Docusnap X - Docusnap Script Linux

Script-based Inventory for Linux

TITLE	Docusnap X - Docusnap Script Linux
AUTHOR	Docusnap Consulting
DATE	2/28/2020
VERSION	1.2 | valid from 07.02.2020

CONTENTS

1. Introduction

For the remote inventory of a Linux system with Docusnap, SSH must be enabled on the Linux system and access as root is possible. In some Linux distributions (such as Ubuntu), remote access for the root user is blocked by default. In this case, Docusnap will not have the required permissions to scan this Linux system. If it is not possible to enable remote access via a root login, you can use the DSLinux script for this purpose.

IGEL ThinClients based on Linux can be inventoried with this script as well.

Corresponding scripts are also available for Windows or Exchange Server systems. They are covered in separate support documents.

Finally, we will demonstrate how to import the resulting information into the Docusnap database.

2. Basics

2.1 Filing location

The Docusnap installer stores the DSLinux script in the Bin sub-directory of the Docusnap installation directory. The script is available in a 32-bit and a 64-bit version.

2.2 Functionality

When you run the DSLinux script, it creates XML output containing all inventory data of the local system. Using a redirection operator, you can write this output to an XML file. Then, this XML file can be imported into Docusnap.

For further details, please refer to the IMPORTING THE INVENTORY DATA section.

2.3 Automation

It is also possible to run the DSLinux script in an automated way. By specifying corresponding parameters, you can store the XML files created by the script in a central network share and then import it automatically at the time specified in the schedule.

