Docusnap

Additional Tools

Configuring and evaluating

TITLE	Additional Tools
AUTHOR	Docusnap Consulting
DATE	1/20/2022
VERSION	1.1 | valid from 1/20/2022

CONTENTS

1. Introduction

Standard information about the workstations and servers is read out via the Windows inventory. With the help of additional tools, you can extend the Windows inventory with certain properties, so that not only WMI classes are queried, but also command line programs, such as systeminfo.exe, values from the registry or even PowerShell scripts. The results are written to the Docusnap database below the respective system.

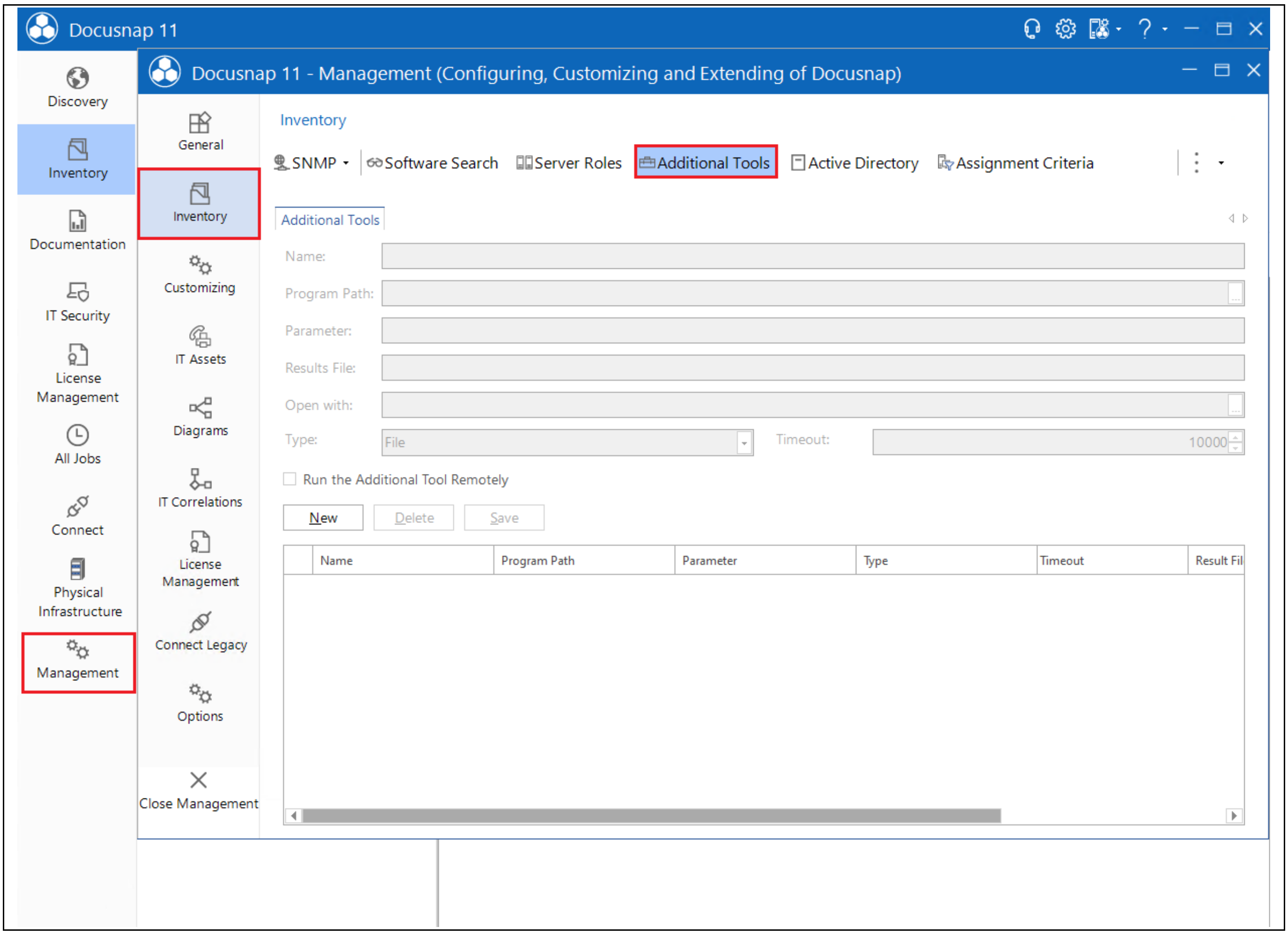
The following HowTo describes how you can configure the additional tools and then also evaluate them in a meaningful way.

