



DocuSnap X - Defining Notifications

E-mail Reminders

TITLE	Docusnap X - Defining Notifications
AUTHOR	Docusnap Consulting
DATE	18/12/2018
VERSION	1.5 valid as of September 26, 2018

The reproduction and distribution of this document as a whole or in part as well as the utilization and disclosure of its contents to third parties without the express authorization by itelio GmbH are prohibited. Offenders will be held liable for the payment of indemnification. All rights reserved.

TABLE OF CONTENTS

1. Introduction	4
2. Notification definitions	5
2.1 Basics	5
2.2 Accessing the notification definition	5
2.3 Creating a definition	6
2.4 Application samples	7
2.4.1 General	7
2.4.2 Checking the SQL time format	7
2.4.3 Preliminary check of notification definitions	8
2.4.4 Contracts	9
2.4.5 Contracts with no object assignment	10
2.4.6 Licenses	11
2.4.7 Query by inventory date	12
2.4.8 Query of the different host types	12
2.5 Creating a notification	13
2.6 Information on SQL	14
2.6.1 DATEADD function	14
2.6.2 Converting the time format	15
2.6.3 SQL date / time format	16

1. Introduction

Docusnap allows you to send various notifications about various events in an automated way by e-mail.

This document describes how to create a notification in Docusnap.

Application samples show you how to extend the notification definitions shipped and installed with Docusnap.

The prerequisite for using the notification e-mailing feature is that the Docusnap Server has been configured and an e-mail server is available. More information on this topic can be found in the Docusnap User Manual under <https://www.docusnap.com/help/docusnap-x/user/docusnap-server.html>.

2. Notification definitions

2.1 Basics

Notification definitions are SQL queries against the Docusnap database.

Notification definitions that report invalid licenses or expired contracts, and include reminders already come with Docusnap. These templates, however, are defined in such a way that the notifications are only sent out on the day of the event (e.g. license expiration). The application samples explain, in particular, how to set up a lead time for the notification which can be adapted to your own requirements.

2.2 Accessing the notification definition

The wizard for the notification definition can be opened from the *SCHEDULING* module.

