



Zusatzprogramme (Telnet / SSH)
SNMP-Inventarisierung

TITEL	Zusatzprogramme (Telnet / SSH)
AUTOR	Docusnap Consulting
DATUM	11.08.2026
VERSION	2.2 gültig ab 11.08.2026

Die Weitergabe, sowie Vervielfältigung dieser Unterlage, auch von Teilen, Verwertung und Mitteilung ihres Inhaltes ist nicht gestattet, soweit nicht ausdrücklich durch die Docusnap GmbH zugestanden. Zuwiderhandlung verpflichtet zu Schadenersatz. Alle Rechte vorbehalten.

This document contains proprietary information and may not be reproduced in any form or parts whatsoever, nor may be used by or its contents divulged to third parties without written permission of Docusnap GmbH. All rights reserved.

INHALTSVERZEICHNIS

1.	EINLEITUNG	4
2.	ANWENDUNGSBEISPIEL ZUSATZPROGRAMME (TELNET / SSH)	5
3.	BEFEHLSSEQUENZEN	7
3.1	SSH-BEFEHLSKANAL	7
3.2	BEFEHLSSEQUENZ ERSTELLEN	8
3.3	ALLGEMEINE BEFEHLSERKLÄRUNG	10
3.4	SEQUENZ TESTEN	12

1. EINLEITUNG

Die Funktion Zusatzprogramme (Telnet / SSH) auslesen bietet die Möglichkeit mit zusätzlichen Befehlen Informationen von Geräten (SNMP) über Telnet oder SSH auszulesen und diese in die Dokumentation aufzunehmen. Dazu müssen sogenannte Befehlssequenzen erstellt werden, in denen die ausgeführten Befehle definiert werden.

Folgende Voraussetzungen müssen erfüllt sein.

- Das System muss bereits als **SNMP-System** inventarisiert worden sein
- Das System muss beim Auslesen der Konfiguration **erneut** als SNMP-System **inventarisiert** werden
- Zusatzprogramme (Telnet / SSH) im Optionen-Dialog muss aktiviert sein
- Eine Befehlssequenz muss für das entsprechende Gerät definiert worden sein

Die beschriebene Funktion ist eine „Experten-Funktion“. Damit können beliebige Befehle auf den SNMP-Systemen via Telnet / SSH ausgeführt werden, auch solche, die nicht nur Informationen auslesen, sondern den Zustand des Geräts oder bestehende Konfigurationen verändern.

Die in der Auslieferung enthaltenen Befehlssequenzen dienen zur Illustration.

2. ANWENDUNGSBEISPIEL ZUSATZPROGRAMME (TELNET / SSH)

Im folgenden Kapitel wird das Auslesen von Zusatzprogrammen anhand eines Zyxel GS2200-8 Switches mit dem SNMP-Namen AZY081 beschrieben. Dieser wurde vorab per SNMP ausgelesen. Eine dazugehörige Befehlssequenz befindet sich in der Datenbank. Diese wird im weiteren Verlauf genauer beschrieben.

Die Funktion Zusatzprogramme Telnet/SSH befindet sich im SNMP-Dialog. Diese Option muss zuerst im Optionen-Dialog aktiviert werden.

Optionen – Inventarisierung – Zusatzprogramme Telnet / SSH

Im Anschluss ist im SNMP-Assistenten ein neuer Schritt Zusatzprogramme Telnet / SSH verfügbar.

Durch Klicken auf **Hinzufügen** wird ein neuer Dialog geöffnet, über den alle Systeme konfiguriert werden können, die sich innerhalb des in Schritt 3 angegebenen SNMP-Suchbereichs befinden. Systeme außerhalb dieses Bereichs werden in der Auswahl nicht angezeigt.

Innerhalb dieses Dialogs besteht die Möglichkeit, die Verbindungen für die entsprechenden Sequenzen zu prüfen und die hinterlegten Anmeldeinformationen zu speichern.

