Docusnap - Defining Notifications

E-mail Reminders

TITLE	Docusnap - Defining Notifications
AUTHOR	Docusnap Consulting
DATE	29/07/2021
VERSION	2.1 | valid as of July 27, 2021

TABLE OF CONTENTS

1. Introduction

Docusnap provides the ability to send notifications about events automatically via email. This document describes how to create a message definition and subsequently a notification in Docusnap.

The message definition uses SQL statements to define the event for which a notification should be sent. The notification task executes the message definition and sends an email when the event occurs.

To use email messaging, the Docusnap server must be configured, and an email server must be available. More information about this can be found in the Docusnap user manual or in the HowTo Installation and Configuration.

It also describes how you can use message definitions for regular maintenance tasks. For example, you can use a maintenance task to periodically populate the virtualization host field of the inventoried systems.

2. Basics

The message definitions are SQL queries. The message definition defines the event for which you want to be notified by e-mail. E.g., that a license or maintenance contract has expired, that a certain software has been found during inventory, etc.

For a message definition (the actual SQL statement) to be executed, a notification task is required. This notification task executes the SQL statement on the database and in case of a result, an email is sent with this result in an Excel list.

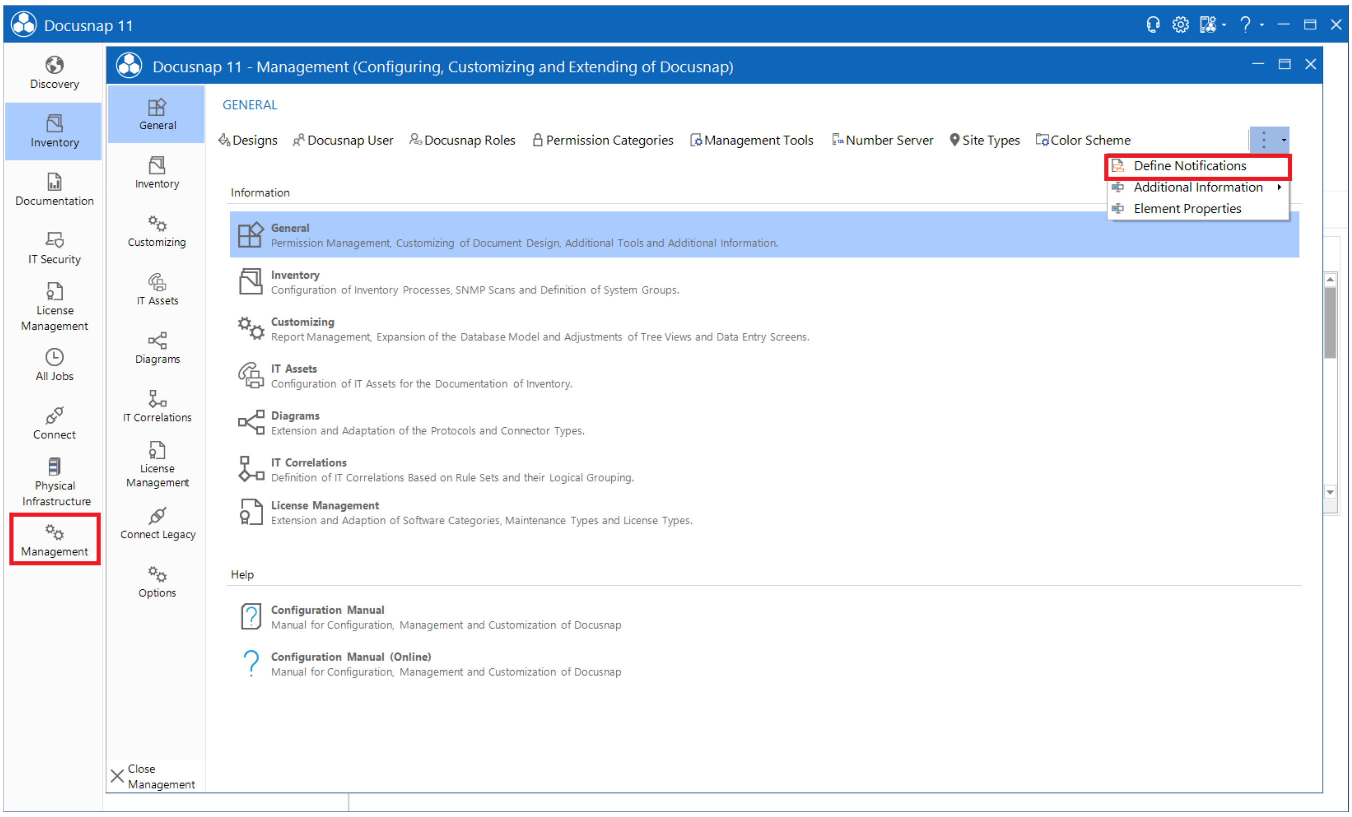
The important thing here is that Docusnap saves the last result. If a change is detected within the result during the next execution, a new email is sent. If there is no change, no new e-mail will be sent!

Furthermore, it should be noted that the Excel file receives the name of the scheduled notification task. Please use only characters that are allowed in a file name.

3. Notification definitions

3.1 Accessing the notification definition

The wizard for the notification definition can be opened from Navigation – Administration - General.



Information

General
Permission Management, Customizing of Document Design, Additional Tools and Additional Information.

Inventory
Configuration of Inventory Processes, SNMP Scans and Definition of System Groups.

Customizing
Report Management, Expansion of the Database Model and Adjustments of Tree Views and Data Entry Screens.

IT Assets
Configuration of IT Assets for the Documentation of Inventory.

Diagrams
Extension and Adaptation of the Protocols and Connector Types.

IT Correlations
Definition of IT Correlations Based on Rule Sets and their Logical Grouping.

License Management
Extension and Adaption of Software Categories, Maintenance Types and License Types.