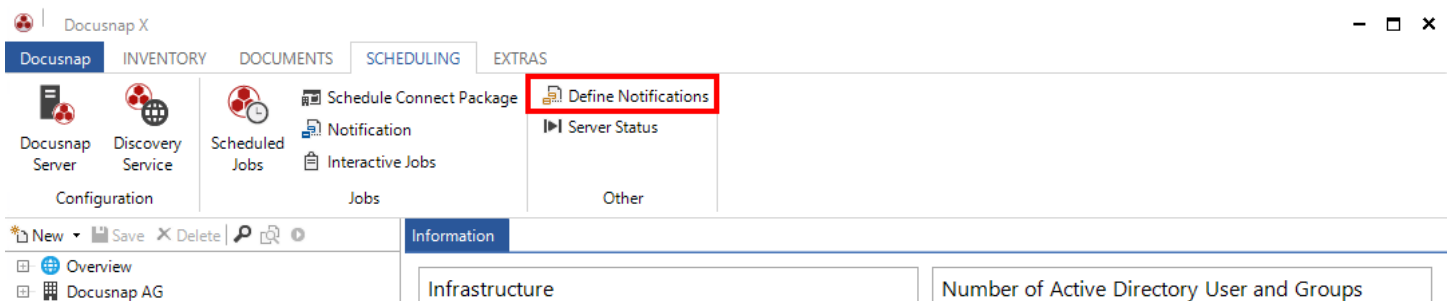


Fig. 1 - Accessing the notification definition

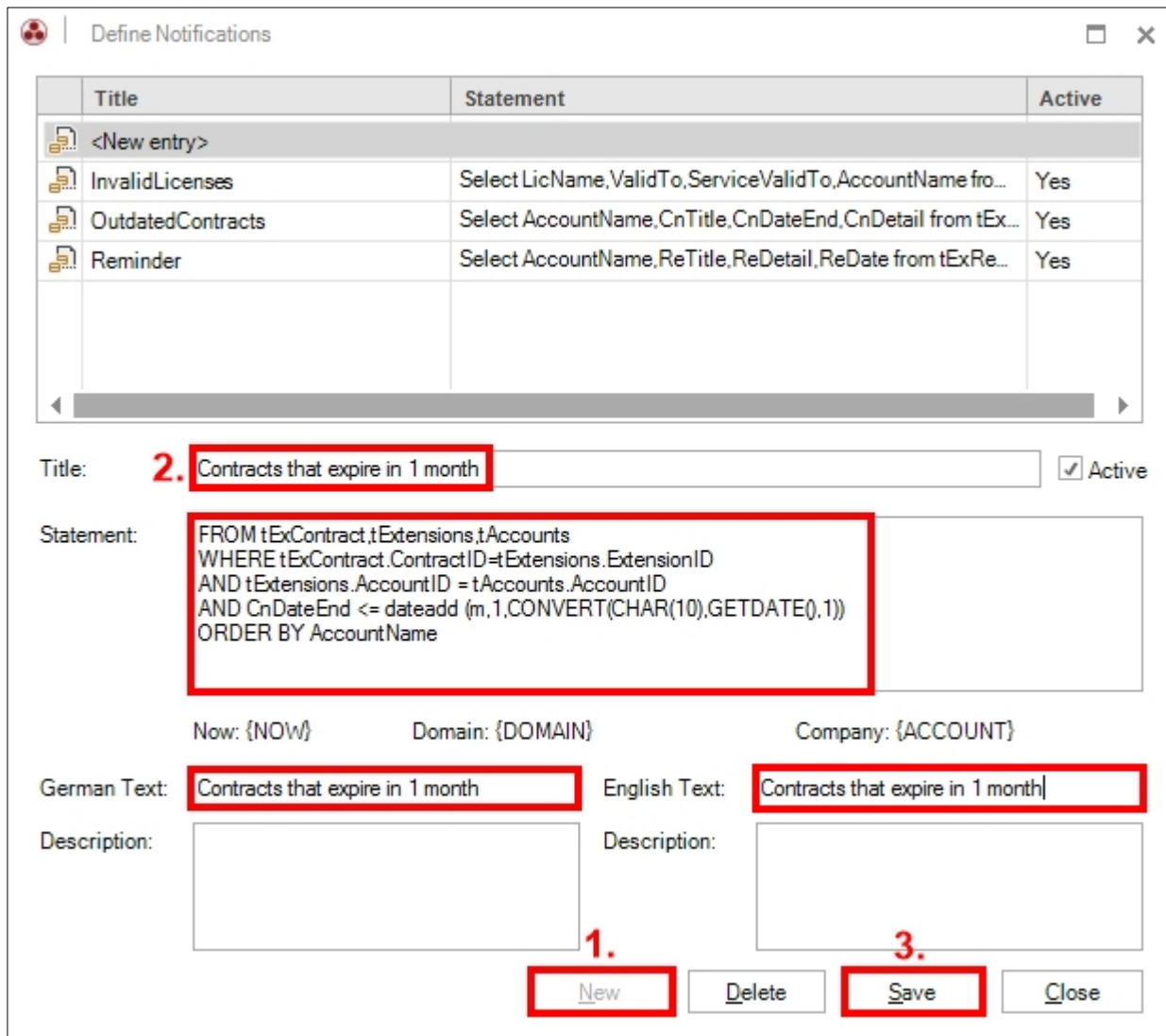
2.3 Creating a definition

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'.



Define Notifications

Title	Statement	Active
<New entry>		
InvalidLicenses	Select LicName,ValidTo,ServiceValidTo,AccountName fro...	Yes
OutdatedContracts	Select AccountName,CnTitle,CnDateEnd,CnDetail from tEx...	Yes
Reminder	Select AccountName,ReTitle,ReDetail,ReDate from tExRe...	Yes

Title: **2.** Active

Statement:

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

German Text: English Text:

Description: Description:

1. **3.**

Fig. 2 - Creating a notification definition

2.4 Application samples

2.4.1 General

In the application samples, notification definitions already supplied with Docusnap will be extended.

For more information on the required SQL commands/functions, see the INFORMATION ON SQL section.

2.4.2 Checking the SQL time format

Before using the following examples you need to check the time format setting at your SQL-Server.

