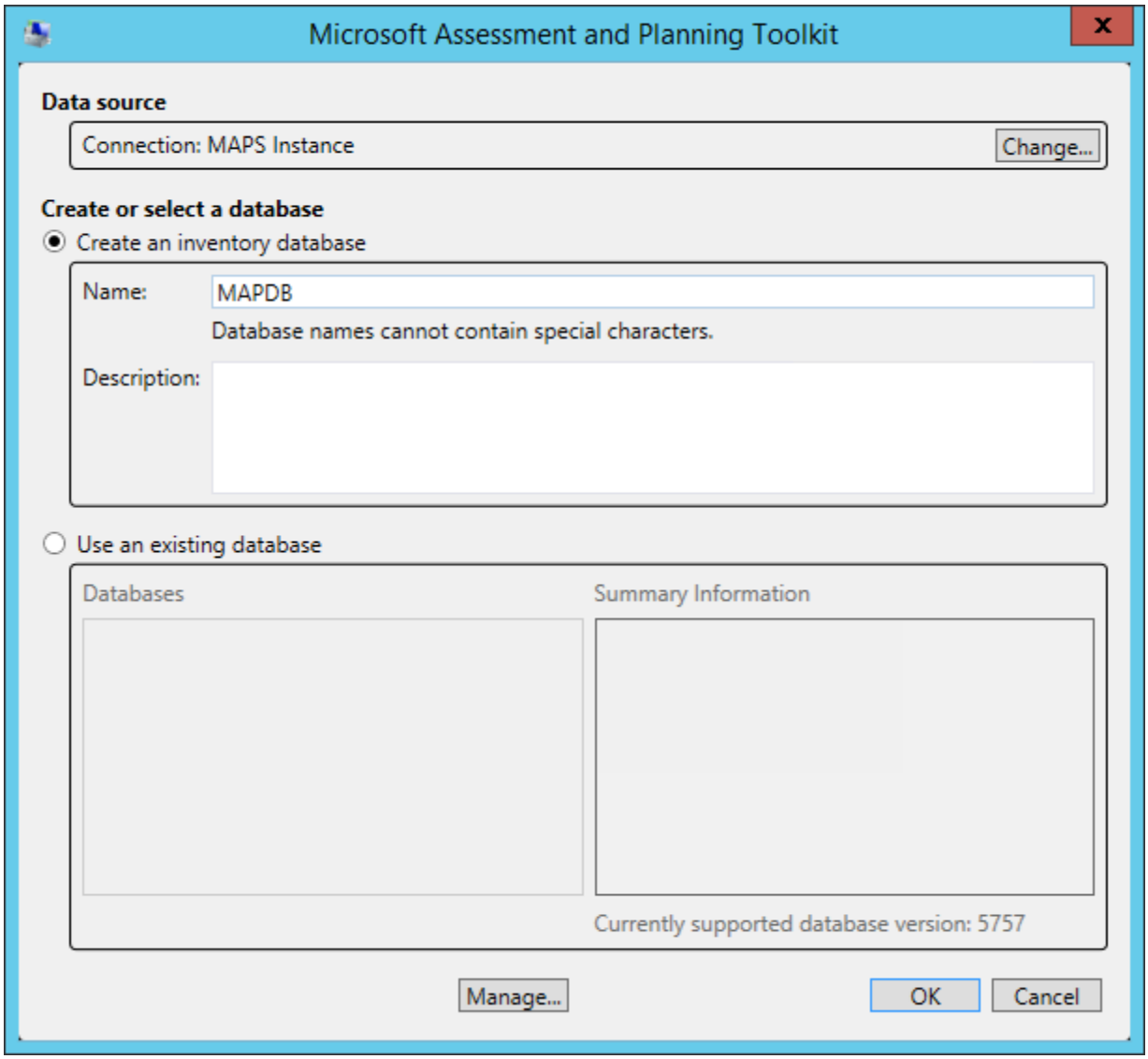


Abbildung - MAPS instance

9.3.4 Inventardaten erheben

Voraussetzungen um Daten zu erheben:

Das MAP Toolkit verwendet WMI um Inventardaten zu erheben. Es muss sichergestellt sein, das die Maschine auf der das MAP Toolkit installiert ist, Zugriff auf alle zu inventarisierenden Geräte hat. Hardware Firewalls bzw. die ggfs. auf den Maschinen installierte Windows Firewall müssen so konfiguriert sein, das MAP zugreifen kann.

Detaillierte Informationen sind hier zu finden:
http://social.technet.microsoft.com/wiki/contents/articles/8657.map-prepare-the-environment-wmi.aspx

In der Seitenleiste auf "Environment" klicken und dann "Collect Inventory Data" auswählen.