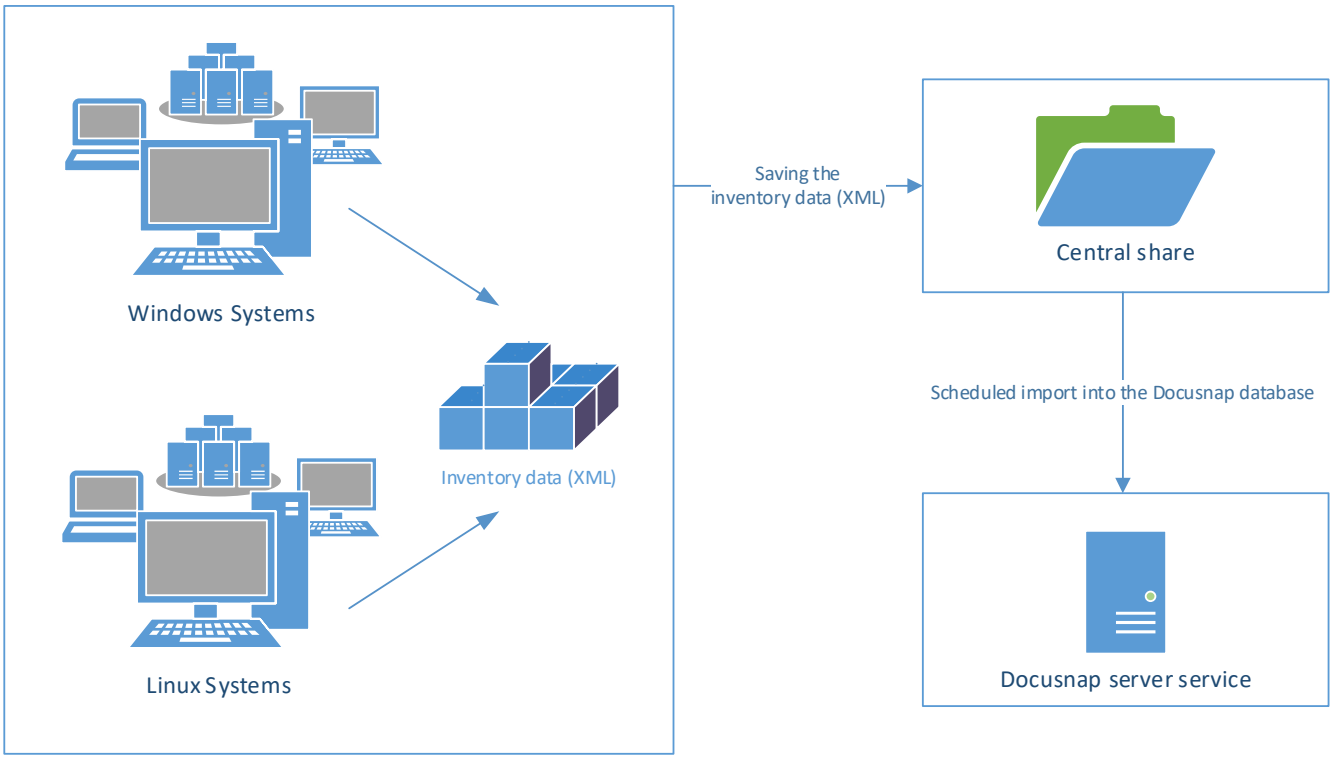

Fig. 1 – Workflow of the automatic import

2.4 Required permissions

In order to properly run the DSLinux script, root permissions are required on the corresponding Linux system.

3. Running the script

3.1 Prerequisites

Before running the DSLinux script, check whether the Linux system to be scanned is a 32-bit or a 64-bit system. Select the matching script file (DSLinux_x32 or DSLinux_x64) for that system.

In most Linux systems, you can use the *uname* command to determine the system architecture.

This document describes how to access the Linux system using SSH.

For remote access to the Linux system to be scanned from a Windows system, the following free tools are available:

- WinSCP
- PuTTY

What is more, the script can be run directly from the console of the Linux system. The commands used in the shell are the same as for remote access.

In both cases, the DSLinux file must be stored either directly on the Linux system or in a directory that can be reached from the Linux system.

3.2 Copying the script to the target system

This section describes how to copy the DSLinux script to the target system using WinSCP.

This step is only required if you cannot access the DSLinux script on the Linux system via a network share. If it is not possible to access the DSLinux script, you can copy the required version to the network share and then continue with the RUNNING THE SCRIPT MANUALLY section. The paths shown in the example must be adapted accordingly.

Once you start WinSCP, the Login dialog opens where you can connect to the target system. To do so, enter the host name, the user name and the associated password.