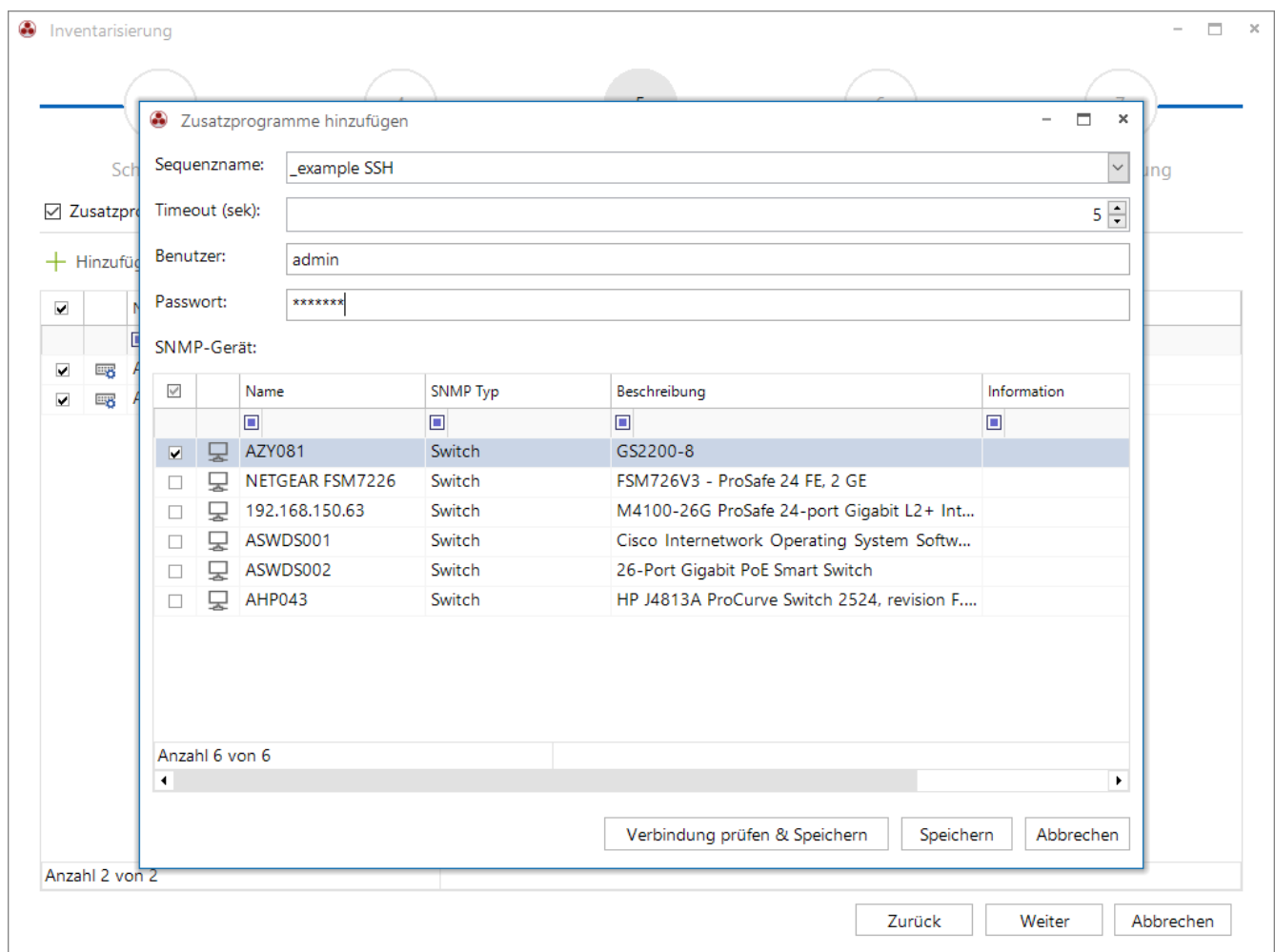


Abbildung 1 - SNMP Dialog - Zusatzprogramme

Nach erfolgreicher Inventarisierung befindet sich das Resultat der ausgewählten Befehlssequenz in der Datenbank und ist in der hierarchischen Struktur aufrufbar.