Help

Configuration Manual
Manual for Configuration, Management and Customization of Docusnap

Configuration Manual (Online)
Manual for Configuration, Management and Customization of Docusnap

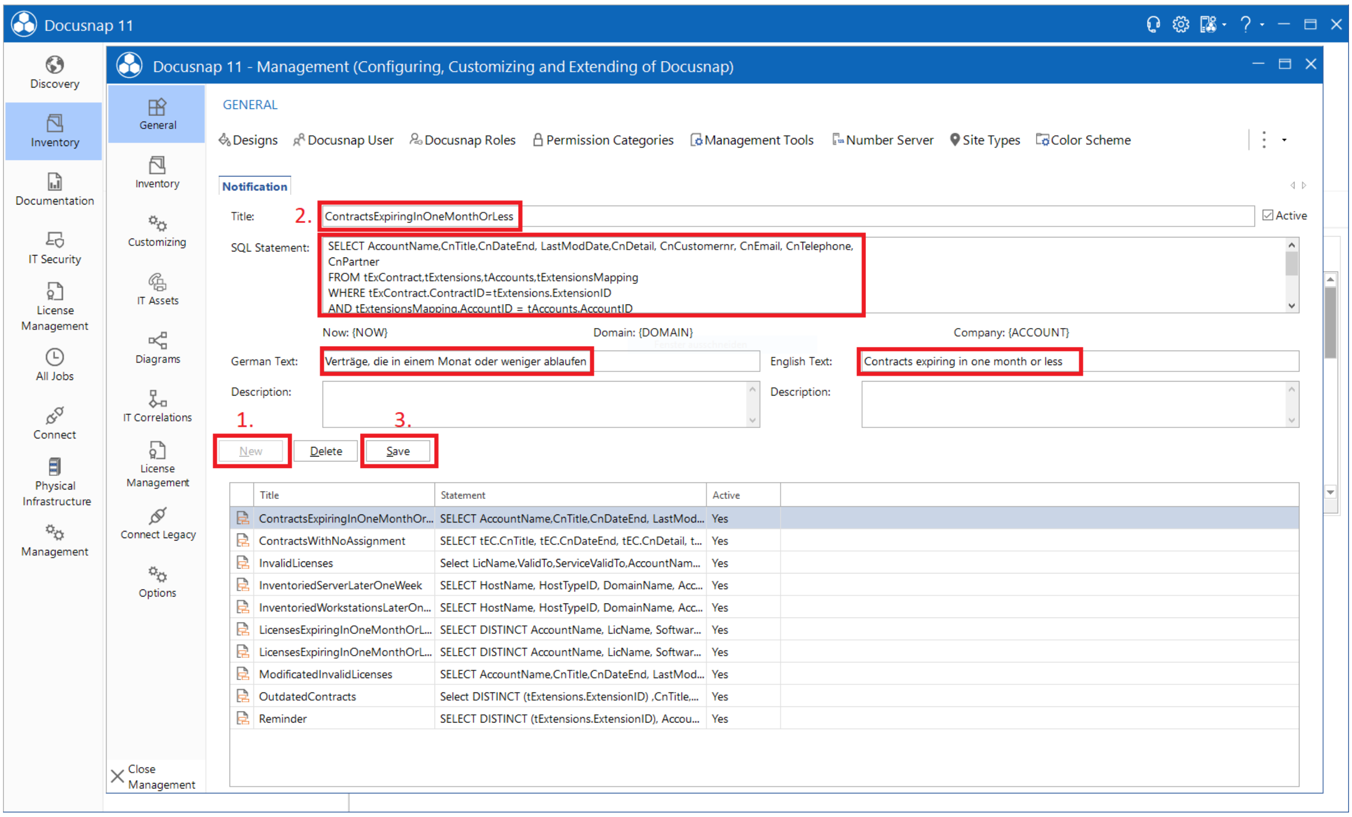
Fig. 1 - Accessing the notification definition

The following dialog can be used to create new notification definitions and to edit existing definitions.

Step 1: Select New.

Step 2: Enter a name, the SQL query and a text in English and German. So both fields need to be filled, just enter the English text in the field "German Text" also.

Step 3: Select Save.



Title: ContractsExpiringInOneMonthOrLess ☑ Active

SQL Statement:

```
SELECT AccountName,CnTitle,CnDateEnd, LastModDate,CnDetail, CnCustomernr, CnEmail, CnTelephone,
CnPartner
FROM tExContract,tExtensions,tAccounts,tExtensionsMapping
WHERE tExContract.ContractID=tExtensions.ExtensionID
AND tExtensionsMapping.AccountID = tAccounts.AccountID
```

Now: {NOW} Domain: {DOMAIN} Company: {ACCOUNT}

German Text: Verträge, die in einem Monat oder weniger ablaufen

English Text: Contracts expiring in one month or less

Description:

Description:

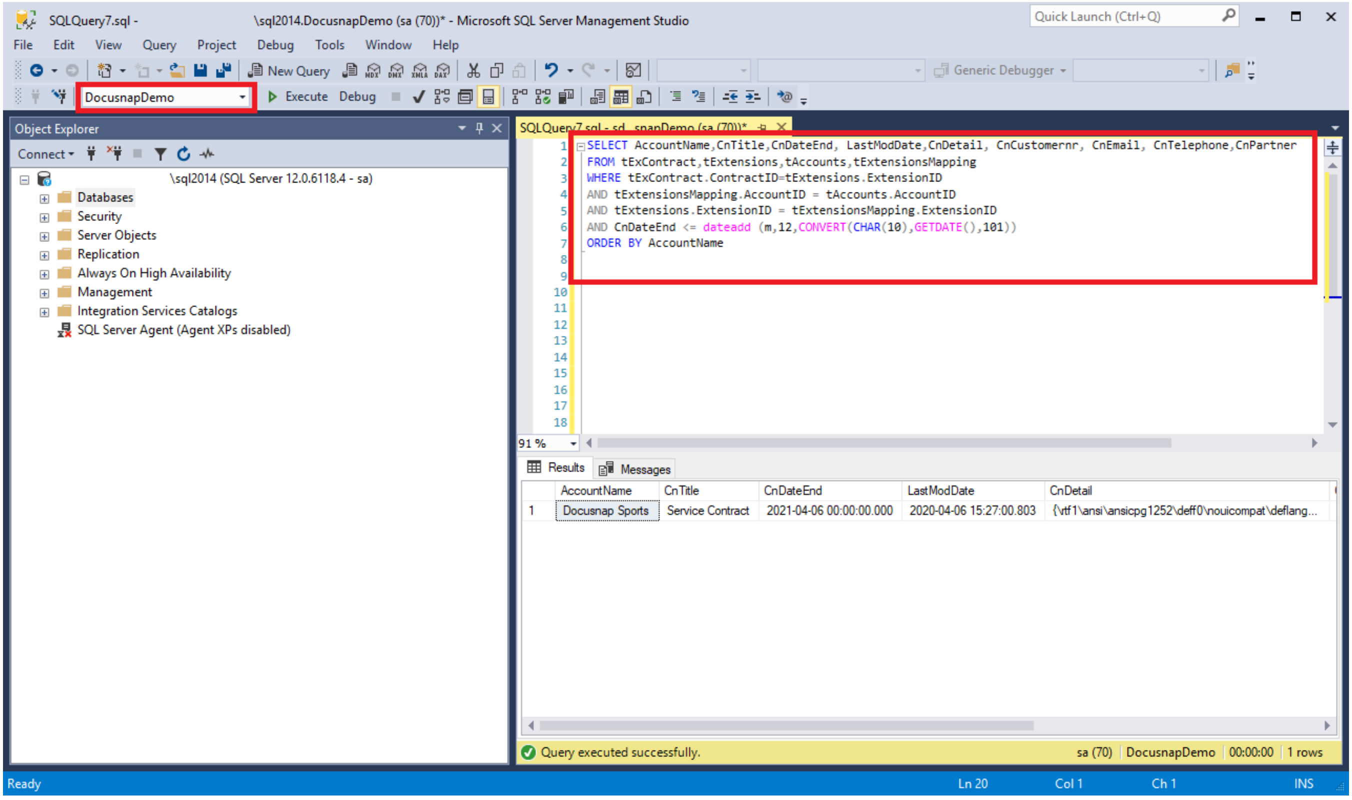
Title	Statement	Active
ContractsExpiringInOneMonthOr...	SELECT AccountName,CnTitle,CnDateEnd, LastMod...	Yes
ContractsWithNoAssignment	SELECT tEC.CnTitle, tEC.CnDateEnd, tEC.CnDetail, t...	Yes
InvalidLicenses	Select LicName,ValidTo,ServiceValidTo,AccountNam...	Yes
InventoriedServerLaterOneWeek	SELECT HostName, HostTypeID, DomainName, Acc...	Yes
InventoriedWorkstationsLaterOn...	SELECT HostName, HostTypeID, DomainName, Acc...	Yes
LicensesExpiringInOneMonthOrL...	SELECT DISTINCT AccountName, LicName, Softwar...	Yes
LicensesExpiringInOneMonthOrL...	SELECT DISTINCT AccountName, LicName, Softwar...	Yes
ModificatedInvalidLicenses	SELECT AccountName,CnTitle,CnDateEnd, LastMod...	Yes
OutdatedContracts	Select DISTINCT (tExtensions.ExtensionID) ,CnTitle,...	Yes
Reminder	SELECT DISTINCT (tExtensions.ExtensionID), Accou...	Yes

Fig. 2 - Creating a notification definition

3.2 Application samples

3.2.1 Preliminary check of notification definitions

Since the notification definitions are SQL queries, they can be checked preliminarily for correctness in SQL Server Management Studio. Here, you also get a preview of the result.



SQLQuery7.sql - \sql2014.DocusnapDemo (sa (70))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Debug Tools Window Help
New Query
Generic Debugger
DocusnapDemo
Execute Debug
Object Explorer
Connect
\sql2014 (SQL Server 12.0.6118.4 - sa)
Databases
Security
Server Objects
Replication
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs disabled)
91 %
Results
Messages
Query executed successfully.
sa (70)
DocusnapDemo
00:00:00
1 rows
Ready
Ln 20
Col 1
Ch 1
INS

```sql
SELECT AccountName,CnTitle,CnDateEnd, LastModDate,CnDetail, CnCustomernr, CnEmail, CnTelephone,CnPartner
FROM tExContract,tExtensions,tAccounts,tExtensionsMapping
WHERE tExContract.ContractID=tExtensions.ExtensionID
AND tExtensionsMapping.AccountID = tAccounts.AccountID
AND tExtensions.ExtensionID = tExtensionsMapping.ExtensionID
AND CnDateEnd <= dateadd (m,12,CONVERT(CHAR(10),GETDATE(),101))
ORDER BY AccountName
```

	AccountName	CnTitle	CnDateEnd	LastModDate	CnDetail
1	Docusnap Sports	Service Contract	2021-04-06 00:00:00.000	2020-04-06 15:27:00.803	{\rtf1\ansi\ansicpg1252\deff0\nouicompat\deflang...

Fig. 3 - Accessing the SQL queries in SQL Server Management Studio

Important: Make sure to run the SQL query with the correct database.

However, the placeholders used in Docusnap must be replaced for a query in SQL Server Management Studio:

Meaning	Placeholder in Docusnap	Replacement in SQL Server Mgmt Studio	Remark
Today	{NOW}	'04/20/15' CURRENT_TIMESTAMP	Make sure to indicate the date in the correct format as used in the database. Alternatively, you can use the SQL variable CURRENT_TIMESTAMP.
Domain	{DOMAIN}	DomainID	The Docusnap domain IDs can be displayed via the following SQL query: `SELECT TOP 100 [DomainID],[DomainName] FROM [tDomains]`
Company	{ACCOUNT}	AccountID	The Docusnap account IDs can be displayed via the following SQL query: `SELECT TOP 100 [AccountID],[AccountName] FROM [tAccounts]`

