



## Additional Tools (Telnet / SSH)

*SNMP inventory*

|                |                                 |
|----------------|---------------------------------|
| <b>TITLE</b>   | Additional Tools (Telnet / SSH) |
| <b>AUTHOR</b>  | Docusnap Consulting             |
| <b>DATE</b>    | 12/12/2023                      |
| <b>VERSION</b> | 2.0   valid from 12/05/2023     |

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.

## CONTENTS

|   |    |
|---|----|
| 1. INTRODUCTION                             | 4  |
| 2. USE CASE ADDITIONAL TOOLS (TELNET / SSH) | 5  |
| 3. COMMAND SEQUENCES                        | 7  |
| 3.1 CREATE COMMAND SEQUENCES                | 7  |
| 3.2 GENERAL COMMAND STATEMENTS              | 9  |
| 3.3 TEST SEQUENCES                          | 11 |

## 1. INTRODUCTION

The function Additional Tools (Telnet / SSH) offers the possibility to get information from devices (SNMP) via Telnet or SSH with additional commands and to include them in the documentation. For this purpose, so-called command sequences must be created in which the executed commands are defined.

Additional Tools can only be retrieved from systems that have been **inventoried as an SNMP device and recognized as a switch (SNMP type 58)**. This functionality is currently not available for other systems that can be accessed via Telnet or SSH! Therefore, in the further course of the document exclusively switches are spoken of.

The following requirements must be met.

- The **switch** must have already been inventoried as an **SNMP system**
- The **switch** must have been detected as an **SNMP type** switch
- The **switch** must be **re-inventoried** as an SNMP system when reading out the configuration
- Additional Tools (Telnet / SSH) in the options dialog must be activated
- A command sequence must have been defined for the corresponding device

The function described is an "expert function". This allows any commands to be executed on the switches, including those that not only retrieve information, but also change the state of the switch or existing configurations.

The command sequences included in the delivery are for illustration purposes.

## 2. USE CASE ADDITIONAL TOOLS (TELNET / SSH)

The following chapter describes the retrieval of Additional Tools using a Zyxel GS2200-8 switch with the SNMP name AZY081. This was inventoried in advance via SNMP and is successfully inventoried as a switch in the database. A corresponding command sequence is in the database. This is described in more detail in the following.