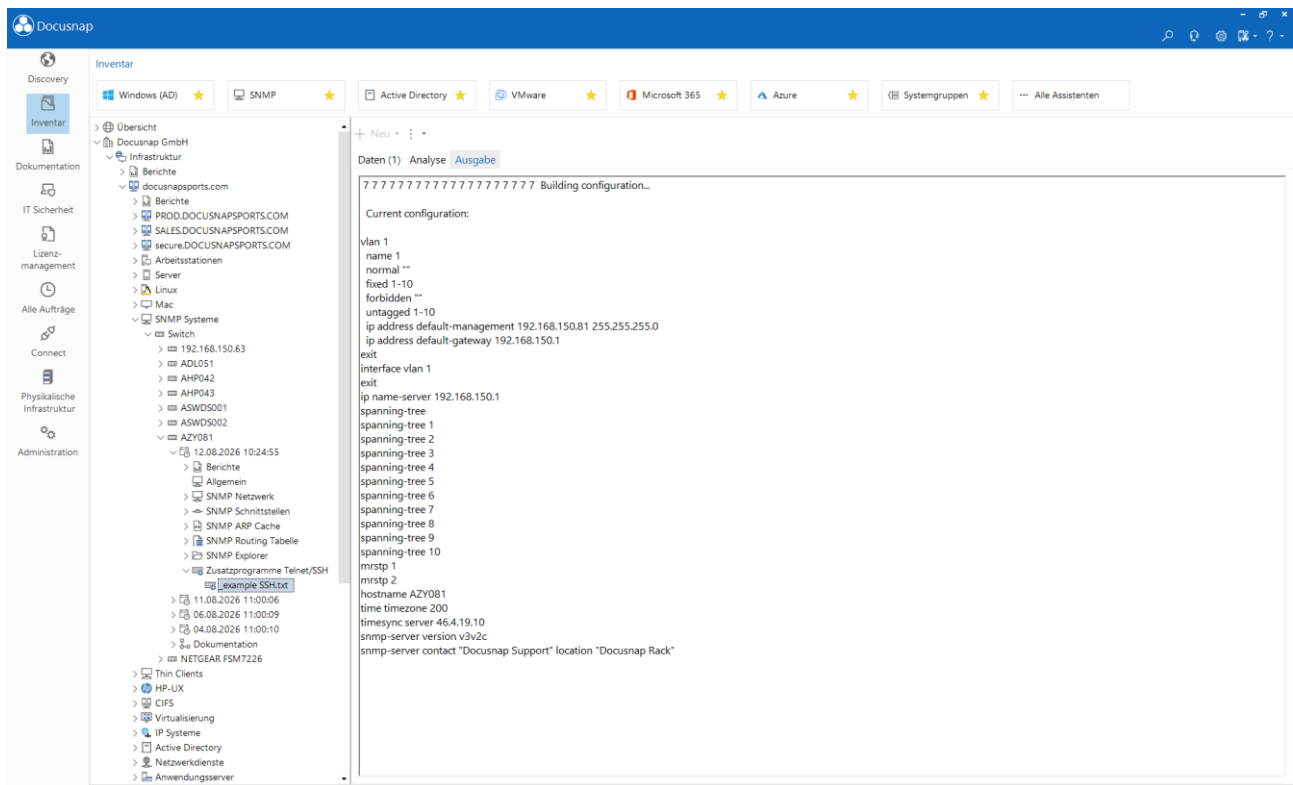


Abbildung 2 – hierarchische Struktur - Zusatzprogramme Ausgabe

3. BEFEHLSSEQUENZEN

Befehlssequenzen beschreiben die Befehle, die über Telnet / SSH ausgeführt werden sollen. Diese können in Docusnap in der Administration erstellt und bearbeitet werden.

