



Additional Tools (Telnet / SSH)

SNMP Inventory

TITLE	Additional Tools (Telnet / SSH)
AUTHOR	Docusnap Consulting
DATE	2026-08-11
VERSION	2.2 valid from 2026-08-11

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.

TABLE OF CONTENTS

1. INTRODUCTION	4
2. EXAMPLE - ADDITIONAL TOOLS (TELNET / SSH)	5
3. COMMAND SEQUENCES	7
3.1 SSH COMMAND CHANNEL	7
3.2 CREATING A COMMAND SEQUENCE	8
3.3 GENERAL COMMAND REFERENCE	10
3.4 TESTING A SEQUENCE	12

1. INTRODUCTION

The Additional Tools (Telnet / SSH) function allows you to retrieve information from SNMP devices via Telnet or SSH by means of additional commands and to include it in the documentation. To do so, so-called command sequences have to be created, in which the commands to be executed are defined.

The following prerequisites must be met.

- The system must already have been inventoried as an **SNMP system**
- The system must be inventoried **again** as an SNMP system when the configuration is retrieved
- Additional Tools (Telnet / SSH) must be enabled in the Options - Inventory dialog
- A command sequence must be defined for the respective device

The function described here is an “expert function”. It can be used to run any command on the SNMP systems via Telnet / SSH – including commands that do not only retrieve information but change the state of the device or existing configurations.

The command sequences included in the delivery are intended as examples.

2. EXAMPLE - ADDITIONAL TOOLS (TELNET / SSH)

The following chapter describes the retrieval of additional tools using a Zyxel GS2200-8 switch with the SNMP name AZY081 as an example. This switch was inventoried via SNMP beforehand. A matching command sequence is available in the database. It is described in more detail below.

The Additional Tools Telnet/SSH function is located in the SNMP dialog. This option must first be enabled in the Options - Inventory dialog.

Options – Inventory – Additional Tools Telnet / SSH

A new Additional Tools Telnet / SSH step is then available in the SNMP wizard.

Clicking **Add** opens a new dialog in which all systems located within the SNMP scan range specified in step 3 can be configured. Systems outside this range are not shown in the selection.

Within this dialog, the connections for the respective sequences can be checked and the stored credentials can be saved.