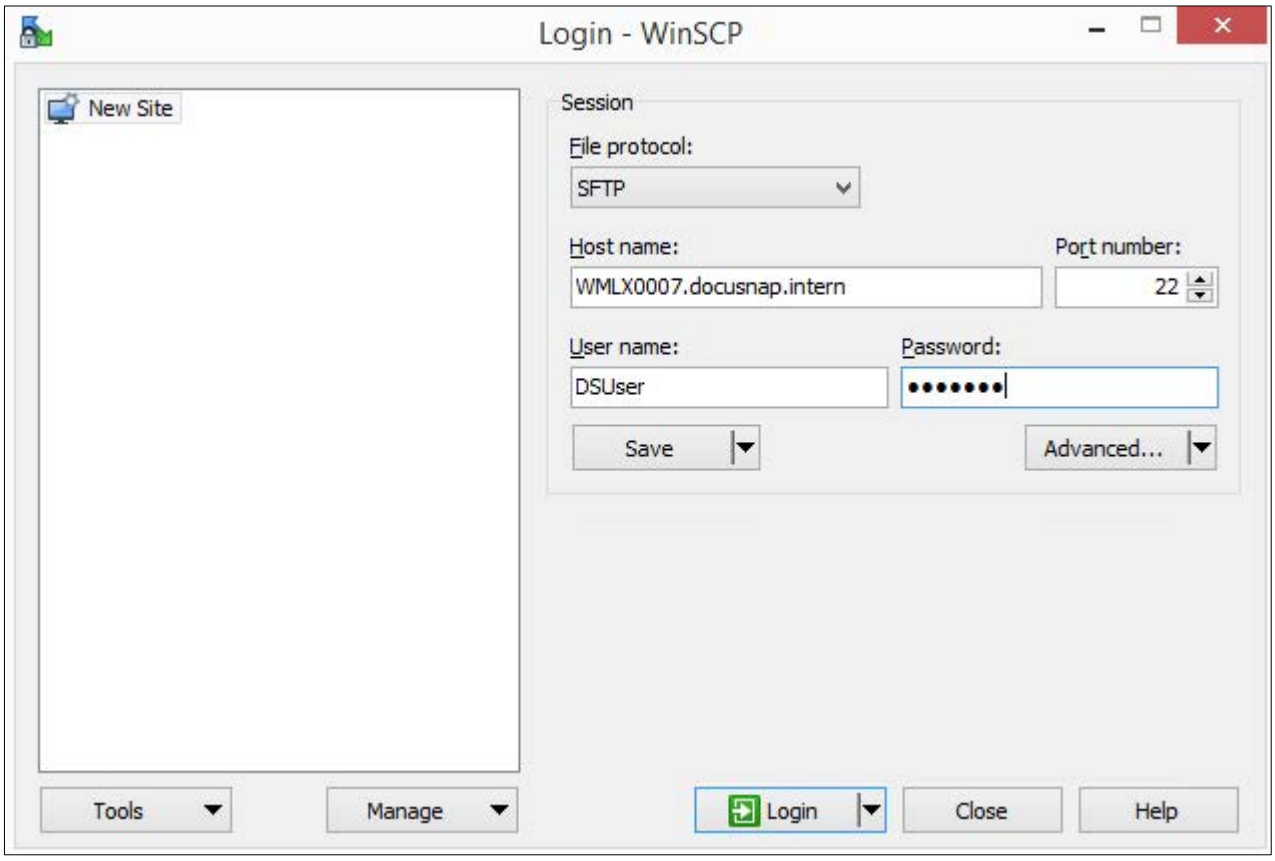

Fig. 2 – WinSCP login

After successful login to the Linux system, the left panel in WinSCP shows the Windows file system, the right panel the file system of the Linux system.