Docusnap – Administration – Zusatzprogramme Telnet / SSH

Docusnap liefert dabei zwei beispielhafte Befehlssequenzen für einen SSH und Telnet Zugriff mit. Diese können nicht bearbeitet, aber kopiert und somit als Vorlage verwendet werden.

3.1 SSH-BEFEHLSKANAL

Mit Docusnap 15 ist die Option SSH-Befehlskanal verwenden hinzugekommen.

Ist die Option aktiviert, fordert Docusnap beim Verbindungsaufbau einen Befehlskanal an (im SSH-Protokoll ein exec-Request). Der Server führt den Befehl dabei in einem eigenen Kanal aus und schließt diesen nach Abschluss. Das Ende der Ausgabe ist damit eindeutig bestimmt, sodass in der Befehlssequenz keine Wait-Anweisungen und keine regulären Ausdrücke zur Erkennung des Prompts mehr erforderlich sind.

Lehnt der Server die Anforderung ab, greift die Verarbeitung automatisch auf die bisherige streambasierte Implementierung zurück – ein zusätzlicher Eingriff ist nicht notwendig. Dieser Fall tritt vor allem bei Netzwerkgeräten auf, die ausschließlich eine interaktive Shell bereitstellen. Für solche Systeme bleiben Wait-Anweisungen und reguläre Ausdrücke in der Befehlssequenz weiterhin zwingend erforderlich.

3.2 BEFEHLSSEQUENZ ERSTELLEN

Um die Erstellung einer Befehlssequenz zu beschreiben, wurde die vorhandene _example SSH-Sequenz dupliziert. Die folgende Abbildung zeigt die Kopie der Befehlssequenz _example SSH.