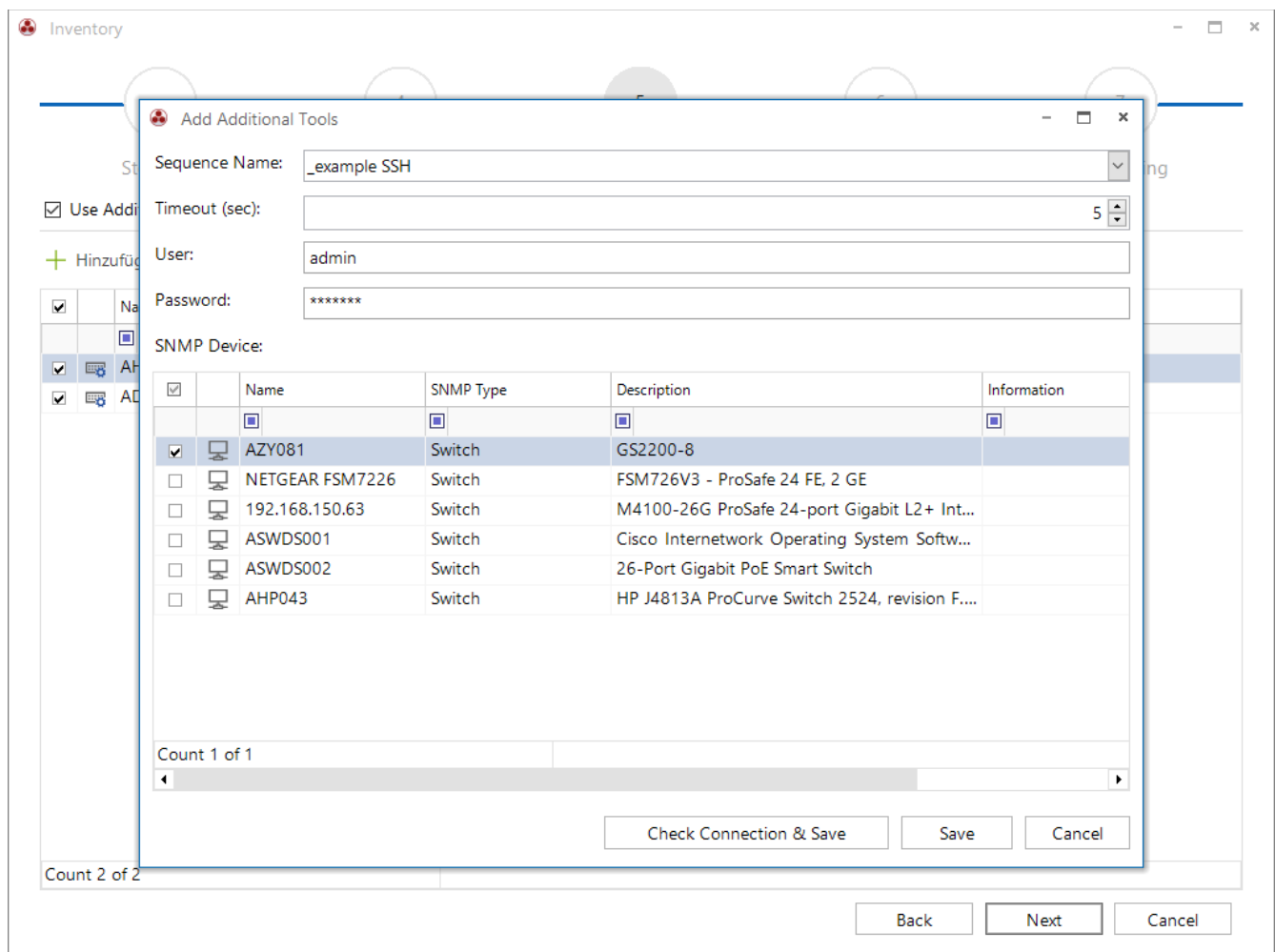


Figure 1 - SNMP dialog - Additional Tools

After a successful inventory, the result of the selected command sequence is stored in the database and can be accessed in the hierarchical structure.