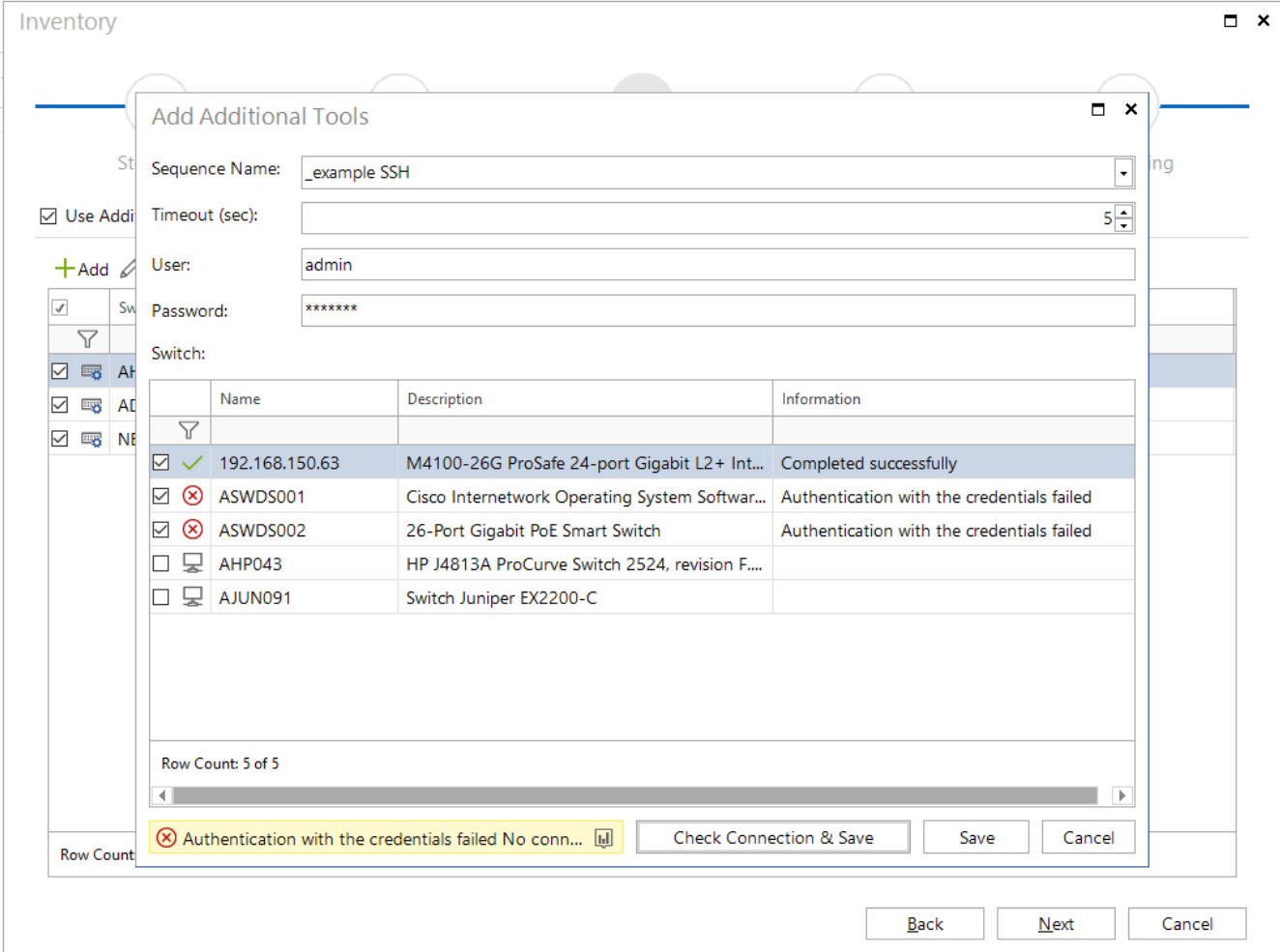
The function Additional Tools Telnet/SSH is located in the SNMP dialog. This option must first be activated in the Options dialog.

### Options - Inventory - Additional Tools Telnet / SSH

Afterwards, a new step Additional Tools Telnet / SSH is available in the SNMP Wizard.

Clicking **Add** opens a new dialog that can be used to configure all systems that are within the SNMP search range specified in step 3. Systems outside this range are not displayed in the selection.

Within this dialog there is the possibility to check the connections for the corresponding sequences and to save the stored credentials.



**Add Additional Tools**

Sequence Name:

Timeout (sec):

User:

Password:

Switch:

|                                     | Name           | Description                                    | Information                                |
|-------------------------------------|----------------|--|--|
| <input checked="" type="checkbox"/> | 192.168.150.63 | M4100-26G ProSafe 24-port Gigabit L2+ Int...   | Completed successfully                     |
| <input checked="" type="checkbox"/> | ASWDS001       | Cisco Internetwork Operating System Softwar... | Authentication with the credentials failed |
| <input checked="" type="checkbox"/> | ASWDS002       | 26-Port Gigabit PoE Smart Switch               | Authentication with the credentials failed |
| <input type="checkbox"/>            | AHP043         | HP J4813A ProCurve Switch 2524, revision F.... |  |
| <input type="checkbox"/>            | AJUN091        | Switch Juniper EX2200-C                        |  |

Row Count: 5 of 5

⚠ Authentication with the credentials failed No conn...