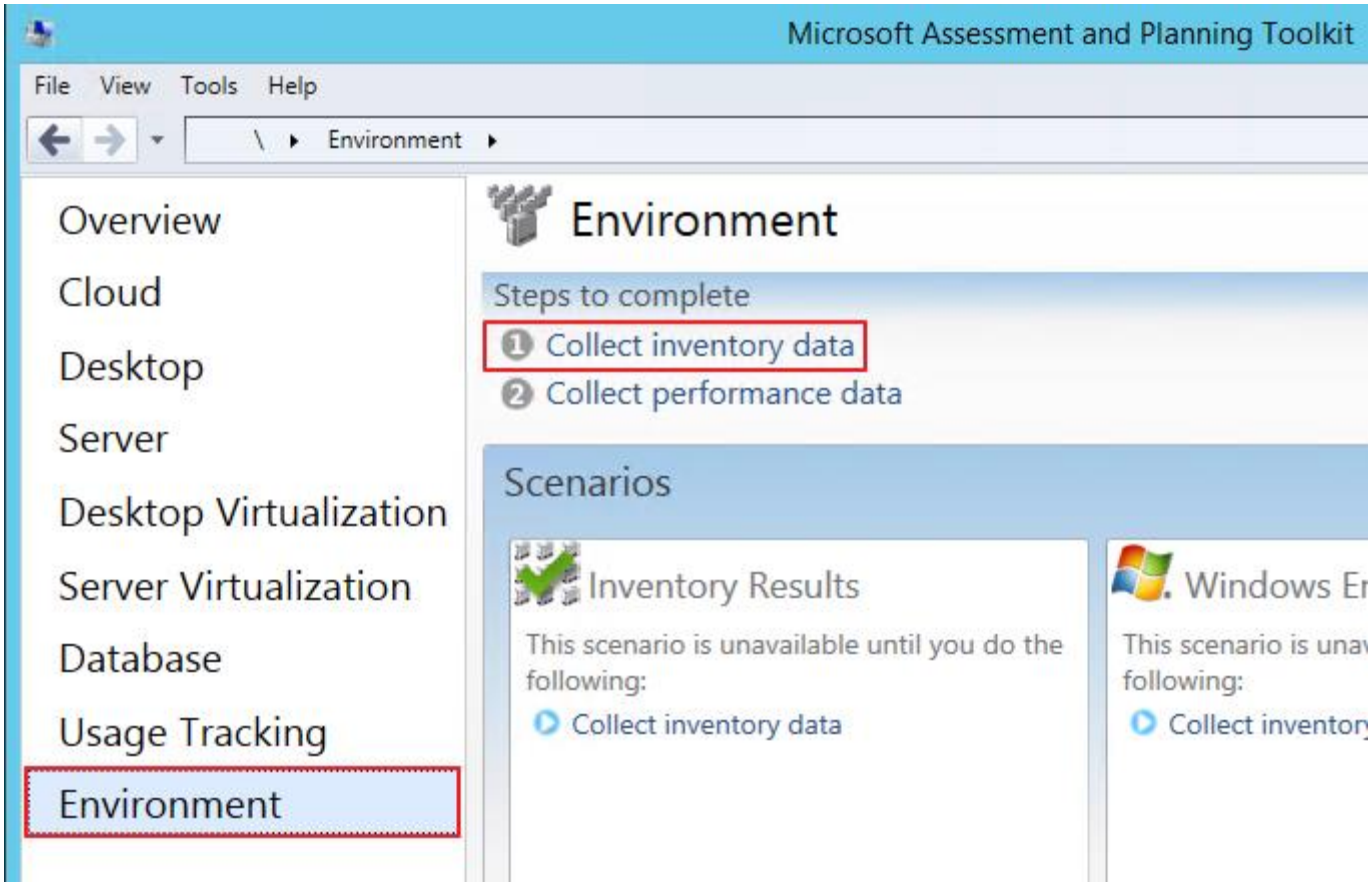


Abbildung - Collect inventory data

Die gewünschten Zielsysteme auswählen, Next.

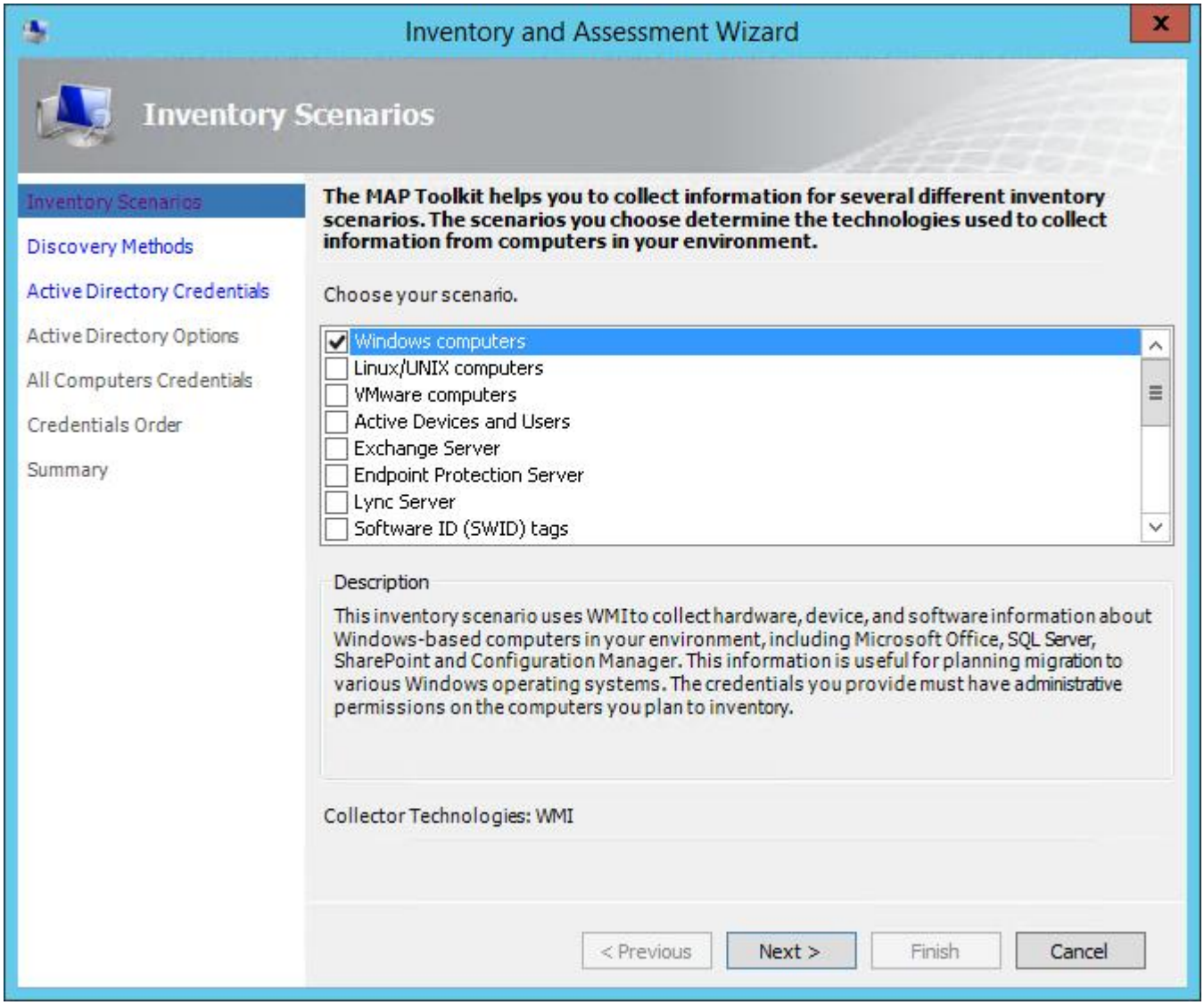


Abbildung - Choose scenario

Auf der Seite mit den Anmeldeinformationen sollte ein Benutzer angegeben werden der lokaler Administrator auf den Zielmaschinen ist, am besten eignet sich hierfür ein Domänenadministrator.