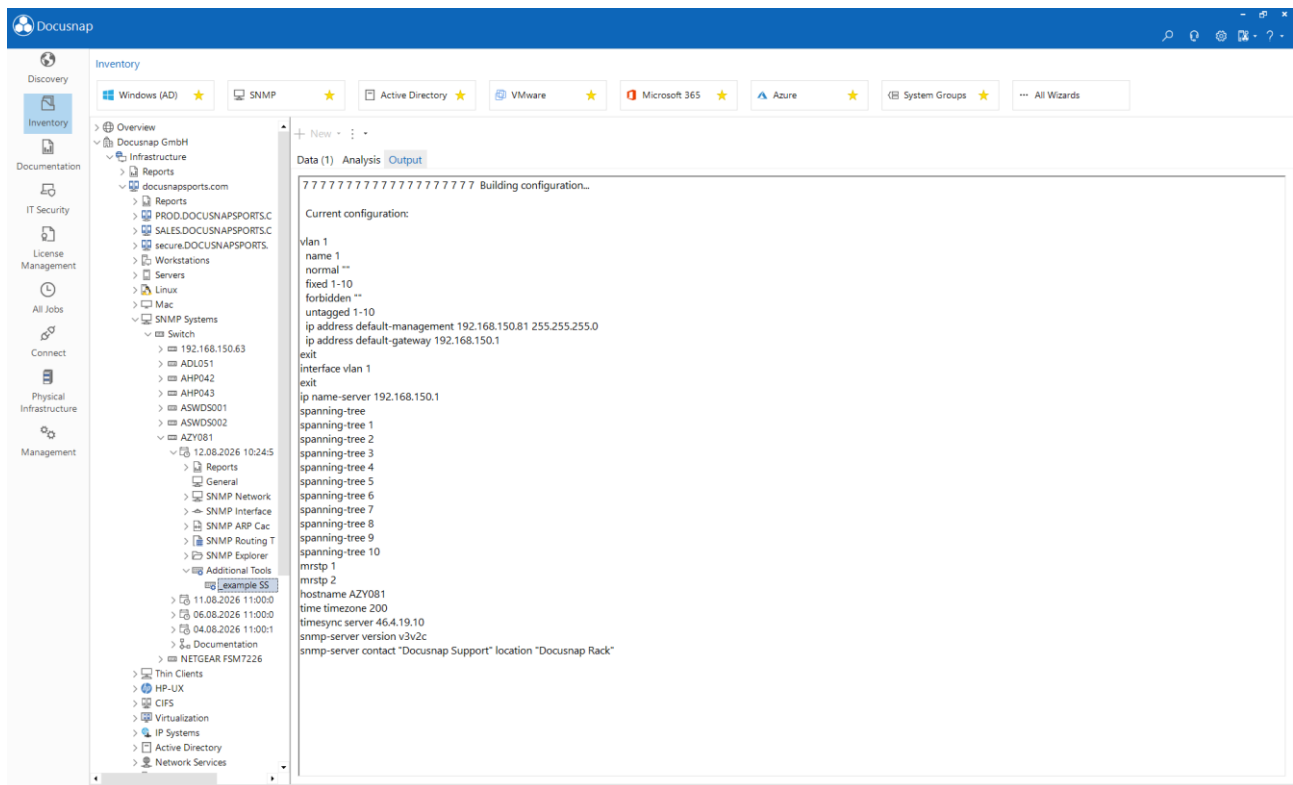


Figure 2 – Hierarchical structure - Additional Tools output

3. COMMAND SEQUENCES

Command sequences describe the commands that are to be executed via Telnet / SSH. They can be created and edited in the Docusnap Management.

Docusnap – Management – Additional Tools Telnet / SSH

Docusnap ships with two sample command sequences, one for SSH and one for Telnet access. They cannot be edited, but they can be copied and used as a template.

3.1 SSH COMMAND CHANNEL

The Use SSH command channel option was added in Docusnap 15.

If the option is enabled, Docusnap requests a command channel when the connection is established (an exec request in the SSH protocol). The server runs the command in a dedicated channel and closes it once execution has finished. The end of the output is therefore unambiguous, which means that wait statements and regular expressions for detecting the prompt are no longer required in the command sequence.

If the server rejects the request, processing automatically falls back to the previous stream-based implementation – no further action is required. This mainly happens with network devices that only provide an interactive shell. For such systems, wait statements and regular expressions remain mandatory in the command sequence.

3.2 CREATING A COMMAND SEQUENCE

To illustrate how a command sequence is created, the existing _example SSH sequence was duplicated. The following figure shows the copy of the _example SSH command sequence.