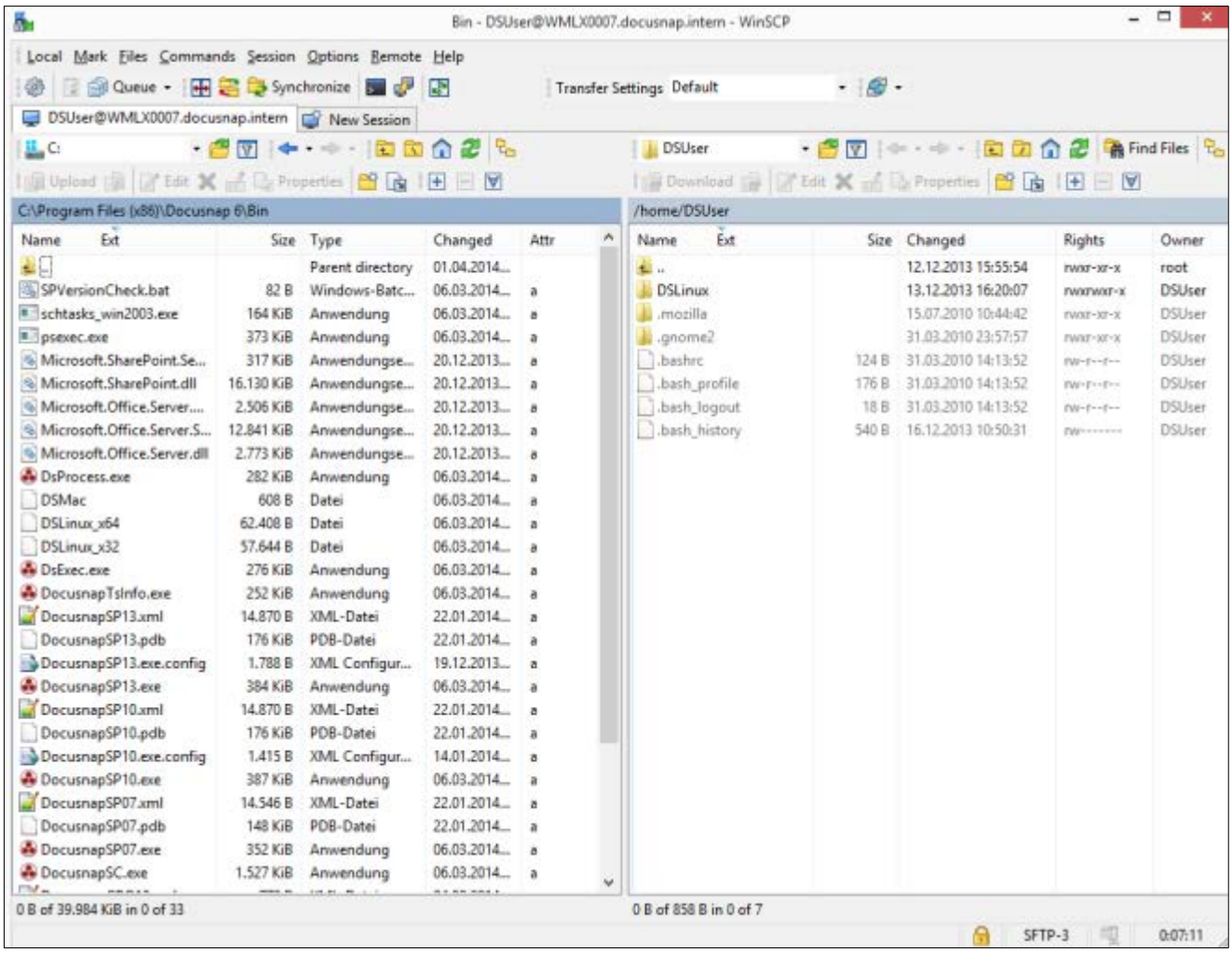

Fig. 3 – Authentication

In the file system of the Windows computer, browse to the "Bin" directory below the Docusnap program directory. For the Linux script, the following two versions are available:

- DSLinux_x32
- DSLinux_x64

Depending on the Linux architecture used on the computer (32-bit or 64-bit), now copy the matching file to any directory on the Linux target system.

In our example the DSLinux script will be copied to the home directory of DSUser.

3.3 Running the script manually

3.3.1 Remote login

After you have copied the DSLinux file successfully to the Linux system, you need to establish an SSH connection. For this purpose, use the PuTTY tool.

In WinSCP, a button is available that opens the PuTTY tool and establishes a connection to the target system.

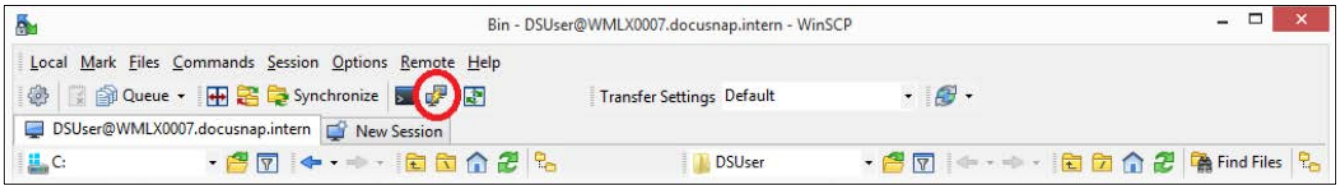

Fig. 4 – PuTTY button

Then, you need to authenticate on the target system.