The following examples use the US-time format with a 4-digit year („101“). The time format code needs to be adjusted in other regions.

Using the wrong time format in a notification definition will lead to an empty result within Docusnap, but it is not showing an error message.

Therefore it is strongly recommended to check the definition of a notification using the SQL-Management-Studio.

For a table on date / time formats and the SQL-query to check the time format, refer to the section 2.6.3 SQL DATE / TIME FORMAT

2.4.3 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.

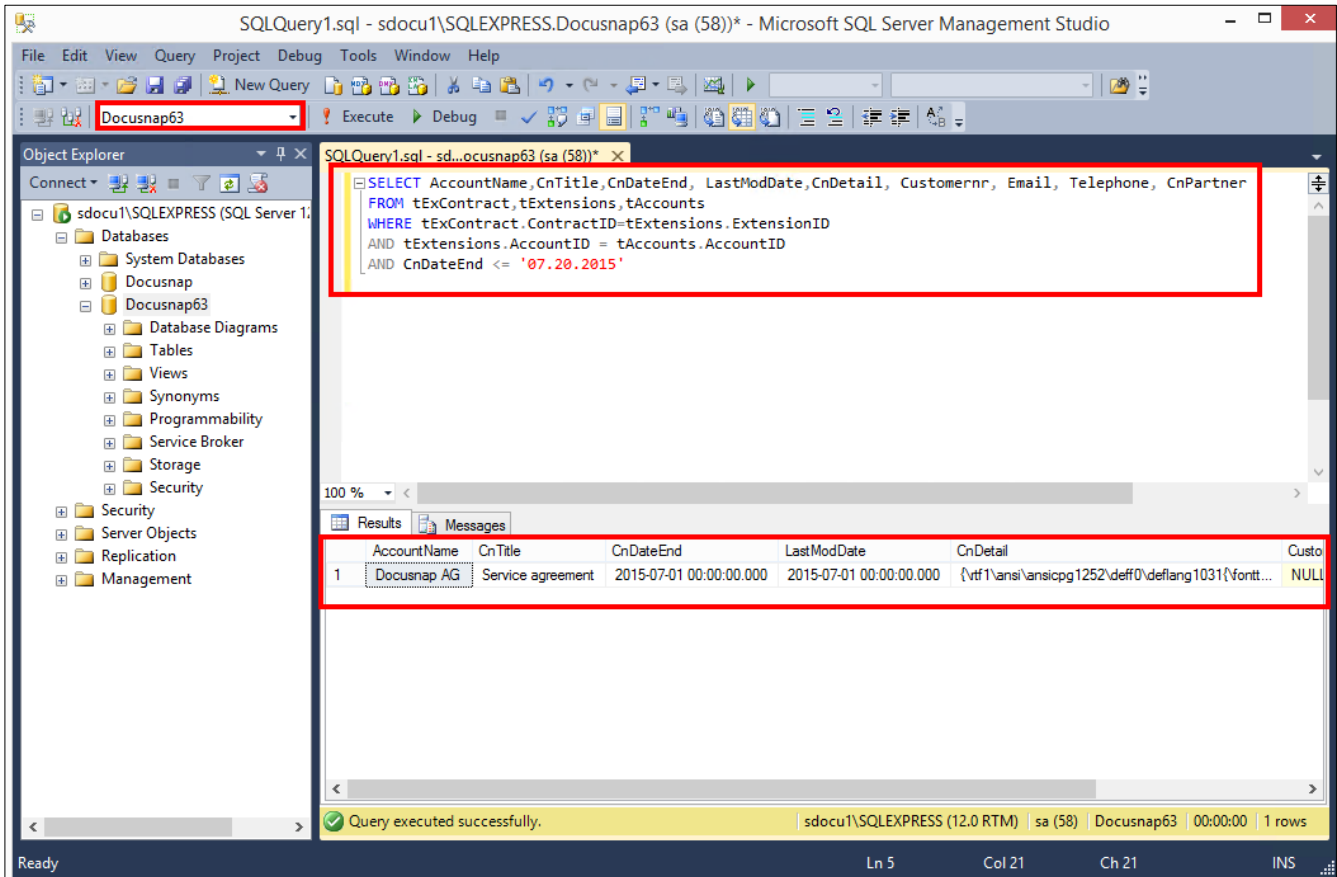


Fig. 3 - Accessing the SQL queries in SQL Server Management Studio

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'	Make sure to indicate the date in the correct format as used in the database. For more information, see the SQL DATE / TIME FORMAT section.
Domain	{DOMAIN}	DomainID	The Docusnap domain IDs can be displayed via the following SQL query: <code>SELECT TOP 100 [DomainID],[DomainName] FROM [tDomains]</code>
Company	{ACCOUNT}	AccountID	The Docusnap account IDs can be displayed via the following SQL query: <code>SELECT TOP 100 [AccountID],[AccountName] FROM [tAccounts]</code>

2.4.4 Contracts

The following SQL query lists all expired contracts assigned to an object:

```
SELECT AccountName,CnTitle,CnDateEnd,CnDetail
FROM tExContract,tExtensions,tAccounts,tExtensionsMapping
WHERE tExContract.ContractID=tExtensions.ExtensionID
AND tExtensionsMapping.AccountID = tAccounts.AccountID
AND tExtensions.ExtensionID = tExtensionsMapping.ExtensionID
AND CnDateEnd <= {now}
```

A modified version supplies additional information on the expired contract (e.g. associated e-mail address or phone number):

```
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 <= {now}
```

Please note that only contracts assigned to an object are considered in the above statements. The reason for this is the use of INNER JOINS with the included table "tExtensionsMapping".

If Contracts are also to be considered which are not assigned to any object, then OUTER JOINS must be used for this purpose, which is described by means of an example in the next subitem.