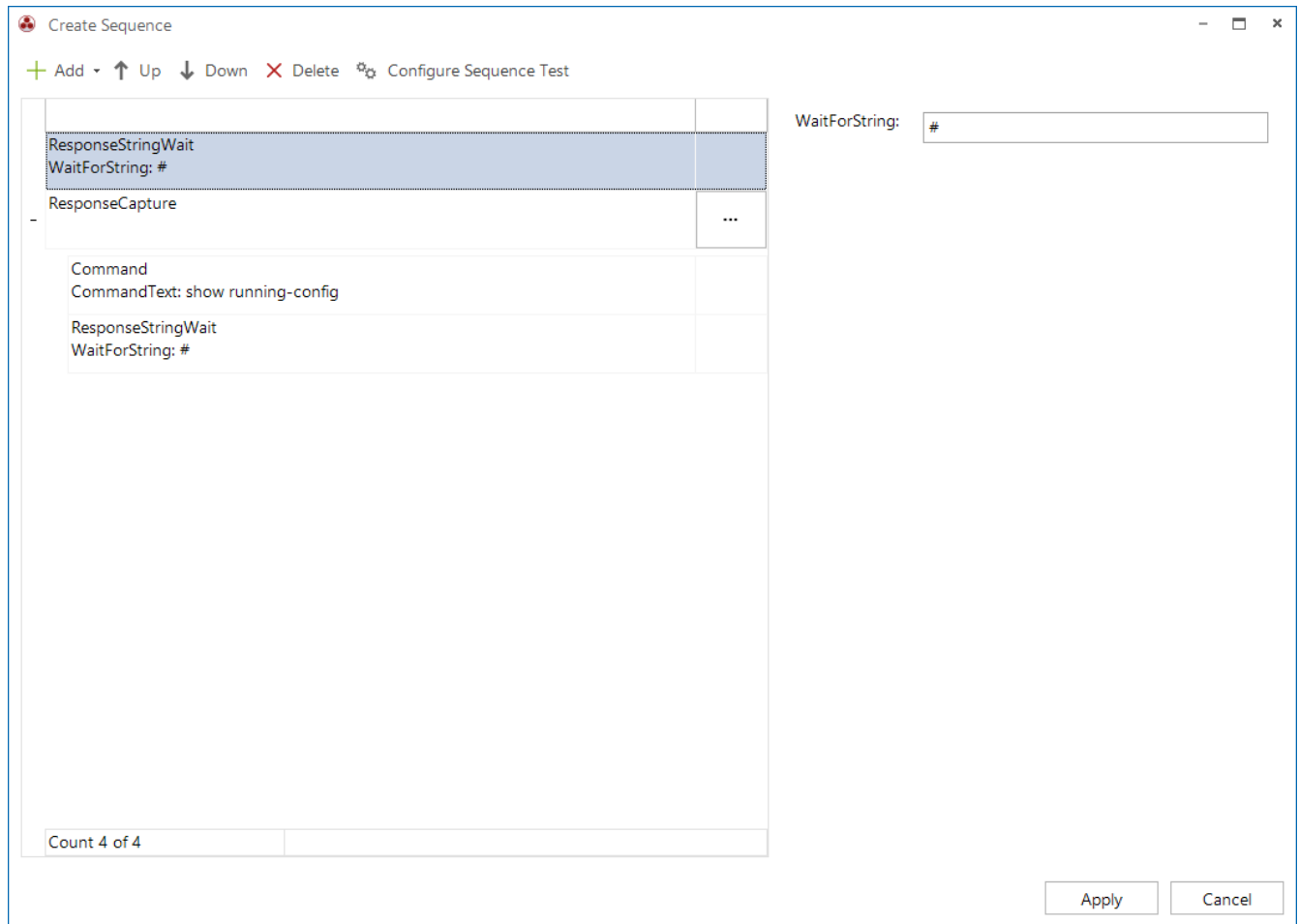


Figure 3 - Creating a command sequence

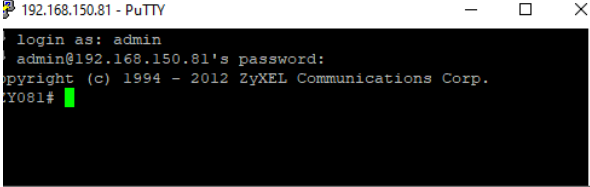
The sample command sequence consists of the following elements. They are explained in more detail below.

- ResponseStringWait
- ResponseCapture
 - Command
 - ResponseStringWait

To describe the sequence, an SSH connection was established in parallel using the command prompt (CMD).

- **ResponseStringWait – WaitForString #**

This step is fulfilled as soon as a # appears in the console output. Since the string AZY081# appears in the console, the condition is met and processing continues with the second step.