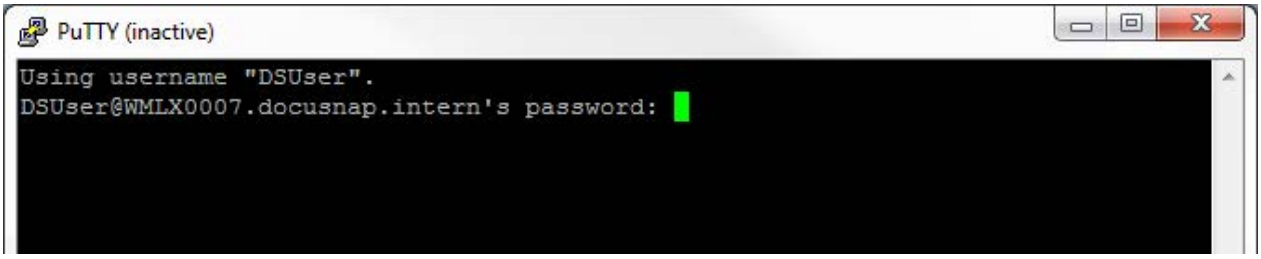

Fig. 5 – Authentication

After successful authentication, the connection via SSH to the target system is established. There, you can execute commands or browse the file system.

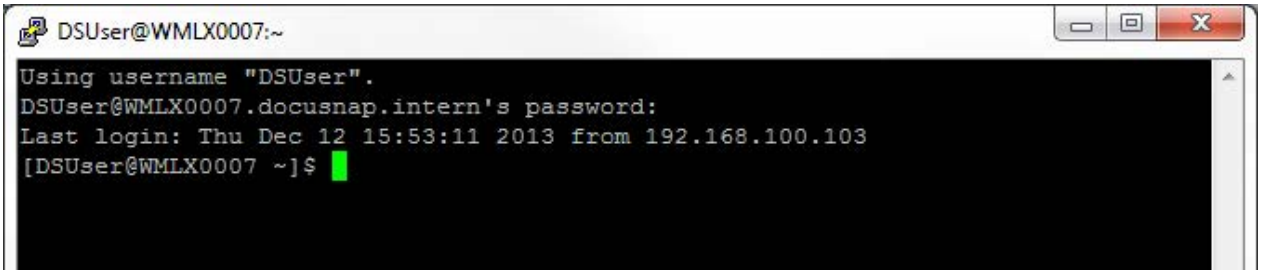

Fig. 6 – Connection successfully established

3.3.2 Making the script executable

Now, navigate to the folder in the Linux file system where the DSLinux script file is stored.

Enter the following command to make it executable:

```
[root@WMLX0007 DSUser]# chmod +x DSLinux_x32
```

or

```
[root@WMLX0007 DSUser]# chmod +x DSLinux_x64
```

3.3.3 RUNNING THE SCRIPT

CAUTION: For this and all further steps, root permissions are required. So first enter the *su* command to obtain root permissions.

```
[root@WMLX0007 DSUser]# ./DSLinux_x32 > outputfile.xml
```

or

```
[root@WMLX0007 DSUser]# ./DSLinux_x64 > outputfile.xml
```

By entering the redirection operator (>), you create the output file (here called "ausgabedatei.xml). You can use any desired name for it. Entering an output file name is mandatory. If you omit this name, the retrieved information will not be written to a file, but only displayed on the console.

It is possible to specify not only a file name, but also the directory where you want to store the output file. To store the output file in the */home/DSUser/DSLinux* directory, enter the following command:

```
[root@WMLX0007 DSUser]# ./DSLinux_x64 > /home/DSUser/DSLinux/outputfile.xml
```

Once the command has been executed successfully, the /home/DSUser/DSLinux directory contains an .XML file named ausgabedatei.xml.

```
[root@WMLX0007 DSUser]# ./DSLinux_x64 > /home/DSUser/DSLinux/outputfile.xml
./DSLinux_x64: line 1104: -n: command not found
[root@WMLX0007 DSUser]# cd DSLinux
[root@WMLX0007 DSUser]# ls
outputfile.xml
```

We recommend to redirect the output file directly to a network share that can be accessed both from Linux and Windows systems.