Figure 1 - SNMP Dialog - Additional Tools

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

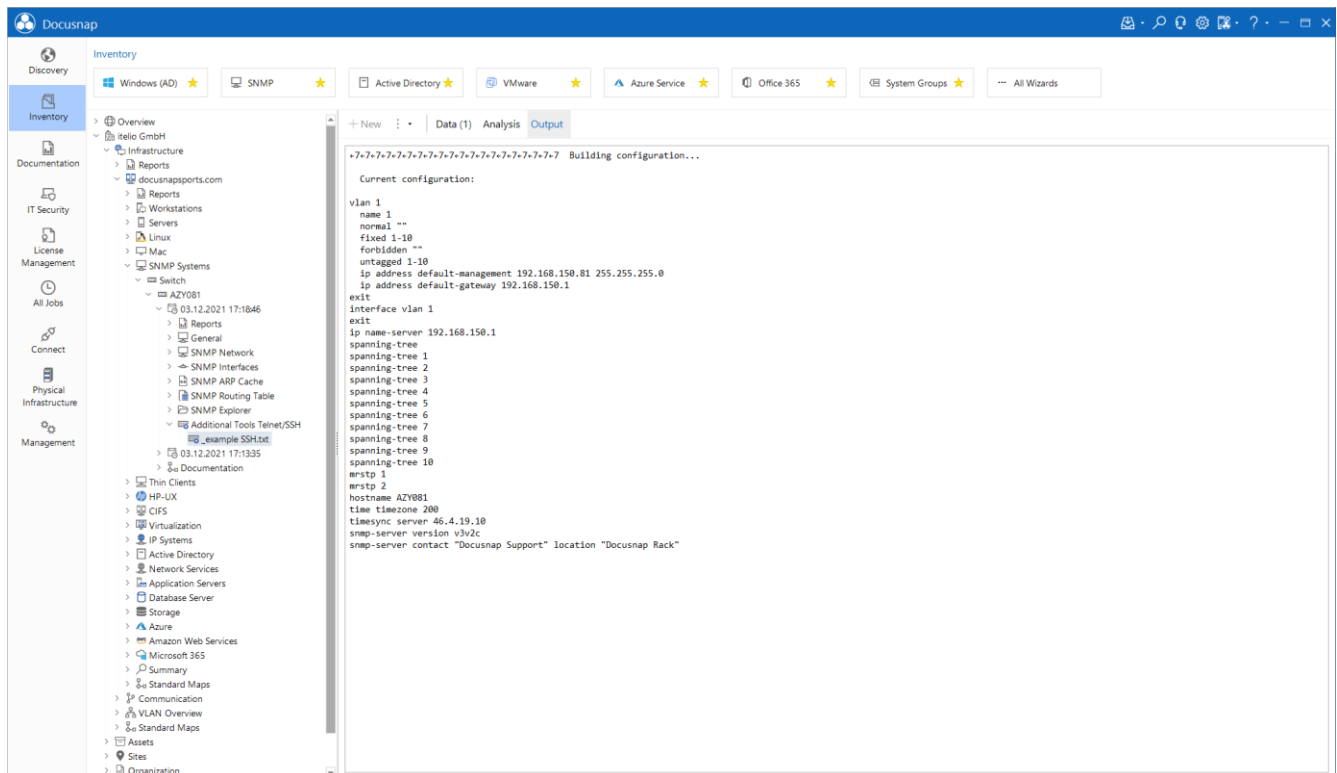


Figure 2 - hierarchical Structure - Additional Tools output

### 3. COMMAND SEQUENCES

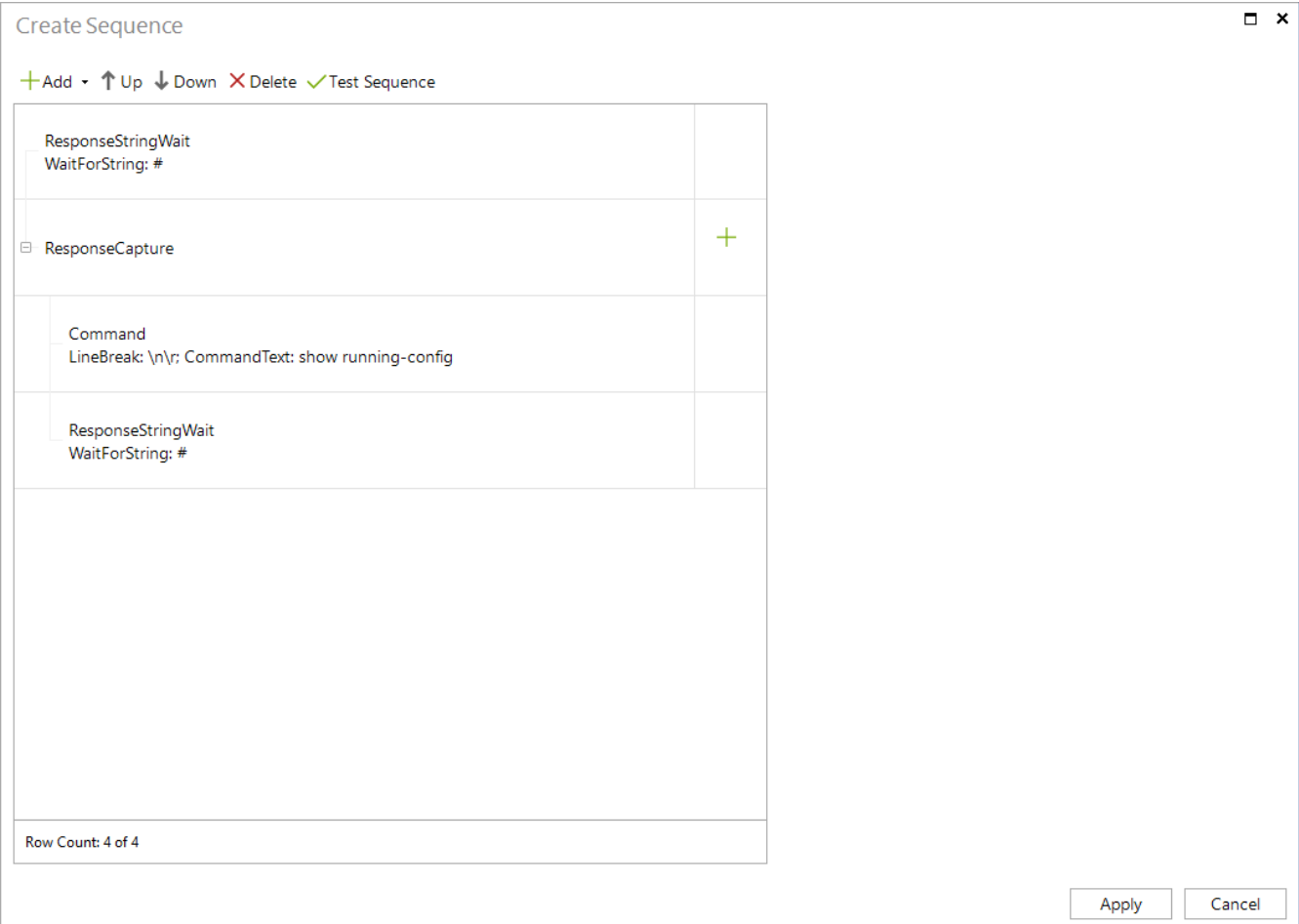
Command sequences describe the commands to be executed via Telnet / SSH. These can be created and edited in Docusnap in the management area.

#### Docusnap – Management - Additional Tools Telnet / SSH

Docusnap provides two sample command sequences for SSH and Telnet access. These cannot be edited but can be copied and thus used as a template.

#### 3.1 CREATE COMMAND SEQUENCES

To describe the creation of a command sequence, the existing \_example SSH sequence was duplicated. The following figure shows the copy of the \_example SSH command sequence.



| Create Sequence  |   |
|--|---|
| + Add   ↑ Up   ↓ Down   ✗ Delete   ✓ Test Sequence           |   |
| ResponseStringWait<br>WaitForString: #                       |   |
| ResponseCapture  | + |
| Command<br>LineBreak: \n\r; CommandText: show running-config |   |
| ResponseStringWait<br>WaitForString: #                       |   |
| Row Count: 4 of 4  |   |
| <div>Apply   Cancel</div>                                    |   |

Figure 3 - Create Sequence

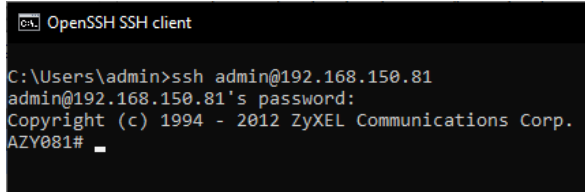
The exemplary command sequence consists of the following elements. These will be explained in more detail later.

- ResponseStringWait
- Response Capture
  - Command
  - ResponseStringWait

To describe the sequence, an SSH connection was established in parallel in the CMD.