- Chapter 3 describes the requirements for the use of the additional programs
- In Chapter 4 you will learn how to create add-on programs
- Chapter 5 describes the topic to you, in the form of application examples
- In chapter 6 you will be introduced to the evaluation options
- In the last part, chapter 7, the limitations are shown

2. Basics

In the Docusnap administration, you will find the administration of the additional tools in the Inventory area. Within this, you can create new additional tools or modify existing ones. The following fields are available as properties of the additional tools:

- Name
- Program Path
- Parameter
- Results File
- Open with
- Type
- Timeout
- Run the Additional Tool Remotely



Name:

Program Path:

Parameter:

Results File:

Open with:

Type: File

Timeout: 10000

☐ Run the Additional Tool Remotely

	Name	Program Path	Parameter	Type	Timeout	Result Fil

Figure 1 - Management of additional tools in Docusnap administration

3. Requirements

In order to run an Additional Tool program remotely on another system, access to the C$ share is required.

If this access is not given, the transfer of the file fails, because this is done via the administrative share.

4. Create Additional Tools

4.1 Define necessary values

In order for Docusnap to successfully output the results of the additional tools, the following values must be stored:

- Name

You can freely define the name of the program.

- Program Path

Next you have to store the path to the executable program. If information from the registry is needed or you execute a PowerShell query, the path to the PowerShell.exe in the file system is needed. For command line programs the name of the program is sufficient (e.g. Systeminfo.exe).

- Parameters

Depending on which method you use, this field is used to pass the entered parameters to the program. If you use PowerShell, enter the actual query here. With command line programs the respective parameters (e.g. /S for the host name, /U for the user name, etc.). If no parameters are required, at least one space must be entered.

Please note that the PowerShell parameter must be entered in brackets!

- Results file

Docusnap saves the results in BLOB format to the database, which needs a name. The name of the result file is again freely definable, but the format must be specified, e.g. whether it is a .TXT file or a .DOCX file, etc.

- Open with

With this field you can define which program is used to open the previously created result file (e.g. notepad.exe, WinWord.exe, etc.).

- Type

When selecting the type, you have two options: File and Stream.

If you choose File, a temporary file will be created on the hard disk during the inventory, where the results will be saved - this file will then be read into the database and saved.

With the Stream choice, the results from the standard output are immediately written to the database. However, the cmd.exe will be visible for a short time.