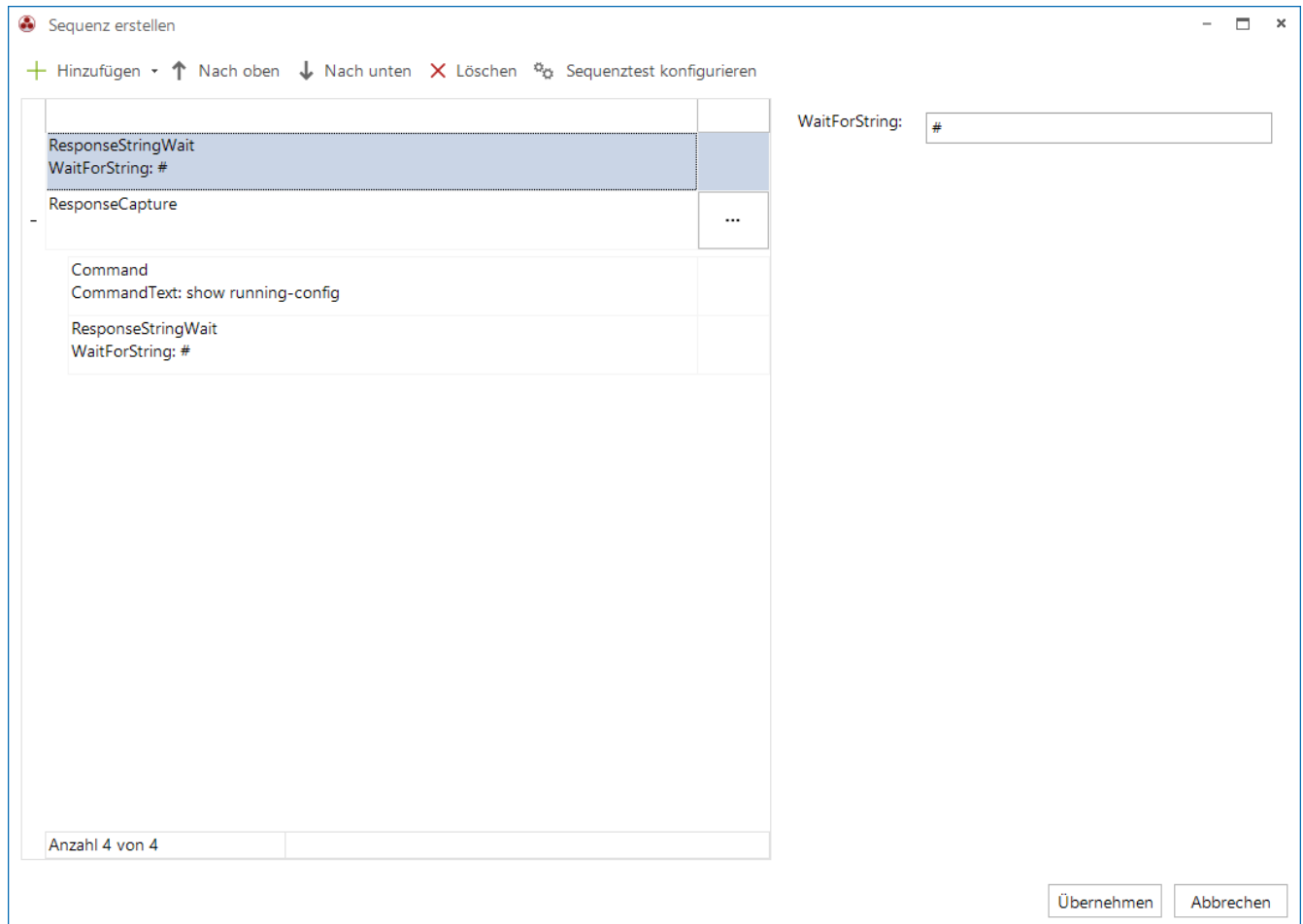


Abbildung 3 - Befehlssequenz erstellen

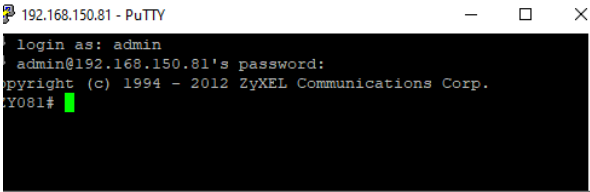
Die exemplarische Befehlssequenz besteht aus den folgenden Elementen. Diese werden im weiteren Verlauf genauer erklärt.

- ResponseStringWait
- ResponseCapture
 - Command
 - ResponseStringWait

Um die Sequenz zu beschreiben, wurde parallel eine SSH-Verbindung in der CMD aufgebaut.

- **ResponseStringWait – WaitForString #**

Dieser Schritt ist erfüllt, wenn in der Konsolenausgabe ein # erscheint. Da in der Konsole der String AZY081# erscheint, ist diese Bedingung erfüllt und es wird zum zweiten Schritt übergegangen.



```
192.168.150.81 - PuTTY
login as: admin
admin@192.168.150.81's password:
Copyright (c) 1994 - 2012 ZyXEL Communications Corp.
AZY081#
```