3.2.2 Conctracts

The following query lists all expired contracts of a company:

```
SELECT  DISTINCT  AccountName,  CnTitle,  CnDateEnd,  LastModDate,CnDetail,  CnCustomernr,  CnEmail,
CnTelephone, CnPartner

FROM tExContract
      LEFT JOIN tExtensions on tExContract.ContractID = tExtensions.ExtensionID
      LEFT JOIN tExtensionsMapping on tExtensions.ExtensionID = tExtensionsMapping.ExtensionID
      LEFT JOIN tAccounts on tExtensionsMapping.AccountID = tAccounts.AccountID

WHERE CnDateEnd <= {now}
      AND tAccounts.AccountID = {ACCOUNT}         --Filtering on selected company
```

Query contracts from all companies that expire in one month or less:

```
SELECT  AccountName,  CnTitle,  CnDateEnd,  LastModDate,CnDetail,  CnCustomernr,  CnEmail,  CnTelephone,
CnPartner

FROM tExContract
      LEFT JOIN tExtensions on tExContract.ContractID = tExtensions.ExtensionID
      LEFT JOIN tExtensionsMapping on tExtensions.ExtensionID = tExtensionsMapping.ExtensionID
      LEFT JOIN tAccounts on tExtensionsMapping.AccountID = tAccounts.AccountID

WHERE CnDateEnd <= DATEADD (MM, 1, SYSDATETIME())  -- MM, 1, ... -> 1 month in advance
```

If you need more lead time, you can set it to three months in advance, for example, as follows:

```
SELECT  AccountName,  CnTitle,  CnDateEnd,  LastModDate,CnDetail,  CnCustomernr,  CnEmail,  CnTelephone,
CnPartner

FROM tExContract
      LEFT JOIN tExtensions on tExContract.ContractID = tExtensions.ExtensionID
      LEFT JOIN tExtensionsMapping on tExtensions.ExtensionID = tExtensionsMapping.ExtensionID
      LEFT JOIN tAccounts on tExtensionsMapping.AccountID = tAccounts.AccountID

WHERE CnDateEnd <= DATEADD (MM, 3, SYSDATETIME())  --MM, 3, ... -> 3 month in advance
```

3.2.3 Licenses

The SQL query lists all licenses of a company that expire in a month or less:

```
SELECT SoftwareName, LicName, ValidTo, ServiceValidTo, AccountName

FROM tSoftwareLicenses
      LEFT JOIN tSoftwareProducts on tSoftwareLicenses.SoftwareID = tSoftwareProducts.SoftwareID
      LEFT JOIN tAccounts on tSoftwareProducts.AccountID = tAccounts.AccountID

WHERE tSoftwareLicenses.ValidInfinite = 0
      AND ValidTo <= DATEADD (MM, 1, SYSDATETIME())      --MM, 1 ,... -> 1 month in advance
      AND tAccounts.AccountID = {ACCOUNT}                --Filtering on the selected company

ORDER BY LicName
```

Query licenses from all companies that expire in a month or less:

```
SELECT  DISTINCT  AccountName,  LicName,  SoftwareName,  SoftwarePublisher,  SoftwareVersion,  ValidTo,
ServiceValidTo, Customernr, Telephone, Email

FROM tSoftwareLicenses
      LEFT JOIN tSoftwareProducts on tSoftwareLicenses.SoftwareID = tSoftwareProducts.SoftwareID
      LEFT JOIN tAccounts on tSoftwareProducts.AccountID = tAccounts.AccountID

WHERE tSoftwareLicenses.ValidInfinite = 0
      AND ValidTo <= DATEADD (MM, 1, SYSDATETIME())      --MM, 1 ,... -> 1 month in advance

ORDER BY AccountName, LicName
```

Query the licenses of all companies that expire within the next 3 months:

```
SELECT  DISTINCT  AccountName,  LicName,  SoftwareName,  SoftwarePublisher,  SoftwareVersion,  ValidTo,
ServiceValidTo, Customernr, Telephone, Email

FROM tSoftwareLicenses
      LEFT JOIN tSoftwareProducts on tSoftwareLicenses.SoftwareID = tSoftwareProducts.SoftwareID
      LEFT JOIN tAccounts on tSoftwareProducts.AccountID = tAccounts.AccountID

WHERE tSoftwareLicenses.ValidInfinite = 0
      AND ValidTo <= DATEADD (MM, 3, SYSDATETIME())      --MM, 3 ,... -> 3 month in advance

ORDER BY AccountName, LicName
```

3.2.4 Maintenance contracts (licenses)

Within a license definition, the validity of the maintenance contract can be stored in addition to the license validity. With the following query you can display all maintenance contracts of a company that expire in one month or less:

```
SELECT SoftwareName, LicName, ServiceValidTo, AccountName

FROM tSoftwareLicenses
      LEFT JOIN tSoftwareProducts on tSoftwareLicenses.SoftwareID = tSoftwareProducts.SoftwareID
      LEFT JOIN tAccounts on tSoftwareProducts.AccountID = tAccounts.AccountID

WHERE tSoftwareLicenses.ValidInfinite = 0
      AND ServiceValidTo <= DATEADD (MM, 1, SYSDATETIME())       --MM, 1 ,... -> 1 month in advance
      AND tAccounts.AccountID = {ACCOUNT}                          --Filtering on the selected company

ORDER BY LicName
```

The following query lists the maintenance contracts of all companies that expire in one month or less.

```
SELECT DISTINCT AccountName, LicName, SoftwareName, SoftwarePublisher, SoftwareVersion, ServiceValidTo,
Customernr, Telephone, Email

FROM tSoftwareLicenses
      LEFT JOIN tSoftwareProducts on tSoftwareLicenses.SoftwareID = tSoftwareProducts.SoftwareID
      LEFT JOIN tAccounts on tSoftwareProducts.AccountID = tAccounts.AccountID

WHERE tSoftwareLicenses.ValidInfinite = 0
      AND ServiceValidTo <= DATEADD (MM, 1, SYSDATETIME())       --MM, 1 ,... -> 1 month in advance

ORDER BY AccountName, LicName
```

4. Creating a notification

The wizard for creating notifications can be opened from All Tasks.

All assistants can be displayed in the multi-function bar. The notification wizard can be opened then. It can be found directly with the search function.