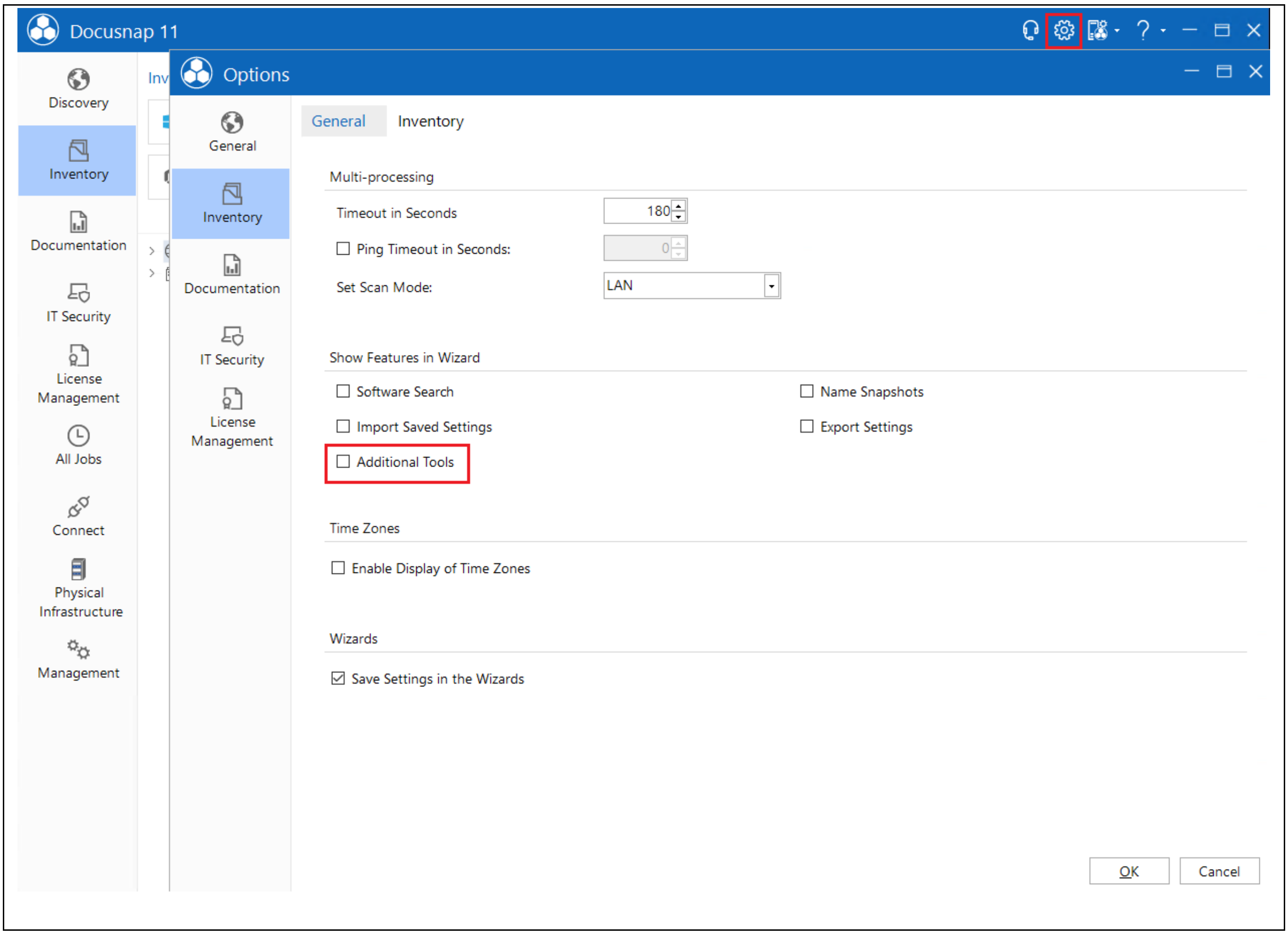
- Run the Additional Tool Remotely

Here it is defined whether the program is executed on the system on which the inventory process takes place or remotely at the respective system that is currently being inventoried.

4.2 Activate Additional Tools for inventory

After you have created the additional tools with the necessary information, you only need to activate them so that the respective additional tools are taken into account during the inventory.

To do this, switch to the Docusnap options via the gear wheel in the upper right corner. In the Inventory section, you can activate the Additional Tools option.