If the message "line 1104: -n command not found" is displayed while you run the script, ignore it because has no effect on the output file.

3.4 Automatic execution

To automate the execution of DSLinux, we recommend that you create a *bash* script and have it started automatically upon login to the Linux system.

The *bash* script should look like this:

```
#!/bin/bash

HostName='hostname'
ResultFile="$HostName.xml"

if [ "$1" = 64 ]
then
      chmod +x DSLinux_x64
      ./DSLinux_x64 > $ResultFile
else
      chmod +x DSLinux_x32
      ./DSLinux_x32 > $ResultFile
fi
```

This bash script retrieves the host name and uses it as the file name for the output file. In addition, you need to pass the parameter 32 or 64 to the script, depending on the Linux operating system architecture. Before actually running DSLinux, ensure that the script is executable by entering the *chmod +x* command.

The bash script must also be made executable beforehand by using the *chmod +x* command.

```
[root@WMLX0007 DSUser]# chmod +x Scriptname
```

Enter the following to run the script:

```
[root@WMLX0007 DSUser]# ./Scriptname 32
```

or

```
[root@WMLX0007 DSUser]# ./Scriptname 64
```

After execution, you will find an .XML file whose file name is the same as the host name in the folder where the DSLinux file resides. As explained above, it is possible to store the output file in any folder you like. To do so, simply precede the $ResultFile variable with the desired directory name.

```
./DSLinux_x64 > /home/DSUser/DSLinux/$ResultFile
```

In addition, you can use the script for automatic rule-based execution via a *crontab*.

4. Importing the inventory data

Using the Docusnap script import feature, you can import the xml files generated by the script into Docusnap.

To open the Import wizard, select *Inventory* > *Import* > *Script Import* from the user interface.

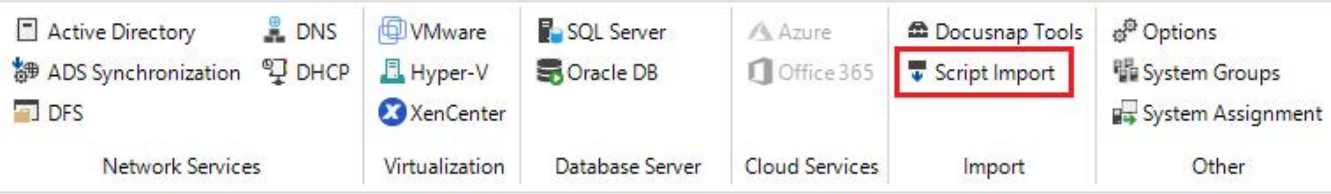

Fig. 7 – Opening the Import wizard

In the first step, select the desired company and domain for the import. Then, specify the path for the XML files. Specify the directory or UNC path of the central share from which to import the XML files.

Fig. 8 – Directory selection

In the next step, choose the button "Start searching for files" – now the wizard is searching for xml files generated by Docusnap - also for xml files which were generated by the Windows or Exchange script.

Fig. 9 – Directory selection

If you want to import Linux system inventory data automatically, schedule the import process. Make sure that the user account which was used to run the Docusnap Server service has read and write access to that directory.

Under Scheduling, you can create a job for this import, i.e. the XML files created by the script will be imported periodically at the times specified there.
However, this is only possible if Docusnap Server has been configured before. Please refer to the User Manual under https://www.docusnap.com/help/docusnap-x/user/docusnap-server.html.

LIST OF FIGURES

VERSION HISTORY

Date	Description
April 27, 2017	Version 1.0 – First Draft
October 24, 2018	Version 1.1 – Replacement of outdated screenshots, links updated
February 07, 2020	Version 1.2 – Correction of the bash script