If the Notification Wizard is marked as a favorite, it can be opened directly from the Multi-Function Bar.

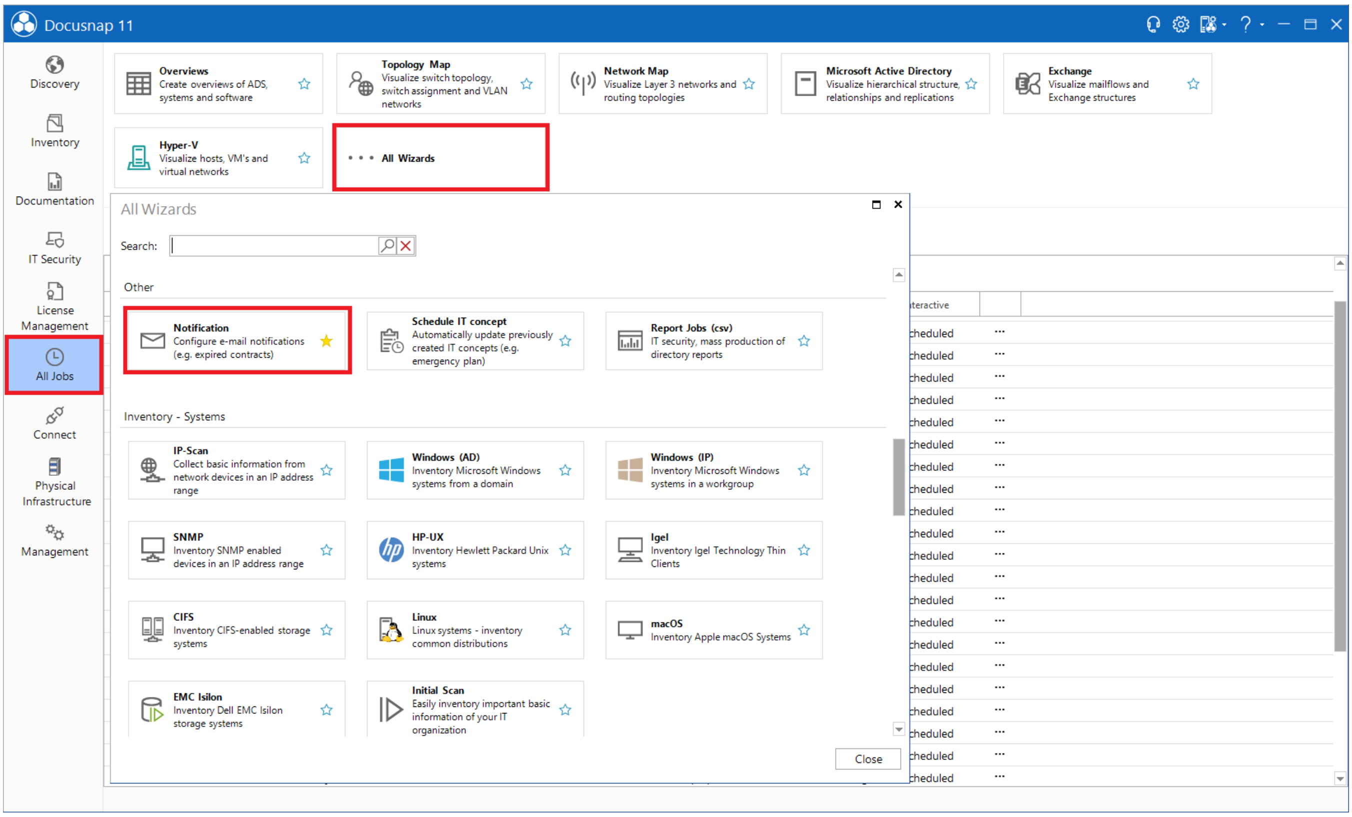

Docusnap 11
Discovery
Inventory
Documentation
IT Security
License Management
All Jobs
Connect
Physical Infrastructure
Management
Overviews
Create overviews of ADS, systems and software
Topology Map
Visualize switch topology, switch assignment and VLAN networks
Network Map
Visualize Layer 3 networks and routing topologies
Microsoft Active Directory
Visualize hierarchical structure, relationships and replications
Exchange
Visualize mailflows and Exchange structures
Hyper-V
Visualize hosts, VM's and virtual networks
• • • All Wizards
All Wizards
Search:
Other
Notification
Configure e-mail notifications (e.g. expired contracts)
Schedule IT concept
Automatically update previously created IT concepts (e.g. emergency plan)
Report Jobs (csv)
IT security, mass production of directory reports
Inventory - Systems
IP-Scan
Collect basic information from network devices in an IP address range
Windows (AD)
Inventory Microsoft Windows systems from a domain
Windows (IP)
Inventory Microsoft Windows systems in a workgroup
SNMP
Inventory SNMP enabled devices in an IP address range
HP-UX
Inventory Hewlett Packard Unix systems
Igel
Inventory Igel Technology Thin Clients
CIFS
Inventory CIFS-enabled storage systems
Linux
Linux systems - inventory common distributions
macOS
Inventory Apple macOS Systems
EMC Isilon
Inventory Dell EMC Isilon storage systems
Initial Scan
Easily inventory important basic information of your IT organization
Close
teractive
cheduled

Fig. 4 – Notification Wizard

The first step is to select a company. If the filtering in the message definition was

```
tAccounts.AccountID = {ACCOUNT}
```

was used in the message definition, the results are limited to this selected client.