Multi-processing

Timeout in Seconds: 180

□ Ping Timeout in Seconds: 0

Set Scan Mode: LAN

Show Features in Wizard

□ Software Search

□ Name Snapshots

□ Import Saved Settings

□ Export Settings

□ Additional Tools

Time Zones

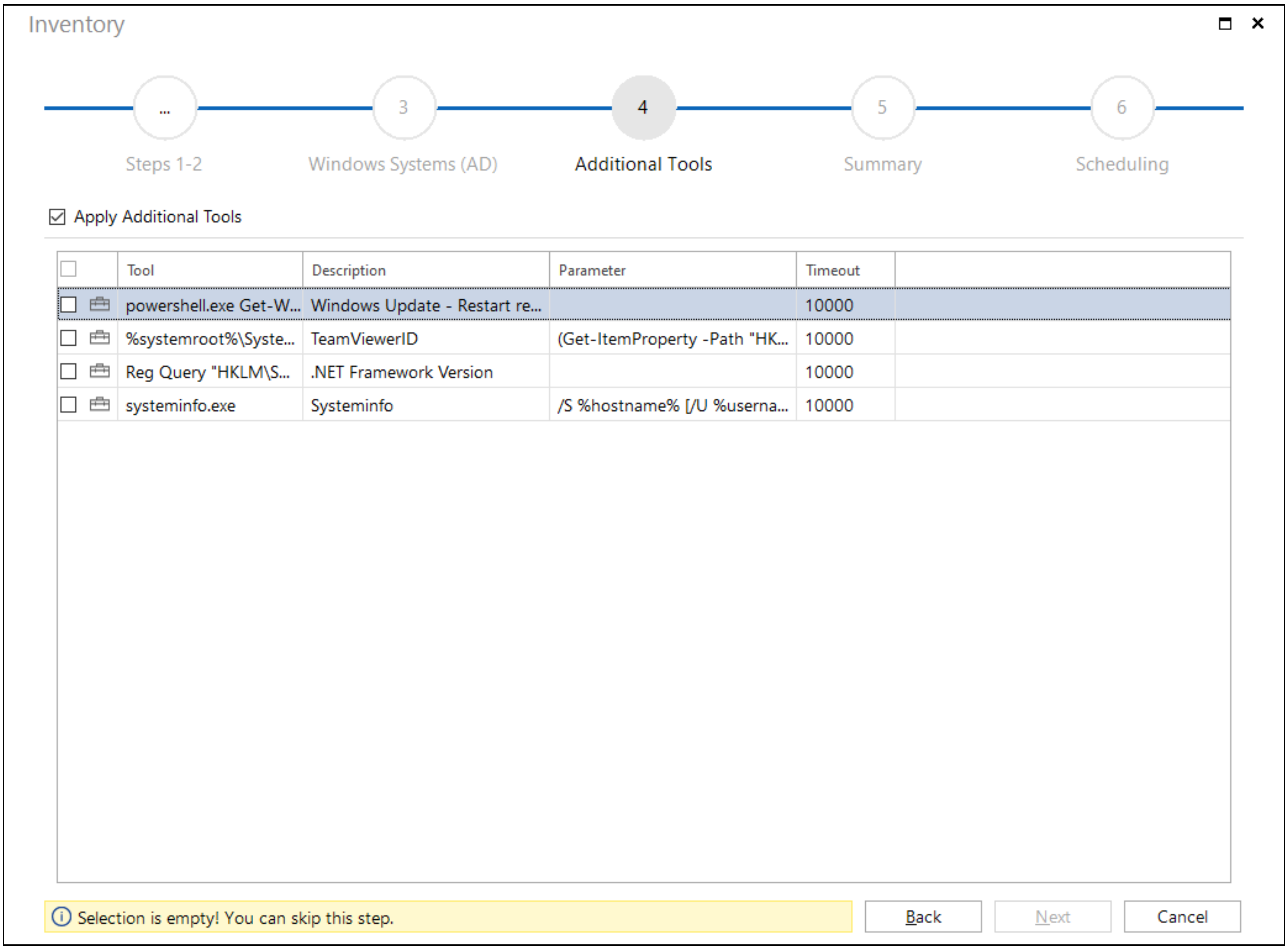
□ Enable Display of Time Zones

Wizards

☑ Save Settings in the Wizards

Figure 2 - Activating the Additional Tools via the options

In the Windows inventories, in step 4 - Additional Tools, the option is now offered to select one or more additional tools which are to be taken into account for the inventory.