2.4.5 Contracts with no object assignment

The following SQL query lists all expired contracts:

```
SELECT tEC.CnTitle, tEC.CnDateEnd, tEC.CnDetail, tA.AccountName FROM tExContract tEC
LEFT OUTER JOIN tExtensions tE ON tE.ExtensionID = tEC.ContractID
LEFT OUTER JOIN tExtensionsMapping tEM ON tEM.ExtensionID = tE.ExtensionID
LEFT OUTER JOIN tAccounts tA ON tEM.AccountID = tA.AccountID
AND CnDateEnd <= {now}
```

A modified version lists contracts which expire within one month or earlier. This data is output with additional information on the contracts:

```
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,1,CONVERT(CHAR(10),GETDATE()),101))
ORDER BY AccountName
```

2.4.6 Licenses

The following SQL query lists all expired licenses of a company:

```
SELECT LicName,ValidTo,ServiceValidTo,AccountName
FROM tSoftwareLicenses,tSoftwareProducts,tAccounts
WHERE tSoftwareLicenses.SoftwareID=tSoftwareProducts.SoftwareID
AND tSoftwareProducts.AccountID = tAccounts.AccountID
AND tAccounts.AccountID = {ACCOUNT}
AND tSoftwareLicenses.ValidInfinite=0
AND ValidTo <= {now}
```

This query retrieves all licenses of a company which expire within one month or earlier:

```
SELECT DISTINCT AccountName, LicName, SoftwareName,SoftwarePublisher, SoftwareVersion, ValidTo,
ServiceValidTo, Customernr, Telephone, Email
FROM tSoftwareLicenses,tSoftwareProducts,tAccounts
WHERE tSoftwareLicenses.SoftwareID=tSoftwareProducts.SoftwareID
AND tSoftwareProducts.AccountID = tAccounts.AccountID
AND tAccounts.AccountID = {ACCOUNT}
AND tSoftwareLicenses.ValidInfinite=0
AND ValidTo <= dateadd (m,1,CONVERT(CHAR(10),GETDATE()),101))
ORDER BY AccountName, LicName
```

The following query retrieves the licenses of all companies which expire within one month or earlier:

```
SELECT DISTINCT AccountName, LicName, SoftwareName,SoftwarePublisher, SoftwareVersion, ValidTo,
ServiceValidTo, Customernr, Telephone, Email
FROM tSoftwareLicenses,tSoftwareProducts,tAccounts
WHERE tSoftwareLicenses.SoftwareID=tSoftwareProducts.SoftwareID
AND tSoftwareProducts.AccountID = tAccounts.AccountID
AND tSoftwareLicenses.ValidInfinite=0
AND ValidTo <= dateadd (m,1,CONVERT(CHAR(10),GETDATE()),101))
ORDER BY AccountName, LicName
```