Nachdem alle Informationen eingegeben sind, beginnt automatisch der Scan, abhängig von der Anzahl der zu inventarisierenden Maschine und gewählten Scan Details kann der Vorgang einige Zeit in Anspruch nehmen.

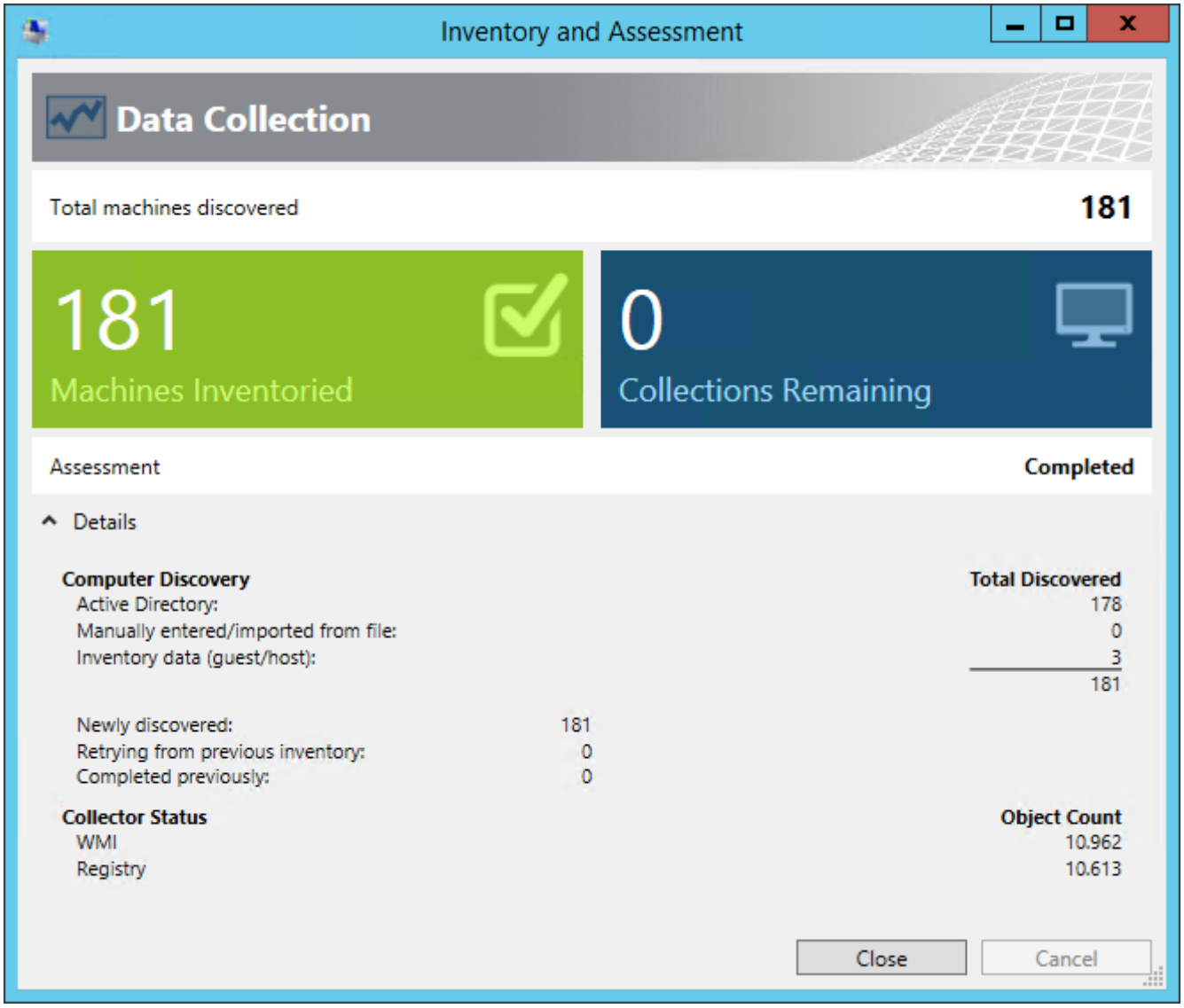


Abbildung - Data Collection

Nach Abschluss des Vorgangs wird der Erfolg des Scans angezeigt und wie viele Objekte erfasst wurden.