☑ Apply Additional Tools

□	Tool	Description	Parameter	Timeout
□	powershell.exe Get-W...	Windows Update - Restart re...		10000
□	%systemroot%\Syste...	TeamViewerID	(Get-ItemProperty -Path "HK...	10000
□	Reg Query "HKLM\S...	.NET Framework Version		10000
□	systeminfo.exe	Systeminfo	/S %hostname% [/U %userna...	10000

Selection is empty! You can skip this step.

Figure 3 - Additional Tools in the scan assistent

Important note: If you already have scheduled inventories in use and would like to take the additional tools for these into account in the future, then the jobs must be deleted once and created again. It is not possible to add the option afterwards.

Additional note: If you use the Docusnap script variant for the inventory, you can also use the additional tools there via parameters. A detailed description can be found in our HowTo - Docusnap Script Windows.

5. Examples

The following section describes the additional tools in more detail using examples.

5.1 Display the stored TeamViewerID (PowerShell)

In the first example, we use a PowerShell script to query the stored TeamViewer ID of a system - if one is stored there.

Additional Tool for TeamViewer ID	
Name:	TeamViewerID
Program Path	%systemroot%\System32\WindowsPowerShell\v1.0\powershell.exe
Parameter:	(Get-ItemProperty -Path "HKLM:\SOFTWARE\Wow6432Node\TeamViewer").ClientID
Results File:	TeamViewerID.txt
Open with:	Notepad.exe
Type:	Stream
Timeout:	10000
Run the Additional Tool Remotely:	Yes

Table 1 - Display the stored TeamViewer ID

Note: If you have installed the 64 bit version of TeamViewer, the following Registry path is relevant in the parameter:

(Get-ItemProperty -Path "HKLM:\SOFTWARE\TeamViewer").ClientID

Please note that the PowerShell parameter must be entered in brackets!

5.2 Get the .NET Framework version (Registry Entry)

As a second example, we query the registry for a specific entry that tells us the installed .NET Framework version of the systems.

Additional Tool for .NET Framework Version	
Name:	.NET Framework Version
Program Path	Reg Query "HKLM\SOFTWARE\Microsoft\NET Framework Setup\NDP\v4\Full" /v Version
Parameter:	
Results File:	NETFramework.txt
Open with:	Notepad.exe
Type:	Stream
Timeout:	10000
Run the Additional Tool Remotely:	Yes

Table 2 - Installed .NET Framework Version

5.3 Systeminfo.exe (command line program)

In the last example we show you the call of the command line program Systeminfo.exe.

Additional Tool for Systeminfo.exe	
Name:	Systeminfo
Program Path	Systeminfo.exe
Parameter:	/S %hostname% [/U %username% /P %password%] > %targetfile%
Results File:	Systeminfo.txt
Open with:	Notepad.exe
Type:	File
Timeout:	10000
Run the Additional Tool Remotely:	Yes