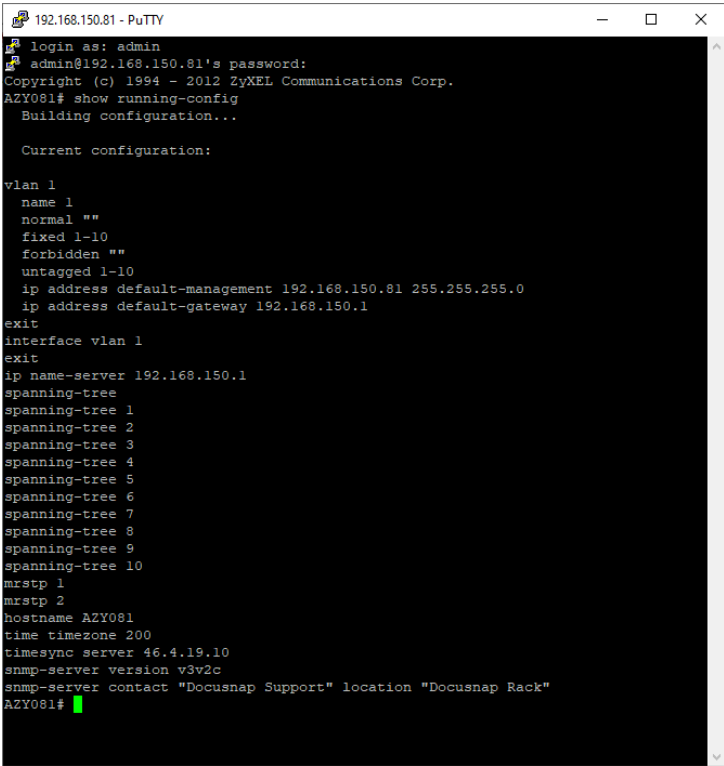
Abbildung 4 - WaitForString #

- **ResponseCapture**

Dadurch wird die Konsolenaufnahme gestartet. Sämtliche Konsolenausgabe innerhalb von ResponseCapture wird später unter Zusatzinformationen angezeigt. Befehle innerhalb der ResponseCapture werden über das dazugehörige + hinzugefügt.

- **Command**

Der Befehl *show running-config* wird nun auf dem Switch ausgeführt



```
192.168.150.81 - PuTTY
login as: admin
admin@192.168.150.81's password:
Copyright (c) 1994 - 2012 ZyXEL Communications Corp.
AZY081# show running-config
Building configuration...

Current configuration:

vlan 1
 name 1
 normal ""
 fixed 1-10
 forbidden ""
 untagged 1-10
 ip address default-management 192.168.150.81 255.255.255.0
 ip address default-gateway 192.168.150.1
exit
interface vlan 1
exit
ip name-server 192.168.150.1
spanning-tree
 spanning-tree 1
 spanning-tree 2
 spanning-tree 3
 spanning-tree 4
 spanning-tree 5
 spanning-tree 6
 spanning-tree 7
 spanning-tree 8
 spanning-tree 9
 spanning-tree 10
mrstp 1
mrstp 2
hostname AZY081
time timezone 200
timesync server 46.4.19.10
snmp-server version v3v2c
snmp-server contact "Docusnap Support" location "Docusnap Rack"
AZY081#
```

Abbildung 5 - Command - show running-config

- **ResponseStringWait – WaitForString #**

Im Anschluss wird wieder auf ein # in der Konsolenausgabe gewartet.

3.3 ALLGEMEINE BEFEHLSERKLÄRUNG

Die folgenden Befehle stehen für die Erstellung einer Sequenz zur Auswahl

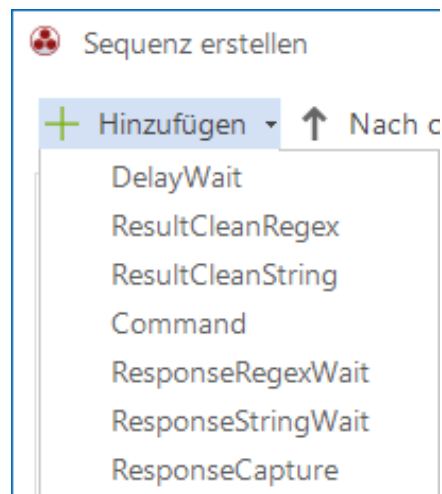


Abbildung 6 - Befehle hinzufügen

- **DelayWait**
Erzeugt eine Verzögerung, um bei Bedarf die Ausführung des nächsten Schrittes zu verzögern. Diese Verzögerung wird in Millisekunden angegeben.
- **ResponseRegexWait**
Wartet, bis der definierte Parameter in der Konsole angezeigt wird. Der Parameter für das Erwarten einer # wird in einem regulären Ausdruck wie folgt definiert:
WaitForPattern: .*#\s\$
- **ResponseStringWait**
Wartet, bis der definierte Parameter in der Konsole angezeigt wird. Der Parameter für das Erwarten einer # wird wie folgt definiert:
WaitForString: #