The following query retrieves the licenses of all companies which expire within the next 3 months:

```
SELECT DISTINCT AccountName, LicName, SoftwareName,SoftwarePublisher, SoftwareVersion, ValidTo,
ServiceValidTo, Customernr, Telephone, Email
FROM tSoftwareLicenses,tSoftwareProducts,tAccounts
WHERE tSoftwareLicenses.SoftwareID=tSoftwareProducts.SoftwareID
AND tSoftwareProducts.AccountID = tAccounts.AccountID
AND tSoftwareLicenses.ValidInfinite=0
AND ValidTo >= CONVERT(CHAR(10),GETDATE()),104)
AND ValidTo <= dateadd (m,3,CONVERT(CHAR(10),GETDATE()),101))
ORDER BY AccountName, LicName
```

2.4.7 Query by inventory date

This query allows you to query the workstations which have been inventoried last one week ago or earlier:

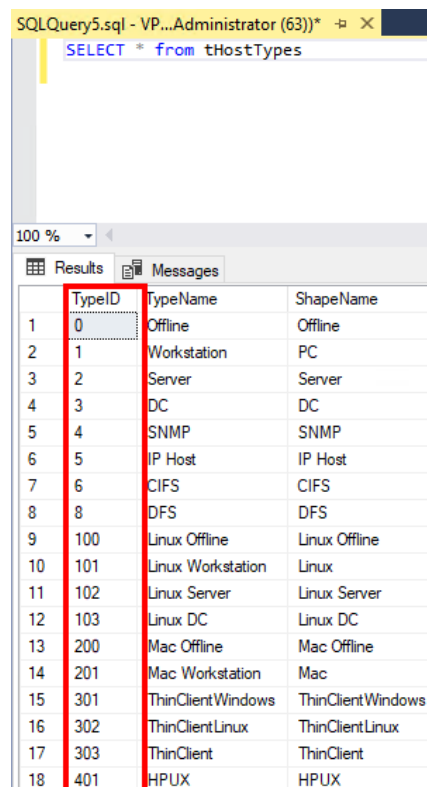
```
SELECT HostName, HostTypeID, DomainName, AccountName, ScanDate FROM tHosts h
INNER JOIN tDocu d on h.HostID = d.HostID
INNER JOIN tDomains do on h.DomainID = do.DomainID
INNER JOIN tAccounts a on do.AccountID = do.AccountID
WHERE h.HostTypeID = 1
AND ScanDate <= DATEADD (w, -1, CONVERT(char(10), GETDATE(), 101))
AND d.Archiv = 0
```

The "1" used in the example stands for the host type workstations. If the query should also consider e.g. server or domain controller, the HostTypeID "2" and "3" must be used for this.

A table of the different host types can be found in the screenshot below.

2.4.8 Query of the different host types

In the following screenshot, you can use the statement to list the different host types.



TypeID	typeName	ShapeName
0	Offline	Offline
1	Workstation	PC
2	Server	Server
3	DC	DC
4	SNMP	SNMP
5	IP Host	IP Host
6	CIFS	CIFS
8	DFS	DFS
100	Linux Offline	Linux Offline
101	Linux Workstation	Linux
102	Linux Server	Linux Server
103	Linux DC	Linux DC
200	Mac Offline	Mac Offline
201	Mac Workstation	Mac
301	ThinClientWindows	ThinClientWindows
302	ThinClientLinux	ThinClientLinux
303	ThinClient	ThinClient
401	HPUX	HPUX

Fig. 4 – Table for the different host types

2.5 Creating a notification

The wizard for creating notifications can be opened from the *SCHEDULING* module.

You need to select a template and specify at least one recipient and a subject for e-mailing. A list of the information returned by the SQL query (e.g. expiring contracts) will be attached to the e-mail.