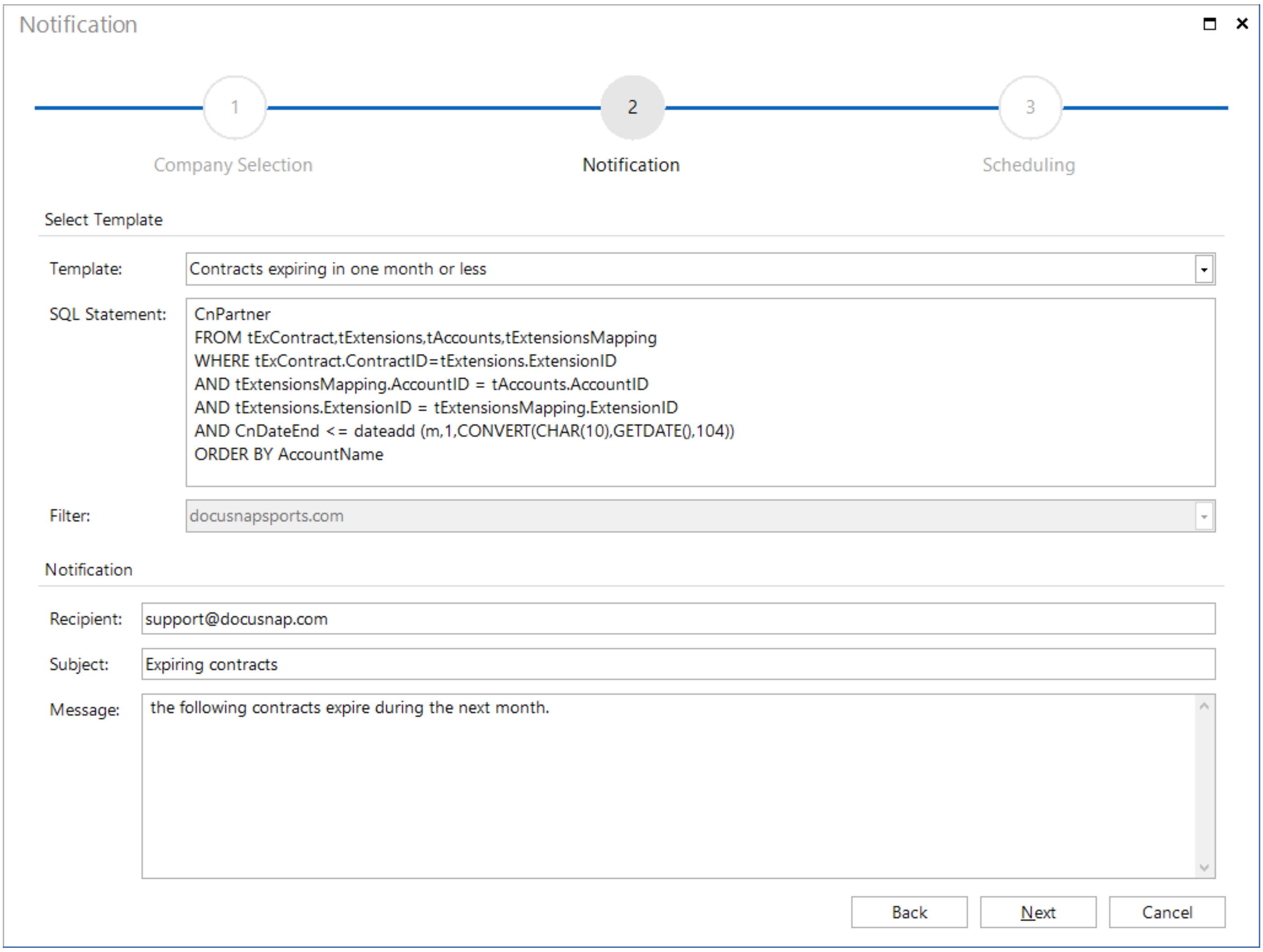
The second step is to select one of the existing message definitions and specify at least one recipient and subject for sending the email. A list of the information returned by the SQL query (e.g., expired contracts) is attached to the email as an Excel file.

If you want to store multiple recipients, you can separate them with a semicolon.

In the Scheduling step you define the execution of the notification task - e.g.

- Task type = Repeated
- Occurrence = Weekly
- Repeated every 1 week on Monday at 9:00 a.m.

Please note that the Excel file is given the name of the scheduled notification task. Please use only characters that are allowed in a file name here.



Select Template

Template: Contracts expiring in one month or less

SQL Statement:

```
CnPartner
FROM tExContract,tExtensions,tAccounts,tExtensionsMapping
WHERE tExContract.ContractID=tExtensions.ExtensionID
AND tExtensionsMapping.AccountID = tAccounts.AccountID
AND tExtensions.ExtensionID = tExtensionsMapping.ExtensionID
AND CnDateEnd <= dateadd (m,1,CONVERT(CHAR(10),GETDATE(),104))
ORDER BY AccountName
```

Filter: docusnapsports.com

Notification

Recipient: support@docusnap.com

Subject: Expiring contracts

Message: the following contracts expire during the next month.