Es empfiehlt sich zwischen den einzelnen Befehlen eine **ResponseRegexWait** oder **ResponseStringWait** Überprüfung einzubauen, um die Stabilität zu erhöhen.

- **Command**
Dadurch wird auf dem Zielsystem ein Befehl ausgeführt wie z.B. *show running-config*. Es können aber nicht nur Befehle ausgeführt werden, die eine Rückgabe liefern, sondern auch solche, die eine erforderliche Eingabe umsetzen. Muss in der Konsole z.B. ein „any key“ übergeben werden, kann das damit erfolgen. Deshalb sollten die Befehle zuvor in der Konsole durchgespielt werden.
- **ResponseCapture**
Der Befehl startet die Aufzeichnung der Konsolenausgabe. Nur Ausgaben innerhalb des ResponseCapture werden aufgezeichnet – zuvor angelegte Befehle sind im Docusnap-Output nicht enthalten. Die darauffolgenden Befehle müssen als Subsequenz mit Hilfe des + angelegt werden. Als Abschlussbefehl wird empfohlen auf einen Konsolenprompt zu prüfen.

Die folgenden Befehle müssen innerhalb eines **ResponseCaptures** ausgeführt werden.

- **ResultCleanRegex**

Der Befehl identifiziert einen String mithilfe eines regulären Ausdrucks und ersetzt diesen. Sollen z.B. sämtliche Zeichen nach dem String Passwort mit ********* ersetzt werden muss folgender Ausdruck gewählt werden

RegexPattern: Passwort\s.*

ReplaceWith: *****

- **ResultCleanString**

Der Befehl findet einen String und ersetzt diesen. Die folgenden Parameter müssen übergeben werden, um z.B. den String „Community Geheim#2021“ mit „Community *********“ zu ersetzen.

Find: Community Geheim#2021

Replace: Community *****

3.4 SEQUENZ TESTEN

Um die Erstellung einer Befehlssequenz zu erleichtern, kann diese direkt getestet werden. Somit muss nicht bei jeder Änderung der Sequenz eine neue Inventarisierung durchgeführt werden, um diese zu testen.