- **ResponseStringWait – WaitForString #**

This step is fulfilled if a # appears in the console output. Since the string AZY081# appears in the console, this condition is done, and we go to the second step.



```

C:\Users\admin>ssh admin@192.168.150.81
admin@192.168.150.81's password:
Copyright (c) 1994 - 2012 ZyXEL Communications Corp.
AZY081#
  
```

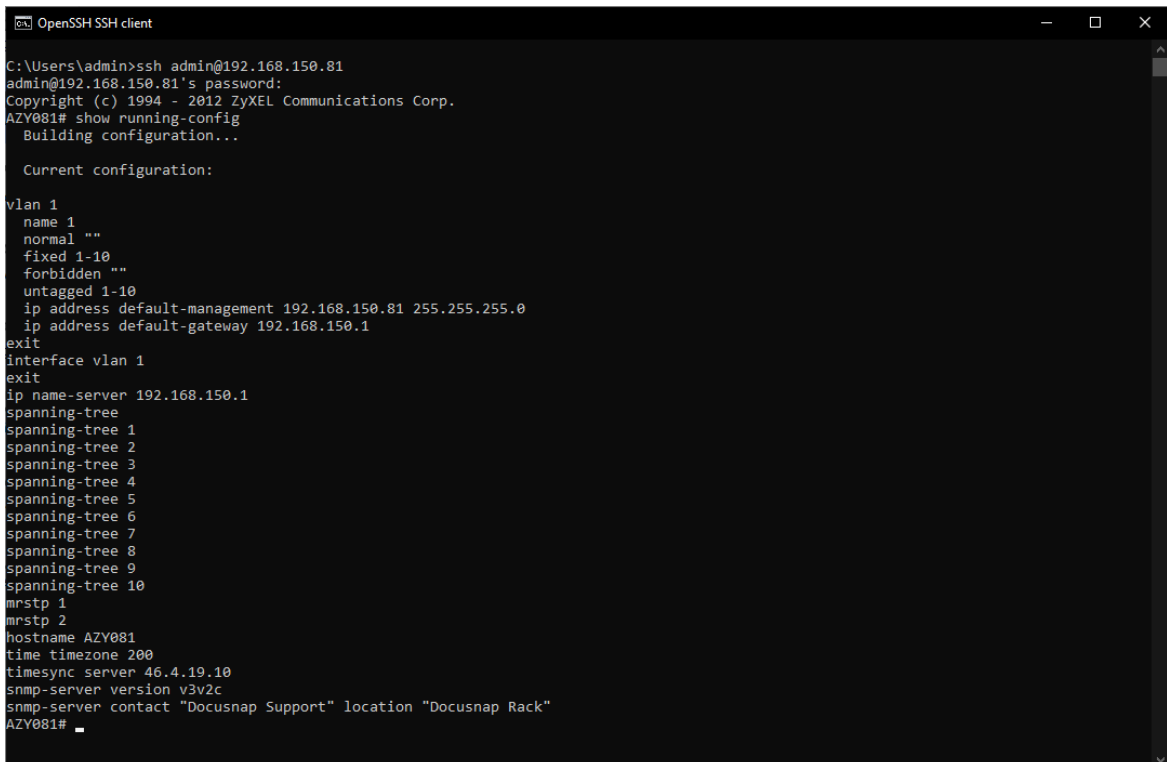
Figure 4 - WaitForString #

- **ResponseCapture**

This will start the console capture. All console output within ResponseCapture will be displayed later under Additional Information. Commands within ResponseCapture are added via the associated +.