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

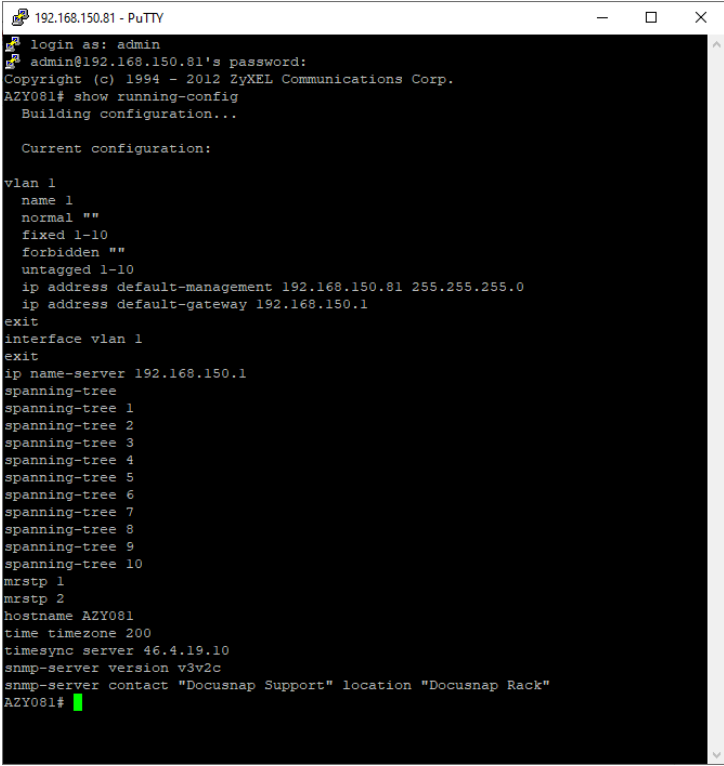
Figure 4 - WaitForString #

- **ResponseCapture**

This starts the console capture. All console output within the ResponseCapture is displayed later under Additional Information. Commands inside the ResponseCapture are added using the corresponding + sign.

- **Command**

The *show running-config* command is now executed on the switch



```
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#
```

Figure 5 - Command - show running-config

- **ResponseStringWait – WaitForString #**

Afterwards, the sequence waits for a # in the console output again.

3.3 GENERAL COMMAND REFERENCE

The following commands are available for creating a sequence

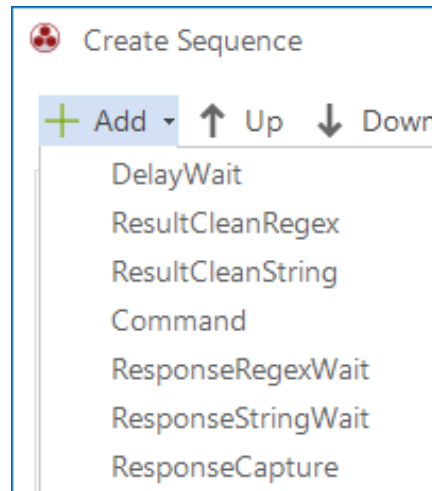


Figure 6 - Adding commands

- **DelayWait**
Creates a delay in order to postpone the execution of the next step if required. The delay is specified in milliseconds.
- **ResponseRegexWait**
Waits until the defined parameter is displayed in the console. The parameter for expecting a # is defined as a regular expression as follows:
WaitForPattern: .*#\s\$
- **ResponseStringWait**
Waits until the defined parameter is displayed in the console. The parameter for expecting a # is defined as follows:
WaitForString: #

It is recommended to add a **ResponseRegexWait** or **ResponseStringWait** check between the individual commands in order to increase stability.

- **Command**
This executes a command on the target system, e.g. *show running-config*. However, it is not only possible to run commands that return a result, but also commands that provide a required input. If, for example, an "any key" input has to be sent in the console, this can be done with Command. The commands should therefore be run through in the console beforehand.
- **ResponseCapture**
This command starts the recording of the console output. Only output produced within the ResponseCapture is recorded – commands created before it are not included in the Docusnap output. Subsequent commands must be created as a subsequence using the + sign. As the final command, checking for a console prompt is recommended.

The following commands must be executed within a **ResponseCapture**.

- **ResultCleanRegex**

This command identifies a string by means of a regular expression and replaces it. If, for example, all characters after the string password are to be replaced with *********, the following expression has to be used

RegexPattern: password\.*

ReplaceWith: *****

- **ResultCleanString**

This command finds a string and replaces it. The following parameters have to be passed in order to replace the string "Community Geheim#2021" with "Community *********", for example.

Find: Community Geheim#2021

Replace: Community *****

3.4 TESTING A SEQUENCE

To make creating a command sequence easier, the sequence can be tested directly. This means that a new inventory does not have to be run every time the sequence is changed in order to test it.