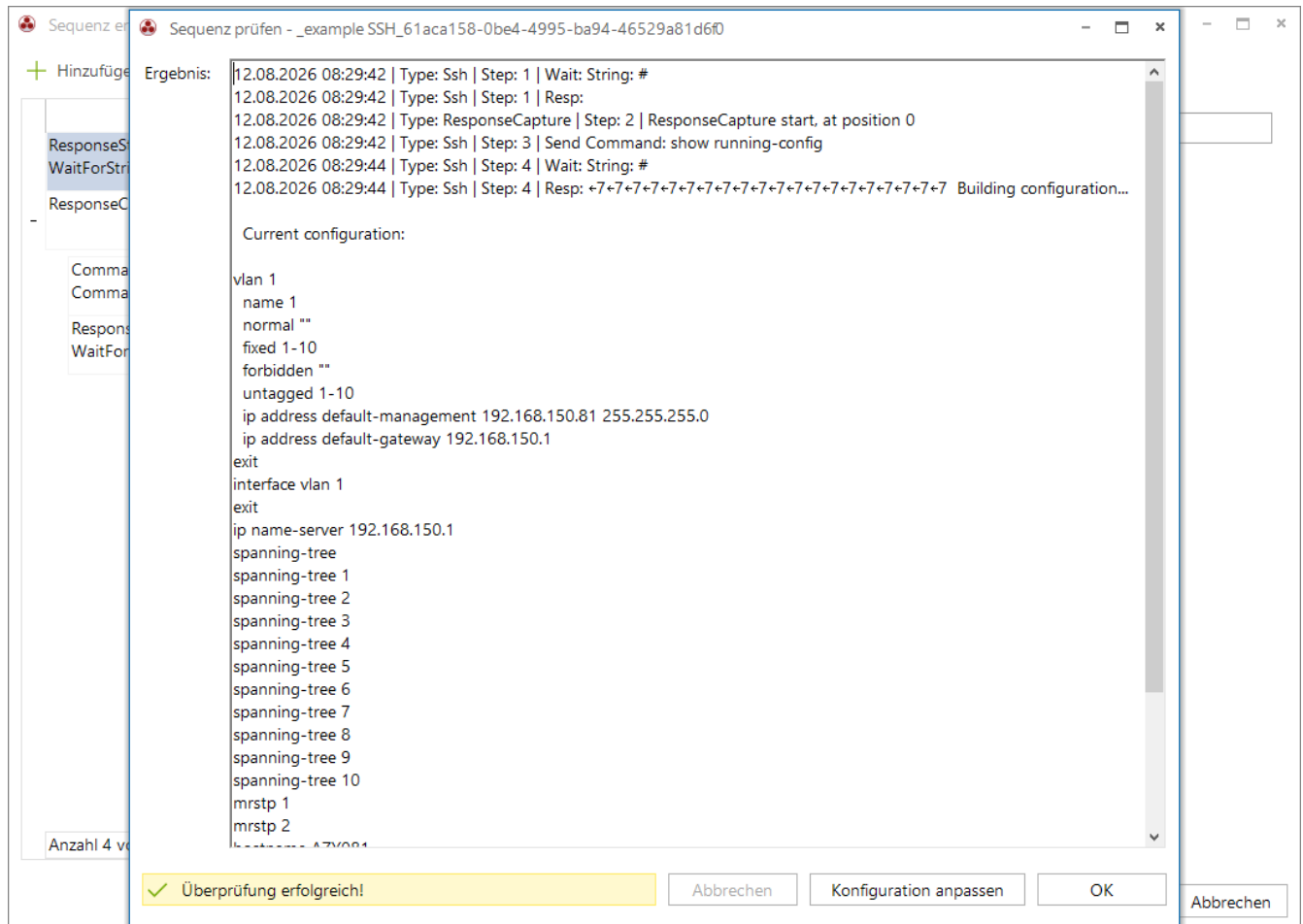


Abbildung 7 - Befehlssequenz testen

ABBILDUNGSVERZEICHNIS

ABBILDUNG 1 - SNMP DIALOG - ZUSATZPROGRAMME	5
ABBILDUNG 2 – HIERARCHISCHE STRUKTUR - ZUSATZPROGRAMME AUSGABE	6
ABBILDUNG 3 - BEFEHLSSEQUENZ ERSTELLEN	8
ABBILDUNG 4 - WAITFORSTRING #	9
ABBILDUNG 5 - COMMAND - SHOW RUNNING-CONFIG	9
ABBILDUNG 6 - BEFEHLE HINZUFÜGEN	10
ABBILDUNG 7 - BEFEHLSSEQUENZ TESTEN	12

VERSIONSHISTORIE

Datum	Beschreibung
24.11.2021	Erstellung des HowTos
05.12.2023	Version 2.0 - Überarbeitung des HowTos für Docusnap 13
28.04.2025	Version 2.1 – Einschränkung auf SNMP-Switche entfernt – Telnet / SSH Befehle können auf alle SNMP Systeme angewendet werden.
11.08.2026	Version 2.2 – SSH-Befehlskanal-Option