Table 3 - Output of Systeminfo.exe

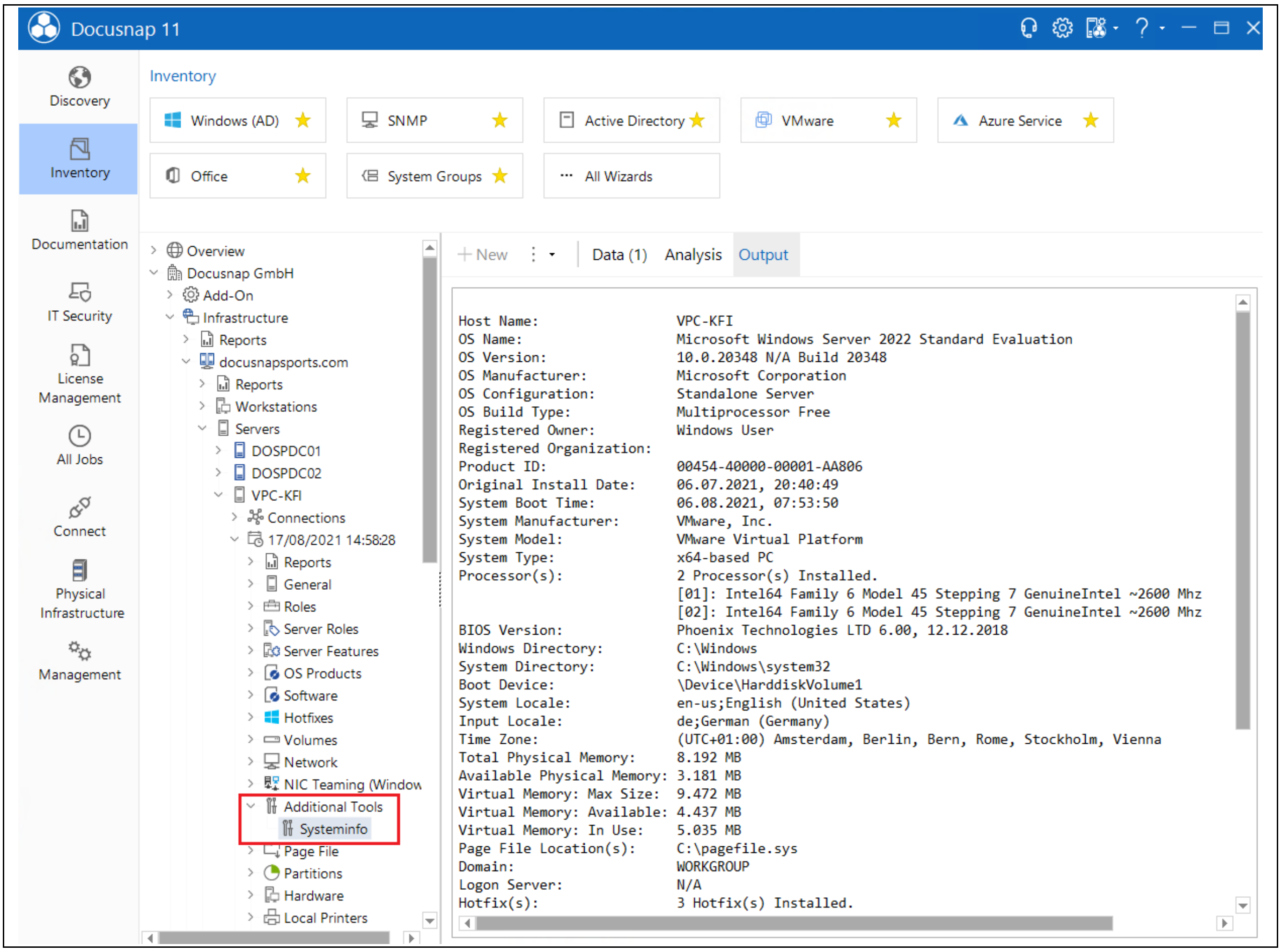
Docusnap provides the following environment variables:

- %Hostname%, %Computername%
- %Ipaddress%, %Ipadresse%
- %Domainname%, %Domain%, %Domäne%
- %Username%, %Benutzer%
- %Password%, %Passwort%
- %Targetfile%, %Zieldatei%
- %Description%, %Beschreibung%
- %Tool%, %Toolname%
- %accountname%
- %accountid%
- %domainid%
- %hostid%
- %docuid%

6. Documentation options

6.1 Output in Docusnap-tree

The first evaluation option is to view the results via the inventory tree. You can find the add-on programs under the Company - Infrastructure - Domain - Workstations/Servers - System - Snapshot - Additional Tools.