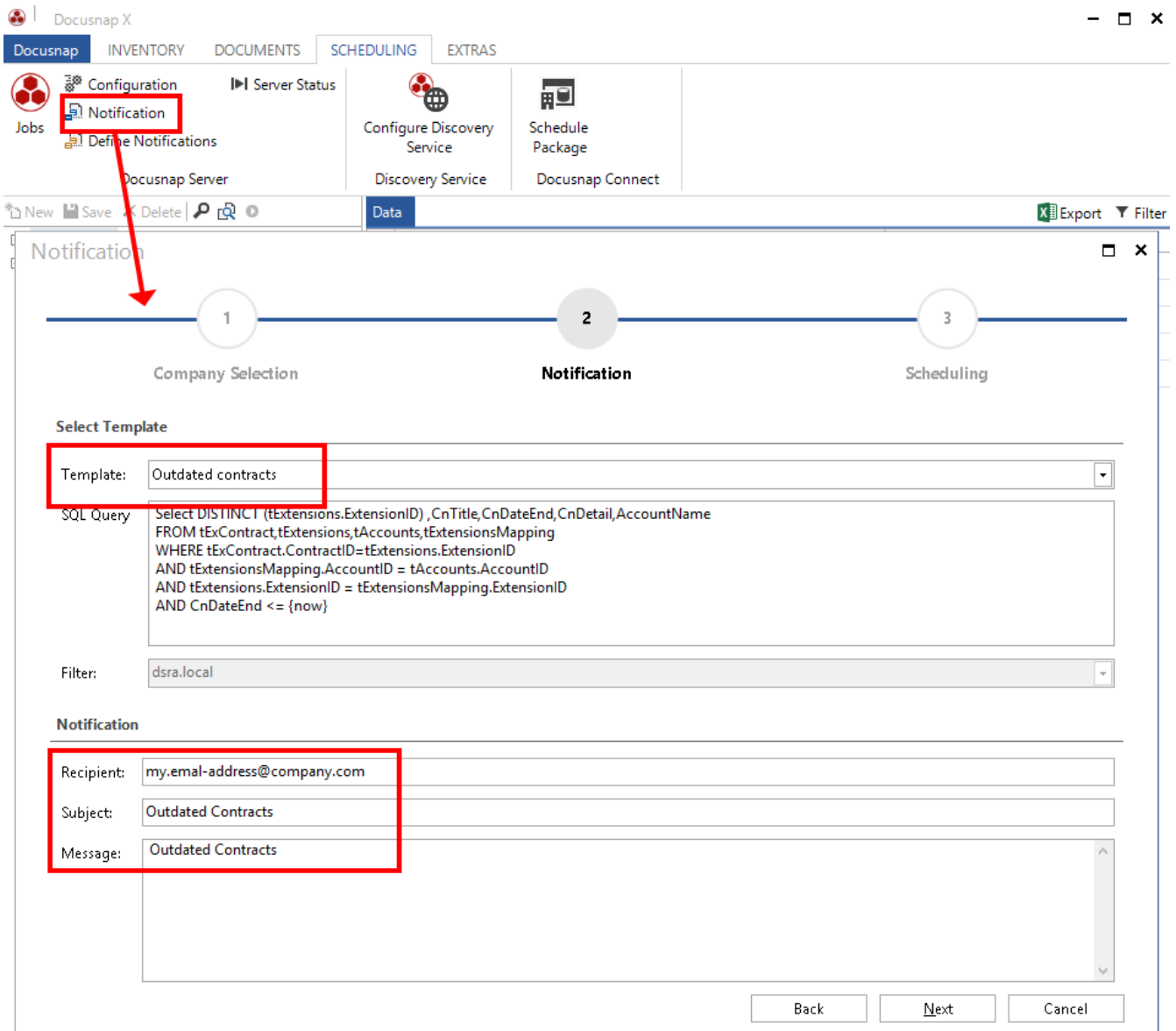


Fig. 5 – Create Notification

In step three – scheduling – it is possible to send configure the notification to be sent in periodic intervals.