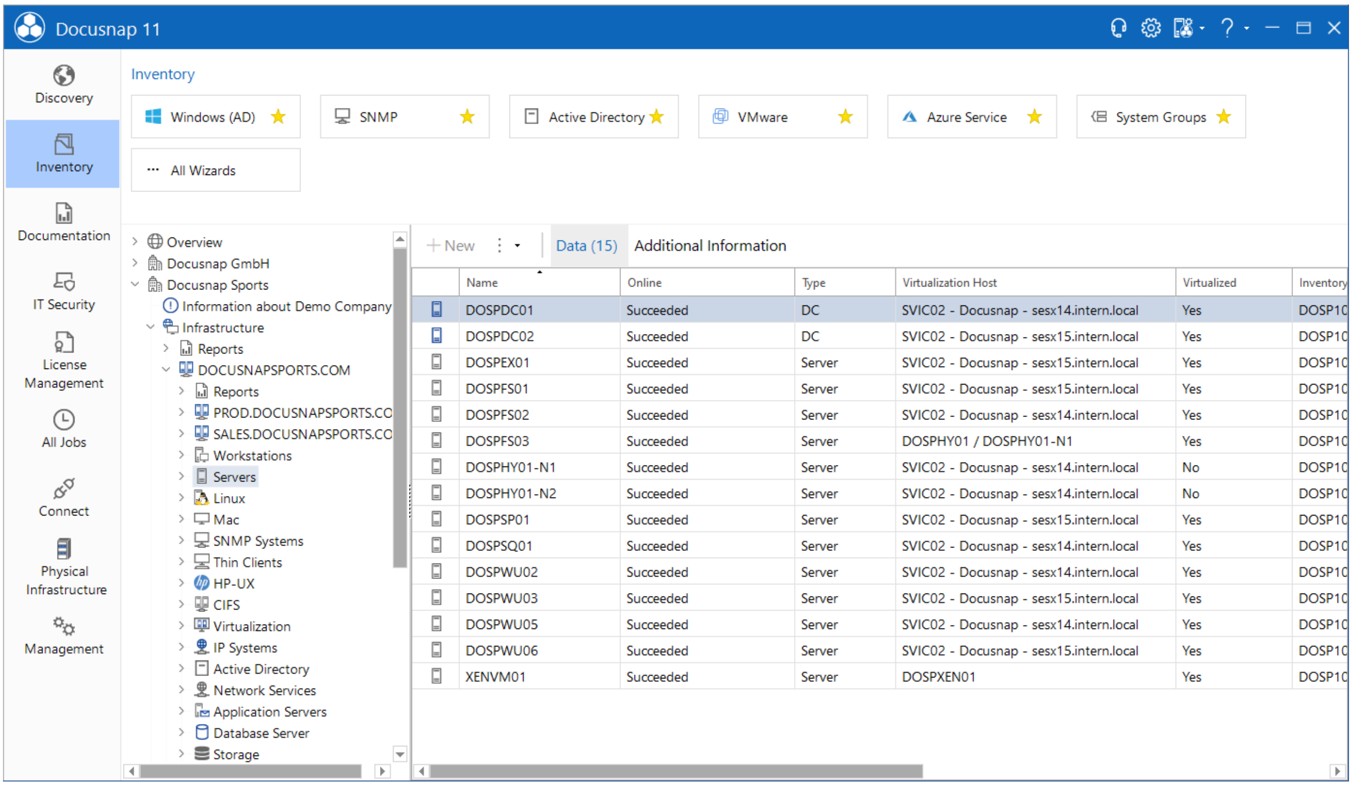
Fig. 5 - Create Notification

5. Maintenance tasks

As discussed previously, message definitions are SQL statements that are executed on the database via the notification task. Instead of SELECT queries, UPDATE statements can also be executed. The steps for creating the message definition, as well as the notification task, are the same as previously described.

Only for scheduling it makes sense to run the notification task more frequently - e.g., daily.

A regular use case for these maintenance tasks is, for example, the automatic filling of the virtualization host field of the inventoried virtual systems:



Name	Online	Type	Virtualization Host	Virtualized	Inventory
DOSPDC01	Succeeded	DC	SVIC02 - Docusnap - sesx14.intern.local	Yes	DOSP10
DOSPDC02	Succeeded	DC	SVIC02 - Docusnap - sesx15.intern.local	Yes	DOSP10
DOSPEX01	Succeeded	Server	SVIC02 - Docusnap - sesx15.intern.local	Yes	DOSP10
DOSPFS01	Succeeded	Server	SVIC02 - Docusnap - sesx15.intern.local	Yes	DOSP10
DOSPFS02	Succeeded	Server	SVIC02 - Docusnap - sesx14.intern.local	Yes	DOSP10
DOSPFS03	Succeeded	Server	DOSPHY01 / DOSPHY01-N1	Yes	DOSP10
DOSPHY01-N1	Succeeded	Server	SVIC02 - Docusnap - sesx14.intern.local	No	DOSP10
DOSPHY01-N2	Succeeded	Server	SVIC02 - Docusnap - sesx14.intern.local	No	DOSP10
DOSPSP01	Succeeded	Server	SVIC02 - Docusnap - sesx15.intern.local	Yes	DOSP10
DOSPSQ01	Succeeded	Server	SVIC02 - Docusnap - sesx14.intern.local	Yes	DOSP10
DOSPWU02	Succeeded	Server	SVIC02 - Docusnap - sesx14.intern.local	Yes	DOSP10
DOSPWU03	Succeeded	Server	SVIC02 - Docusnap - sesx15.intern.local	Yes	DOSP10
DOSPWU05	Succeeded	Server	SVIC02 - Docusnap - sesx14.intern.local	Yes	DOSP10
DOSPWU06	Succeeded	Server	SVIC02 - Docusnap - sesx15.intern.local	Yes	DOSP10
XENVM01	Succeeded	Server	DOSPXEN01	Yes	DOSP10

Fig. 6 - Virtualization host for inventoried systems

In addition to the actual systems, Docusnap also inventories the virtualization solutions HyperV, VMware and Citrix Hypervisor.

If the VM names match the system names, the information about which VM is running on which host can be automatically applied. Below you will find the examples with corresponding update statements.

VMWare

```
UPDATE tHosts SET VirtualServer = CONCAT(ServiceHostName, ' - ', DataCenterName, ' - ', ESXHostName)
FROM    tESXVirtualMachine,    tESXHost,    tHosts,    tESXComputeResource,    tESXDataCenterHostFolder,
tESXDataCenter, tESXFolder, tESXServiceDocu, tESXDataCenterFolder, tDomains AS HostDomain, tESXService,
tDomains AS ESXDomain, tESXVmHwNic, tESXVmHardware,
(
      select tNetworkAdapter.*, tHosts.HostID, tDomains.AccountID from tNetworkAdapter, tDocu, tHosts,
tDomains
      where tNetworkAdapter.AdapterID in (
      select tNetworkAdapterConfiguration.AdapterID from tNetworkAdapterConfiguration)
      and tNetworkAdapter.DocuID = tdocu.DocuID
      and tdocu.Archiv = 0
      and tdocu.HostID = thosts.HostID
      and tDomains.DomainID = thosts.DomainID
) as vNetworkAdapter
WHERE tESXVmHwNic.MacAddress = vNetworkAdapter.MACAddress
      and thosts.HostID = vNetworkAdapter.HostID
      AND tESXVirtualMachine.HostIdentifier = tESXHost.Identifier
      AND tESXComputeResource.ComputeResourceID = tESXHost.ComputeResourceID
      AND tESXDataCenterHostFolder.HostFolderID = tESXComputeResource.HostFolderID
      AND tESXDataCenter.DataCenterID = tESXDataCenterHostFolder.DataCenterID
      AND tESXFolder.ESXFolderID = tESXDataCenter.ESXFolderID
      AND tESXServiceDocu.DocuID = tESXFolder.DocuID
      AND tESXDataCenterFolder.DataCenterID = tESXDataCenter.DataCenterID
      AND tESXDataCenterFolder.FolderID = tESXVirtualMachine.FolderID
      AND tESXServiceDocu.Archiv = 0
      AND tHosts.DomainID = HostDomain.DomainID
      AND tESXServiceDocu.ServiceID = tESXService.ServiceID
      AND tESXService.DomainID = ESXDomain.DomainID
      AND tESXVmHardware.VirtualMachineID = tESXVirtualMachine.VirtualMachineID
      AND tESXVmHwNic.VmHardwareID = tESXVmHardware.VmHardwareID
      AND HostDomain.AccountID = ESXDomain.AccountID
      AND vNetworkAdapter.AccountID = HostDomain.AccountID
      AND ESXDomain.AccountID = {ACCOUNT}                --Filtering on the selected company
      AND HostDomain.AccountID = {ACCOUNT}               --Filtering on the selected company
```