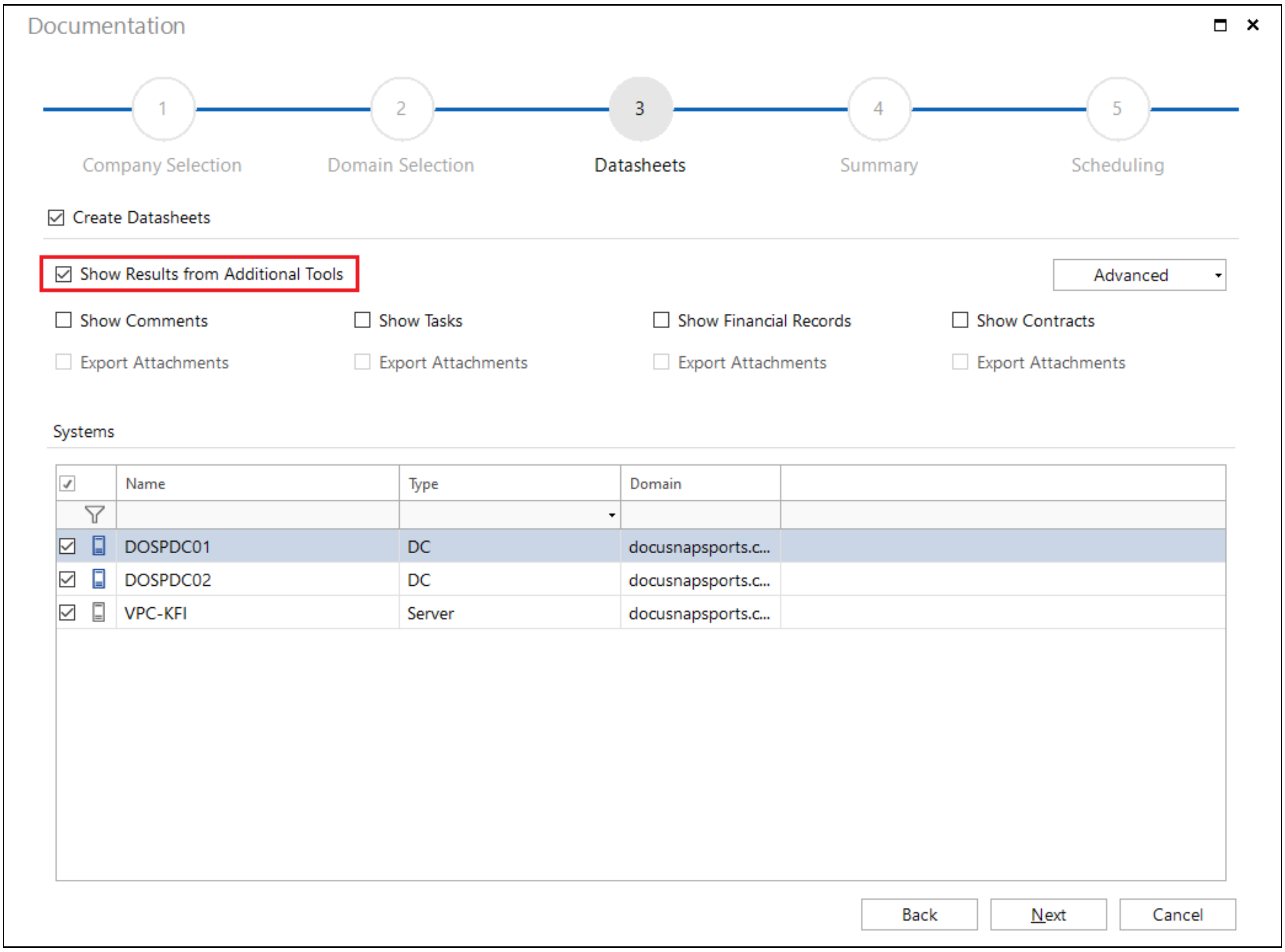
```
Host Name:                 VPC-KFI
OS Name:                   Microsoft Windows Server 2022 Standard Evaluation
OS Version:                10.0.20348 N/A Build 20348
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Server
OS Build Type:             Multiprocessor Free
Registered Owner:          Windows User
Registered Organization:
Product ID:                00454-40000-00001-AA806
Original Install Date:     06.07.2021, 20:40:49
System Boot Time:          06.08.2021, 07:53:50
System Manufacturer:       VMware, Inc.
System Model:              VMware Virtual Platform
System Type:               x64-based PC
Processor(s):              2 Processor(s) Installed.
                           [01]: Intel64 Family 6 Model 45 Stepping 7 GenuineIntel ~2600 Mhz
                           [02]: Intel64 Family 6 Model 45 Stepping 7 GenuineIntel ~2600 Mhz
BIOS Version:              Phoenix Technologies LTD 6.00, 12.12.2018
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             en-us;English (United States)
Input Locale:              de;German (Germany)
Time Zone:                 (UTC+01:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna
Total Physical Memory:     8.192 MB
Available Physical Memory: 3.181 MB
Virtual Memory: Max Size:  9.472 MB
Virtual Memory: Available: 4.437 MB
Virtual Memory: In Use:    5.035 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    WORKGROUP
Logon Server:              N/A
Hotfix(s):                 3 Hotfix(s) Installed.
```