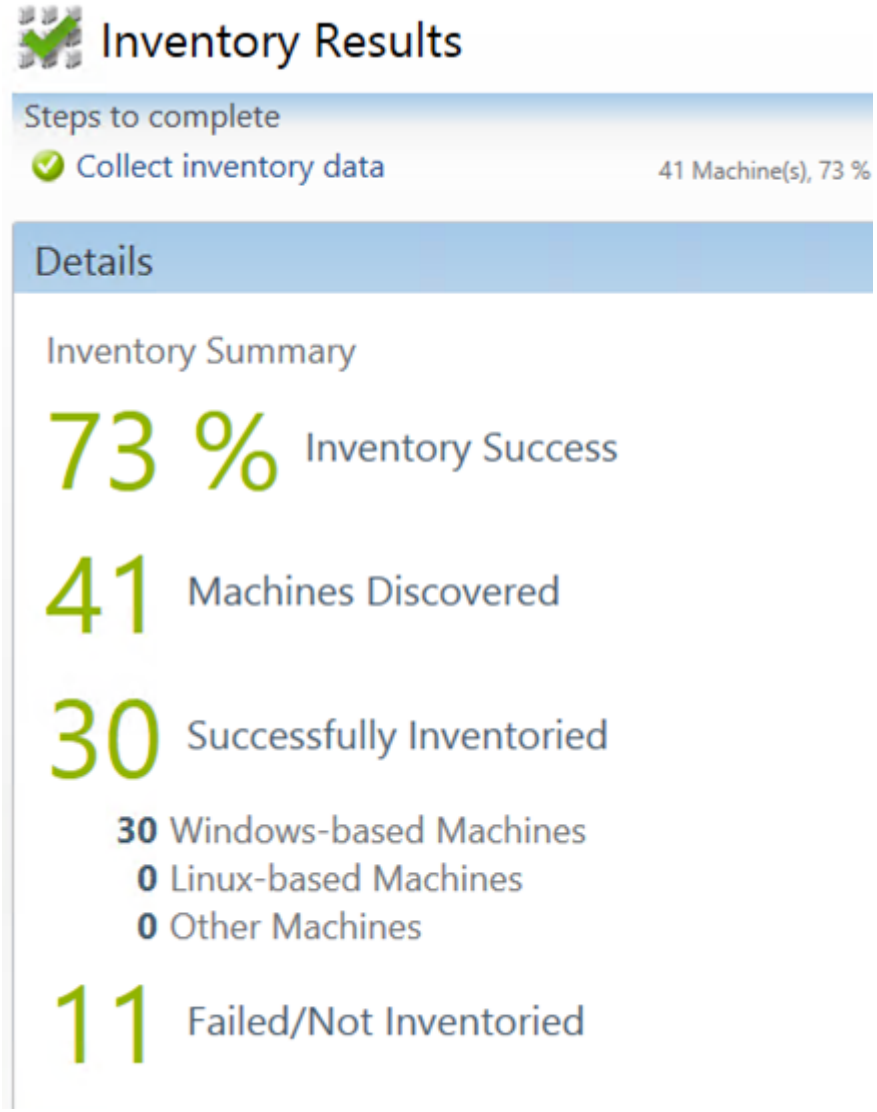


Abbildung - Inventory Results

VMware Daten

Die Voraussetzungen werden hier genannt:
http://social.technet.microsoft.com/wiki/contents/articles/12160.map-prepare-the-environment-vmware.aspx