2.6 Information on SQL

2.6.1 DATEADD function

To specify a time in the future, you can use the SQL DATEADD function:

```
DATEADD (datepart , number , date )
```

For DATEADD, the following parameters can be used:

datepart

datepart	Abbreviations
year	yy, yyyy
quarter	qq, q
month	mm, m
dayofyear	dy, y
day	dd, d
week	wk, ww
weekday	dw, w

number

This parameter defines the number of datepart values to be added to the date. It is possible to specify user-defined variables. If you enter a value with a decimal fraction, the fraction part will be truncated rather than rounded.

date

This is an expression which can be resolved to a date / time format. To avoid ambiguities, specify the year with four digits.

Example:

```
select DATEADD (m,1,'10-27-2011')
```

This queries the following date: October 27, 2011. 'm' is the placeholder for month and the expression '1' adds a month to the specified date.

2.6.2 Converting the time format

To retrieve the current date in SQL, it is necessary to convert the format.

For the conversion, use the **Convert** function:

```
CONVERT(CHAR(10),GETDATE(),101)
```

Depending on the specific language used by the SQL server, the corresponding parameter needs to be set. In this case '101' for the four-digit year date format used in the US.

For a table on date / time formats, refer to the next section.

Example using DATEADD and CONVERT:

```
CnDateEnd = dateadd (m,1,CONVERT(CHAR(10),GETDATE(),101))
```

2.6.3 SQL date / time format

The following command can be used to query the current date / time format from the *MASTER* database.

```
SELECT dateformat FROM master..syslanguages WHERE name = @@LANGUAGE
```

Select the correct format from the following list:

Style Code	Style	Format	Example
0 or 100	Default. Equivalent to not specifying a style code.	mon dd yyyy hh:mmAM	Sep 8 2007 9:00PM
1	USA date.	mm/dd/yy	39303
2	ANSI date.	yy.mm.dd	39698
3	UK / French date.	dd/mm/yy	39333
4	German date.	dd.mm.yy	39333
5	Italian date.	dd-mm-yy	39333
6	Abbreviated month.	dd mmm yy	39333
7	Abbreviated month.	mmm dd, yy	Sep 08, 07
8 or 108	24-hour time.	HH:mm:ss	0.875
9 or 109	Default formatting with seconds and milliseconds appended.	mon dd yyyy hh:mm:ss:fffAM	Sep 8 2007 9:00:00:000PM
10	USA date with hyphen separators.	mm-dd-yy	39303
11	Japanese date.	yy/mm/dd	39698
12	ISO date.	yymmdd	70908
13 or 113	European default with seconds and milliseconds.	dd mon yyyy HH:mm:ss:fff	08 Sep 2007 21:00:00:000
14 or 114	24 hour time with milliseconds.	HH:mm:ss:fff	21:00:00:000
20 or 120	ODBC canonical date and time.	yyyy-mm-dd HH:mm:ss	39333.875
21 or 121	ODBC canonical date and time with milliseconds.	yyyy-mm-dd HH:mm:ss:fff	2007-09-08 21:00:00.000
101	USA date with century.	mm/dd/yyyy	39303
102	ANSI date with century.	yyyy.mm.dd	39333
103	UK / French date with century.	dd/mm/yyyy	39333
104	German date with century.	dd.mm.yyyy	39333
105	Italian date with century.	dd-mm-yyyy	39333
106	Abbreviated month with century.	dd mmm yyyy	39333
107	Abbreviated month with century.	mmm dd, yyyy	Sep 08, 2007
110	USA date with hyphen separators and century.	mm-dd-yyyy	39303
111	Japanese date with century.	yyyy/mm/dd	39333
112	ISO date with century.	yymmdd	20070908
126	ISO8601, for use in XML.	yyyy-mm-ddThh:mm:ss	2007-09-08T21:00:00

Table 1: Format codes

LIST OF FIGURES

FIG. 1 - ACCESSING THE NOTIFICATION DEFINITION.....	5
FIG. 2 - CREATING A NOTIFICATION DEFINITION.....	6
FIG. 3 - ACCESSING THE SQL QUERIES IN SQL SERVER MANAGEMENT STUDIO.....	8
FIG. 4 – TABLE FOR THE DIFFERENT HOST TYPES.....	12
FIG. 5 – CREATE NOTIFICATION.....	13

LIST OF TABLES

TABLE 1: FORMAT CODES.....	16
----------------------------	----

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