- **Command**

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



```

C:\Users\admin>ssh admin@192.168.150.81
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 #**

Following this, we again wait for a # in the console output.



## 3.2 GENERAL COMMAND STATEMENTS

The following commands are available for the creation of a sequence

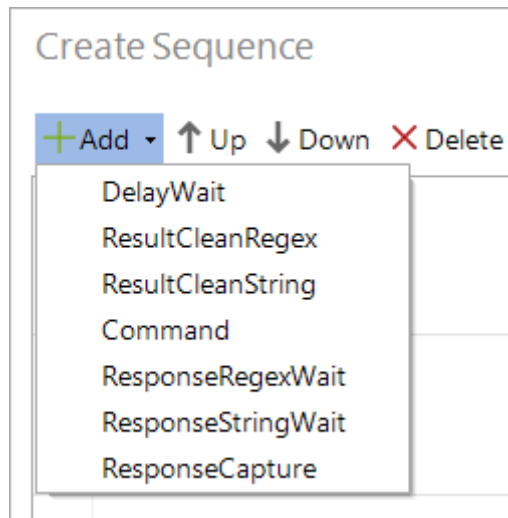


Figure 6 - Add Commands

- **DelayWait**  
Creates a delay to delay the execution of the next step if needed. This delay is specified in milliseconds.
- **ResponseRegexWait**  
Waits until the defined parameter is displayed in the console. The parameter for expecting a # is defined in 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 include a **ResponseRegexWait** or **ResponseStringWait** check between each command to increase stability.

- **Command**  
This executes a command on the target system such as *show running-config*. However, commands are not only executed that return information, but also implement a required input. If, for example, an "any key" must be passed in the console, this can be done with this. Therefore, the commands should be tested in the console before.
- **ResponseCapture**  
The command starts recording the console output. The previously created commands are not present in the Docusnap output. The subsequent commands must be created as a subsequence using the +. As a final command it is recommended to check for a console prompt.

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

- **ResultCleanRegex**

The command identifies a string using a regular expression and replaces it. For example, if you want to replace all characters after the string password with **\*\*\*\*\*** you must use the following expression

**RegexPattern:** Password\s.\*

**ReplaceWith:** **\*\*\*\*\***

- **ResultCleanString**

The command identifies a string and replaces it. The following parameters must be passed to replace the string "Community Secret#2021" with "Community **\*\*\*\*\***", for example.

**Find:** Community Secret#2021

**Replace:** Community **\*\*\*\*\***

### 3.3 TEST SEQUENCES

To facilitate the creation of a command sequence, it can be tested directly. Thus, a new inventory does not have to be performed every time the sequence is changed to test it.

**Check Sequence - \_example SSH\_d4f404ec-1a59-4bb5-b5d6-008d1b98a19f**

|          |                     |                |                    |
|----------|---------------------|----------------|--------------------|
| Company: | itelio GmbH         | Domain:        | docusnapsports.com |
| Switch:  | AZY081 ( GS2200-8 ) | Timeout (sec): | 5                  |
| User:    | admin               | Password:      | *****              |

| <input type="checkbox"/>            | Status | Discovery Service | Host Name | Plugin Version  |  |
|-------------------------------------|--------|-------------------|-----------|-----------------|--|
| <input checked="" type="checkbox"/> |        | Client-VPC-SDA    | VPC-SDA   | 11.0.1783.21319 |  |
| <input type="checkbox"/>            |        | Server-VPC-SDA    | VPC-SDA   | 11.0.1783.21319 |  |

Row Count: 2 of 2

**Check Sequence**

**Result:**

```

03.12.2021 16:26:37 | Type: Telnet | Step: 1 | Wait: String: #
03.12.2021 16:26:37 | Type: Telnet | Step: 1 | Resp:
03.12.2021 16:26:37 | Type: ResponseCapture | Step: 2 | ResponseCapture start, at position 0
03.12.2021 16:26:37 | Type: Telnet | Step: 3 | Send Command: show running-config
03.12.2021 16:26:38 | Type: Telnet | Step: 4 | Wait: String: #
03.12.2021 16:26:38 | Type: Telnet | Step: 4 | Resp: +7+7+7+7+7+7+7+7+7+7+7+7+7 Building configuration...

Current configuration:

vlan 1
 name 1
 normal ""
 fixed 1-10
 forbidden ""
    
```

✓ Verification successful!

OK

Figure 7 - Check Sequence

## LIST OF FIGURES

|   |    |
|---|----|
| FIGURE 1 - SNMP DIALOG - ADDITIONAL TOOLS                   | 5  |
| FIGURE 2 - HIERARCHICAL STRUCTURE - ADDITIONAL TOOLS OUTPUT | 6  |
| FIGURE 3 - CREATE SEQUENCE                                  | 7  |
| FIGURE 4 - WAITFORSTRING #                                  | 8  |
| FIGURE 5 - COMMAND - SHOW RUNNING-CONFIG                    | 8  |
| FIGURE 6 - ADD COMMANDS                                     | 9  |
| FIGURE 7 - CHECK SEQUENCE                                   | 11 |

## VERSION HISTORY

---

| Date       | Description   |
|------------|---|
| 12/05/2023 | Version 2.0 – Revision of the HowTo for Docusnap 13 |

---