VMware auswählen

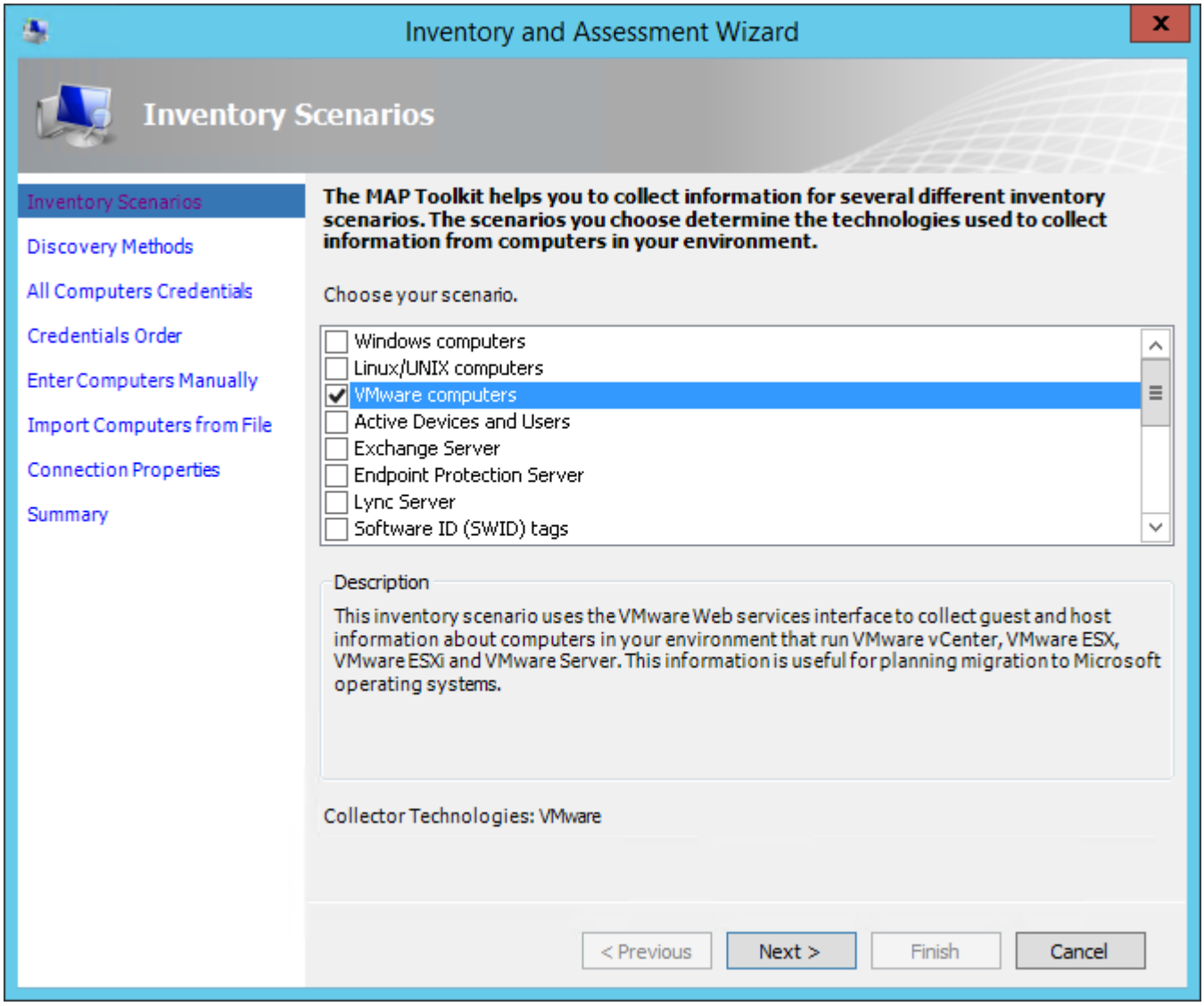


Abbildung - VMware Computers

Auswahl die vCenter Server manuell anzugeben

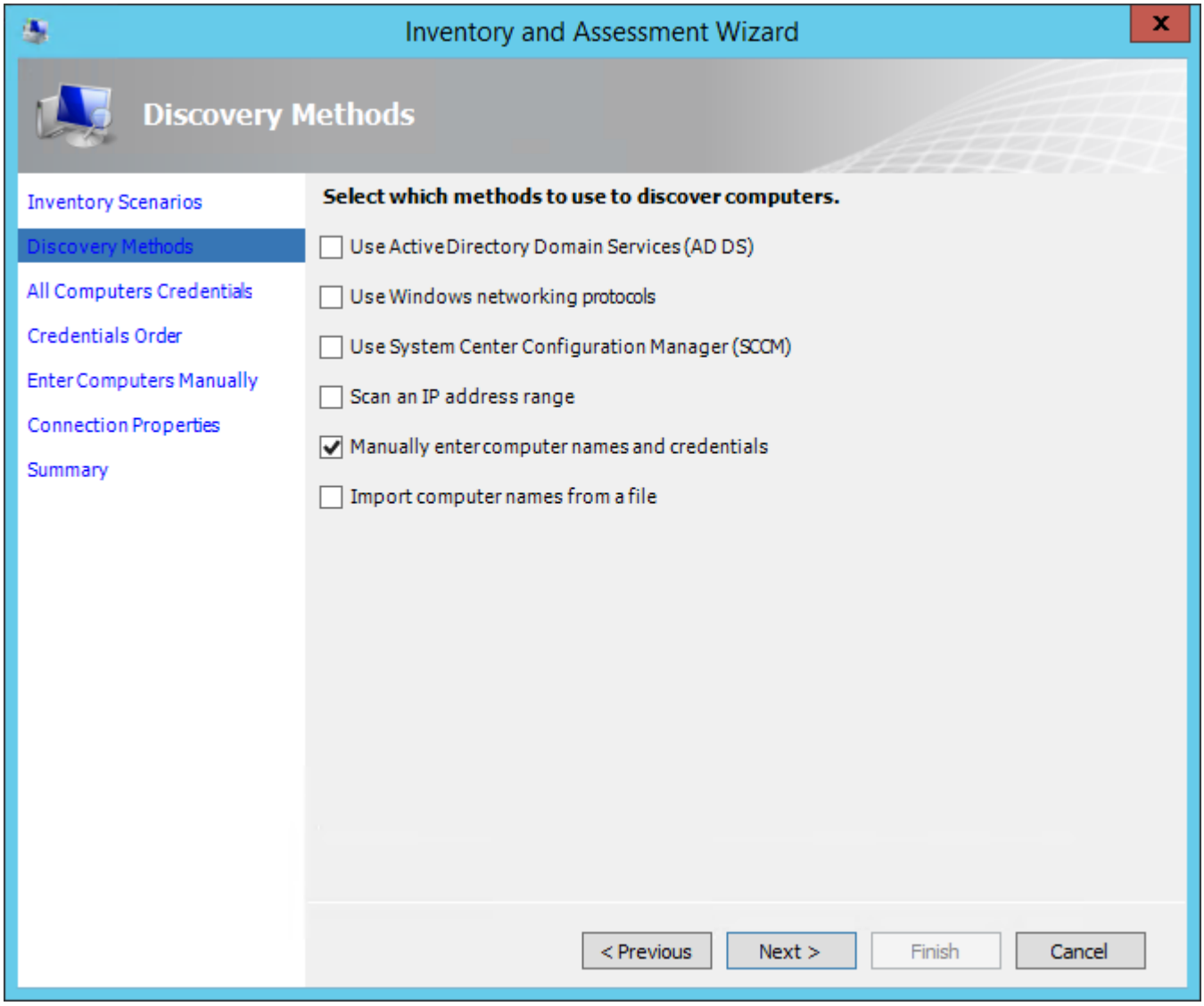


Abbildung - Manually enter...

Anmeldeinformationen eingeben:

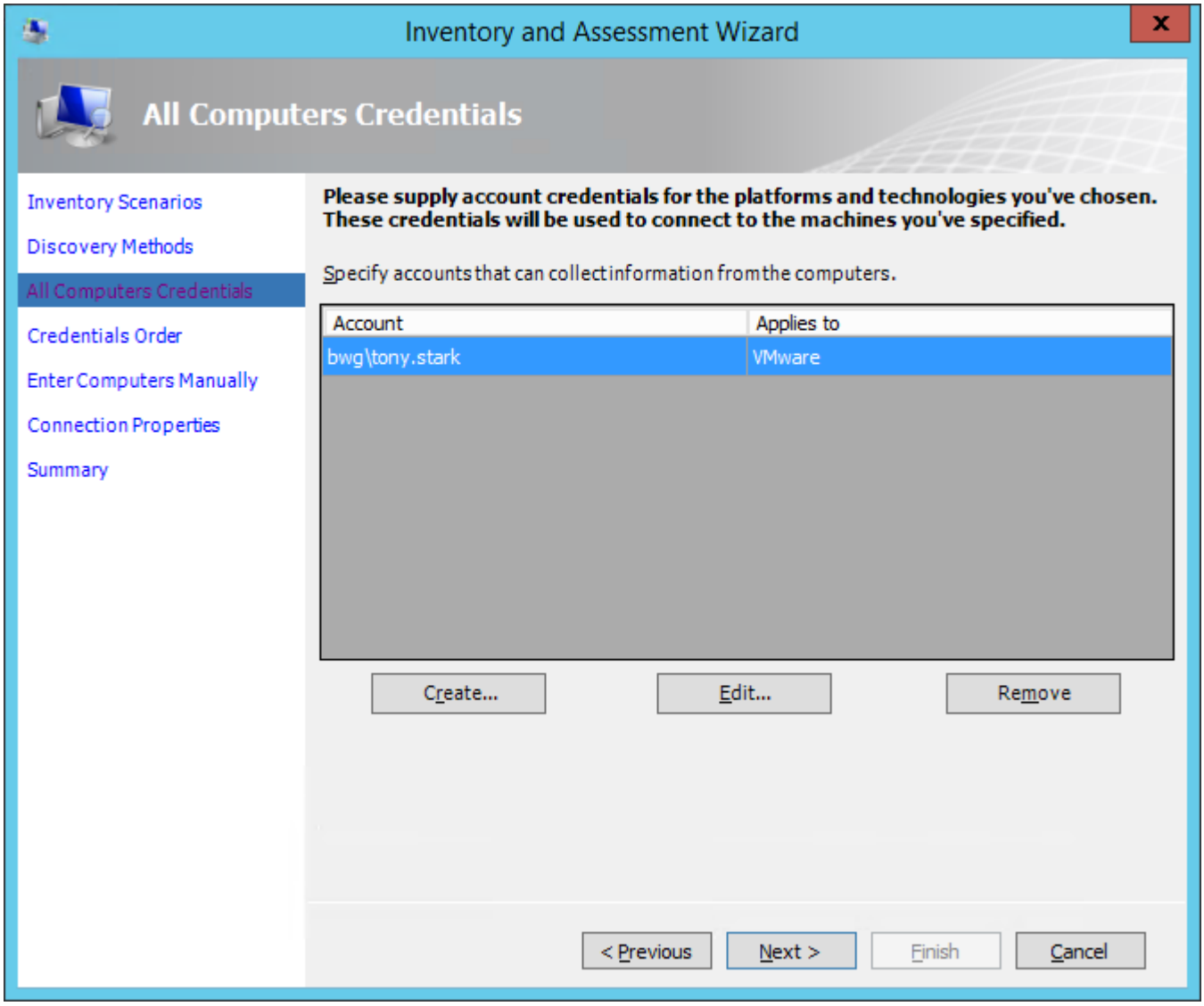


Abbildung - Credentials

Falls mehrere Anmeldinformationen vorhanden, Priorität festlegen.

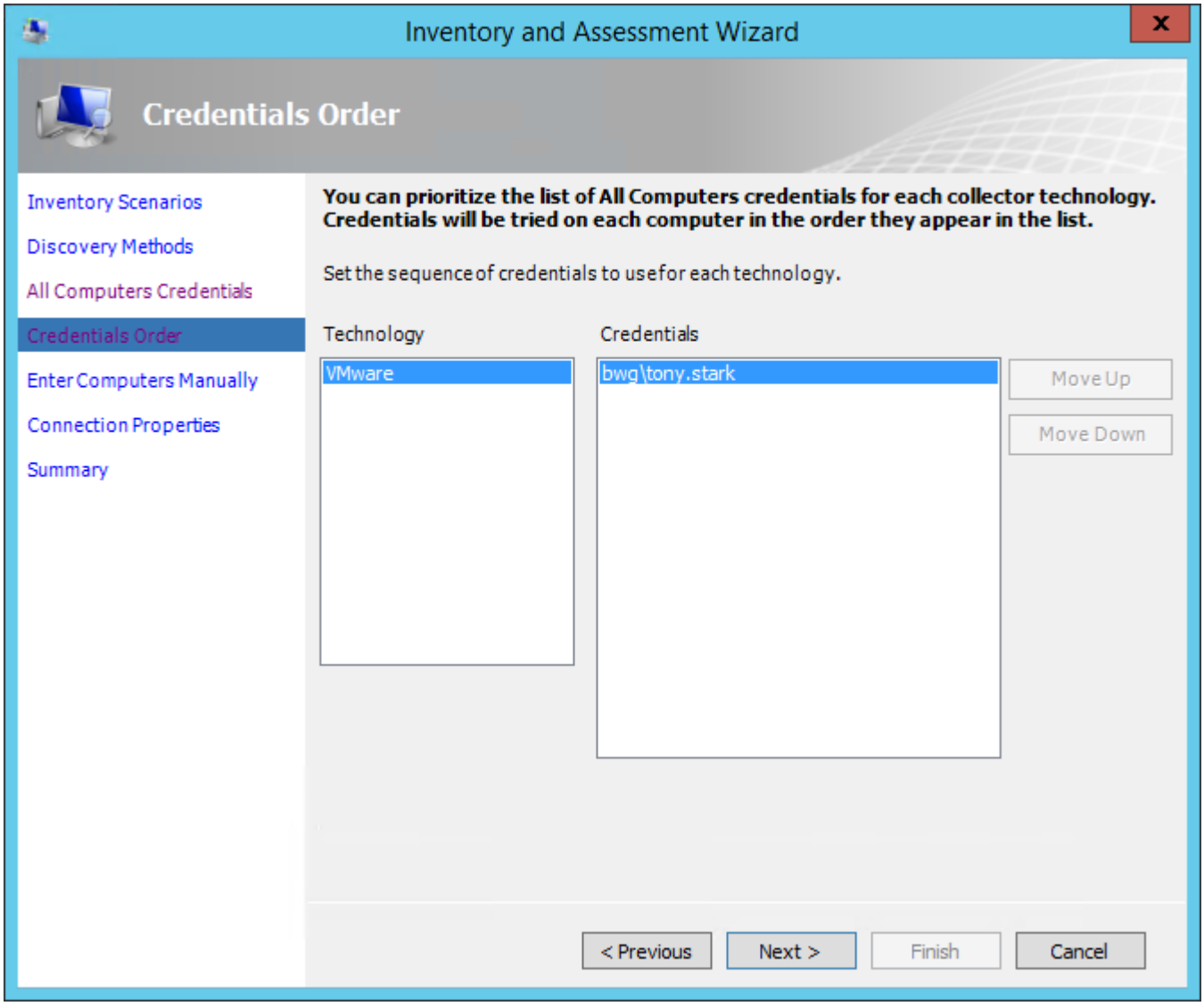


Abbildung - Credentials Order

Liste der vCenter Server angeben:

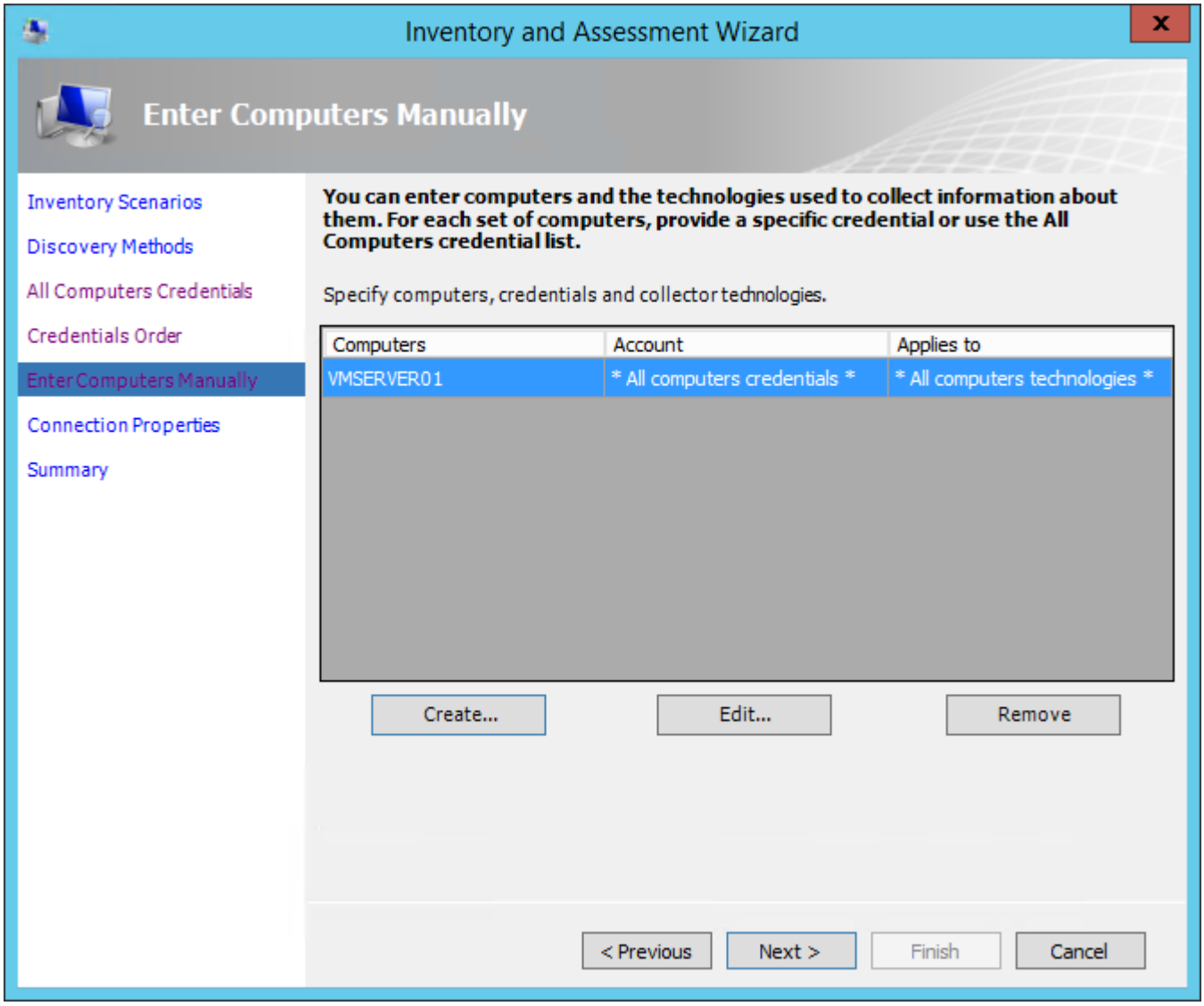


Abbildung - vCenter Servers

Eigenschaften der vCenter Server konfigurieren:

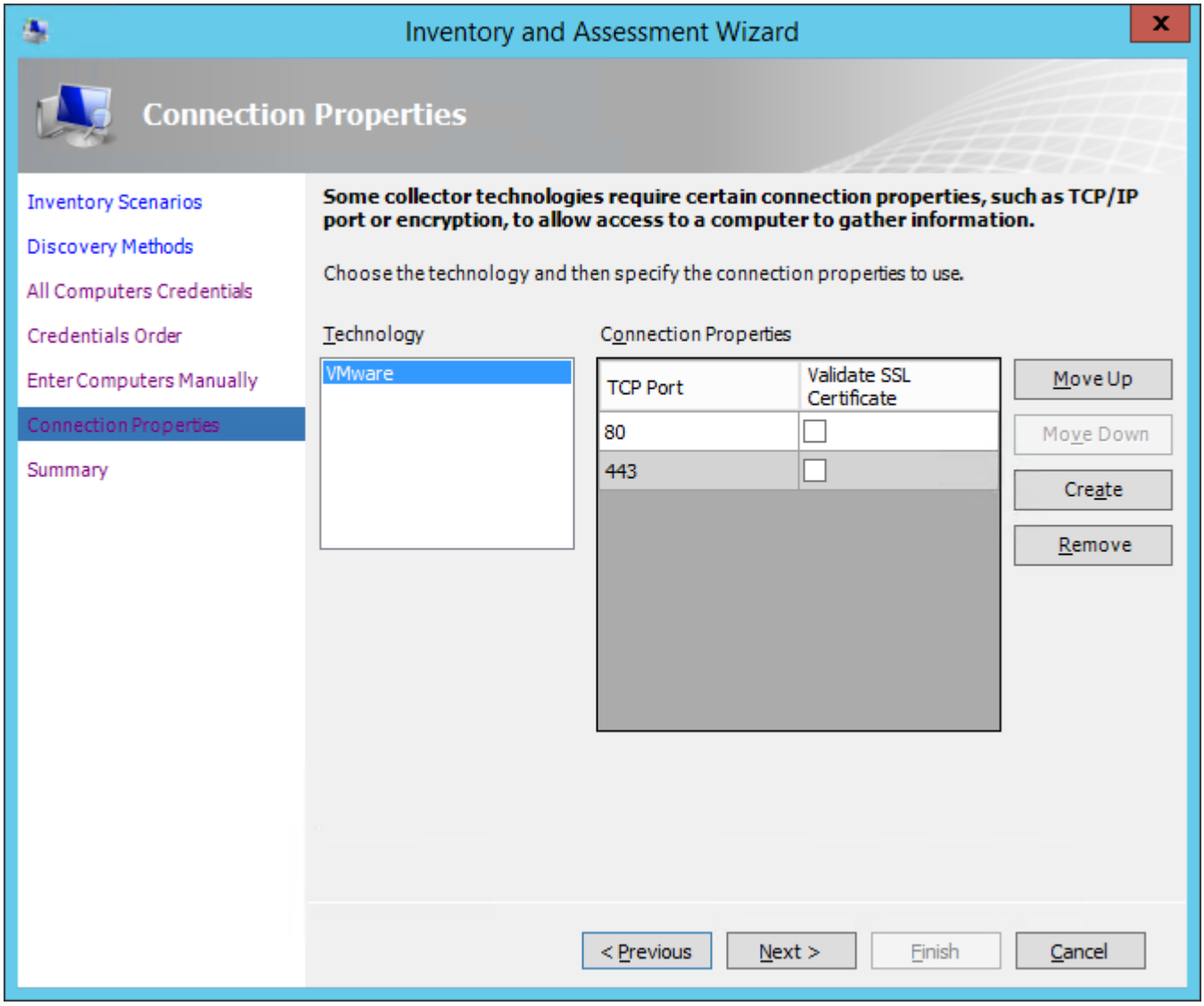


Abbildung - vCenter configuration

Überprüfen der Einstellungen

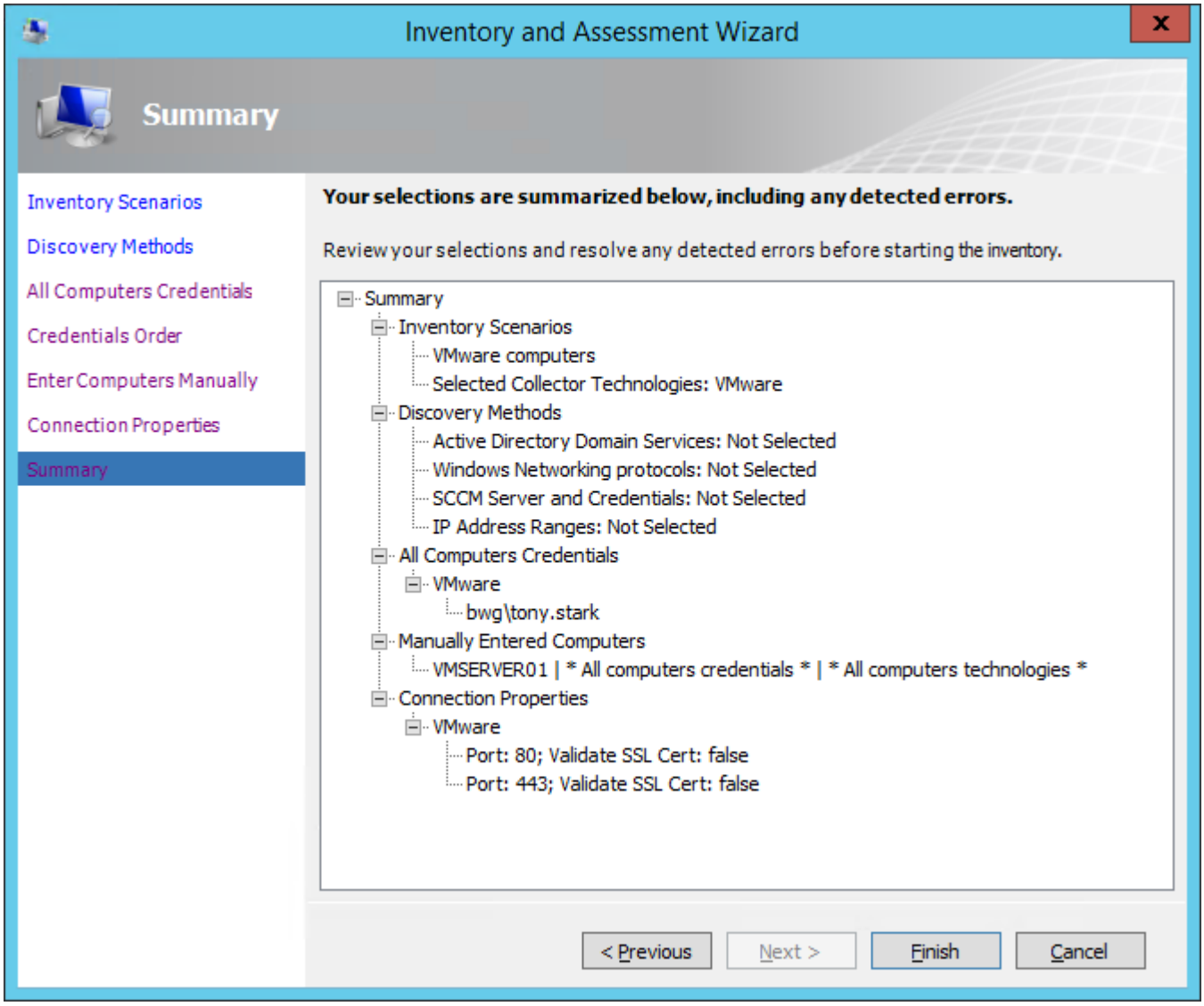


Abbildung - Summary

Nach dem das Erheben der Daten erledigt ist, kann nun damit gearbeitet werden.