The statement adds the following information as a virtualization host:

- ESX Service Name (the inventoried vCenter).
- Data Center Name
- ESX Host

Furthermore, within the statement, the last two columns filter on the account/company for which the notification task is set up. If there are multiple companies in your Docusnap database and you do not want to set up a separate notification task for each, then simply remove the last two lines.

HyperV

```
update tHosts set VirtualServer = tHVNode.Name
from tHosts, tHVNode, tHVComputerSystem, tHVDocu, tDocu, tHVSystemService, tDomains AS HostDomain,
tDomains AS HyperVDomain, tHVVirtualSystemSettings, tHVEthernet,
(
      select tNetworkAdapter.*, tHosts.HostID, tDomains.AccountID from tNetworkAdapter, tDocu, tHosts,
tDomains
      where tNetworkAdapter.AdapterID in (
      select tNetworkAdapterConfiguration.AdapterID from tNetworkAdapterConfiguration)
      and tNetworkAdapter.DocuID = tdocu.DocuID
      and tdocu.Archiv = 0
      and tdocu.HostID = thosts.HostID
      and tDomains.DomainID = thosts.DomainID
) as vNetworkAdapter
where tHVNode.HVNodeID = tHVComputerSystem.HVNodeID
      and thosts.HostID = vNetworkAdapter.HostID
      and tHVComputerSystem.HVComputerSystemID = tHVVirtualSystemSettings.HVComputerSystemID
      and tHVEthernet.HVVirtualSystemSettingsID = tHVVirtualSystemSettings.HVVirtualSystemSettingsID
      and vNetworkAdapter.MACAddress = tHVEthernet.Address
      and tHosts.HostID = tDocu.HostID
      and tHVNode.HVDocuID = tHVDocu.HVDocuID
      and tHVSystemService.HVSystemServiceID = tHVDocu.HVSystemServiceID
      and tDocu.Archiv = 0
      and tHVDocu.Archiv = 0
      and tHosts.DomainID = HostDomain.DomainID
      and tHVSystemService.DomainID = HyperVDomain.DomainID
      and HostDomain.AccountID = HyperVDomain.AccountID
      and HostDomain.AccountID = vNetworkAdapter.AccountID
      and HyperVDomain.AccountID = {ACCOUNT}              -- Filtering on the selected company
      and HostDomain.AccountID = {ACCOUNT}                -- Filtering on the selected company
```

The statement adds the HyperV host.

Furthermore, the last two columns within the statement filter on the account/company for which the notification task is set up. If there are multiple companies in your Docusnap database and you do not want to set up a notification task for each of them, simply remove the last two lines.

Citrix Hypervisor

```
UPDATE tHosts SET VirtualServer = tXenHost.Name
FROM tXenHost, tXenHostVMs, tXenPoolDocu, tHosts, tXen, tDomains AS HostDomain, tDomains AS XenDomain,
tXenVMNICs,
(
      select tNetworkAdapter.*, tHosts.HostID, tDomains.AccountID from tNetworkAdapter, tDocu, tHosts,
tDomains
      where tNetworkAdapter.AdapterID in (
      select tNetworkAdapterConfiguration.AdapterID from tNetworkAdapterConfiguration)
      and tNetworkAdapter.DocuID = tdocu.DocuID
      and tdocu.Archiv = 0
      and tdocu.HostID = thosts.HostID
      and tDomains.DomainID = thosts.DomainID
) as vNetworkAdapter
WHERE tXenHost.HostID = tXenHostVMs.HostID
      and tHosts.HostID = vNetworkAdapter.HostID
      and tXenVMNICs.MAC = vNetworkAdapter.MACAddress
      AND tXenVMNICs.VMID = tXenHostVMs.VMID
      AND tXenPoolDocu.DocuID = tXenHost.DocuID
      and tHosts.DomainID = HostDomain.DomainID
      and tXenPoolDocu.XenID = tXen.XenID
      and tXen.DomainID = XenDomain.DomainID
      and HostDomain.AccountID = XenDomain.AccountID
      and vNetworkAdapter.AccountID = HostDomain.AccountID
      AND tXenPoolDocu.Archiv = 0
      and XenDomain.AccountID = {ACCOUNT}          -- Filtering on the selected company
      and HostDomain.AccountID = {ACCOUNT}         -- Filtering on the selected company
```

The statement adds the Xen host.

Furthermore, within the statement, the last two columns filter on the account/company for which the notification task is set up. If there are multiple companies in your Docusnap database and you do not want to set up a separate notification task for each, then simply remove the last two lines.

LIST OF FIGURES

VERSION HISTORY

Date	Description
1/3/2016	Statement modified (2.4.6)
3/3/2017	List of figures modified
24/10/2018	Screenshots updated; Statement modified (2.4.5, 2.4.7 und 2.4.8); Insert Version history
29/04/2020	Version 2.0 - Revision of the HowTo for Docusnap 11