Figure 4 - Output from Additional Tools in inventory tree

6.2 Output in datasheets

The second evaluation option is possible in the form of data sheets. In the wizard for creating the data sheets, you have the option of setting the Output results of additional tools option. This causes the results to be output as a corresponding file in your documentation path.

You will find detailed instructions on the data sheets in the Data sheets section of our user manual.



Documentation

1 Company Selection — 2 Domain Selection — 3 Datasheets — 4 Summary — 5 Scheduling

☑ Create Datasheets

☑ Show Results from Additional Tools

Advanced

☐ Show Comments ☐ Show Tasks ☐ Show Financial Records ☐ Show Contracts

☐ Export Attachments ☐ Export Attachments ☐ Export Attachments ☐ Export Attachments

Systems

☑	Name	Type	Domain
☑	DOSPDC01	DC	docusnapsports.c...
☑	DOSPDC02	DC	docusnapsports.c...
☑	VPC-KFI	Server	docusnapsports.c...

Figure 5 - Output Additional Tools with data sheets

6.3 Output per View

As a final evaluation option, you can also use a more advanced variant in the form of an SQL query. For example, a new node can be created in the inventory tree, which displays the results of the additional tools and the most important system information.

```
select tHosts.HostID, tHosts.Hostname, SiteID, OS,
(select ToolResultVarChar
from tToolResult where DocuID = (select DocuID from tDocu where tDocu.HostID = tHosts.HostID and Archiv
= 0)
and Filename like '%Beispiel.exe%' --the name of the result file must be entered here
) AS BuildNumber
from tHosts, tDocu, tDocuWindows
where tHosts.HostID = tDocu.HostID
and tDocu.DocuID = tDocuWindows.DocuID
and HostTypeID = 1
and Archiv = 0
```

The SQL queries can also be customized to start from a certain character. For this you need the substring function in SQL. In this example, everything up to the 84th digit is truncated and then the following 100 characters are output.

```
SUBSTRING(convert(varchar(max), cast(ToolResult as varbinary(max))),84,100)
```

7. Limitations

When it comes to Additional Tools, there are also few limitations that you should be aware of.

- Calling a script from a network share is not possible. The program to be executed must exist either on the Docusnap system that performs the inventory or on the target system that is currently being queried. Storage in the form of a UNC path (\\hostname\share\example script.exe) is not possible.
- Command line commands that are executed exclusively in the context of the 64-bit shell cannot be stored in the additional tools. Only commands that can be executed in the 32-bit and 64-bit context will work.

LIST OF FIGURES

INDEX OF TABLES

VERSION HISTORY

Date	Description
9/28/2021	Document created
1/3/2022	Requirements supplemented

Docusnap