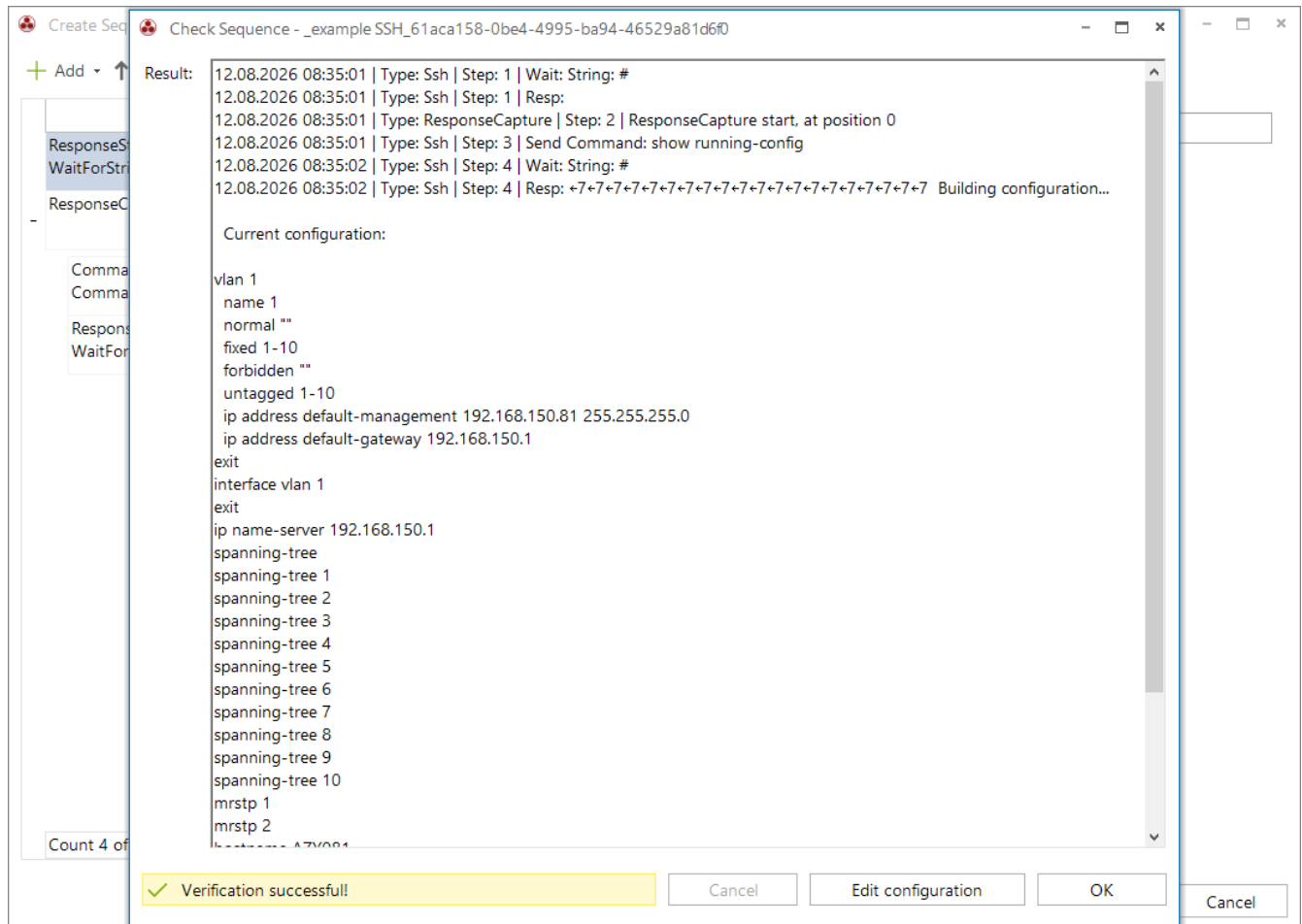


Figure 7 - Testing a command sequence

TABLE OF FIGURES

FIGURE 1 - SNMP DIALOG - ADDITIONAL TOOLS.....	5
FIGURE 2 – HIERARCHICAL STRUCTURE - ADDITIONAL TOOLS OUTPUT	6
FIGURE 3 - CREATING A COMMAND SEQUENCE	8
FIGURE 4 - WAITFORSTRING #	9
FIGURE 5 - COMMAND - SHOW RUNNING-CONFIG	9
FIGURE 6 - ADDING COMMANDS	10
FIGURE 7 - TESTING A COMMAND SEQUENCE	12

VERSION HISTORY

Date	Description
2021-11-24	Initial creation of the HowTo
2023-12-05	Version 2.0 - HowTo revised for Docusnap 13
2025-04-28	Version 2.1 – Restriction to SNMP switches removed – Telnet / SSH commands can be applied to all SNMP systems.
2026-08-11	Version 2.2 – SSH command channel option
