# dox42 AX

## Documentation

Solutions Factory / dox42

# Table of Contents


Table of Contents ........ 2

What is dox42 AX? ........ 3

Installation for the dox42 Office Add-Ins ........ 3

System Requirements ........ 4

Installation of dox42 AX for dox42 Server ........ 4

Installation of dox42 AX Model on AX 2012 R3 CU9 ........ 6

Implementation of dox42 within AX ........ 7

Preparations ........ 7

Configuration ........ 8

Templates and update processes ........ 11

Example (sales invoice) ........ 12

Process for replacing an SSRS report with dox42 ........ 15

# What is dox42 AX?

The dox42 AX data source enables you to use AX-data for document automation. What is more, multiple data sources can be integrated into one document at the same time.

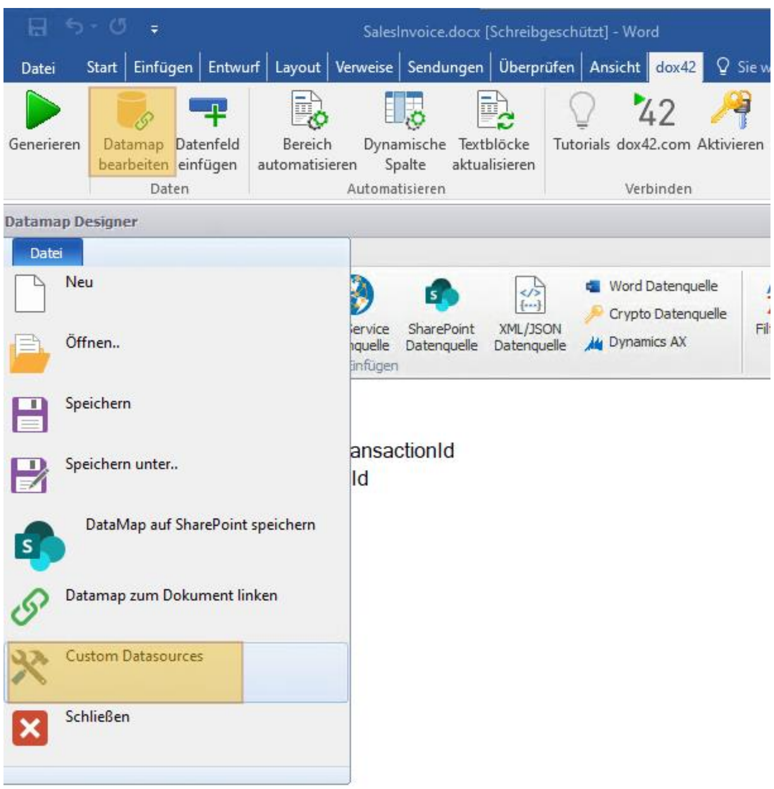
dox42 AX is a supplementary package for dox42. dox42 AX is a Custom Data Source.

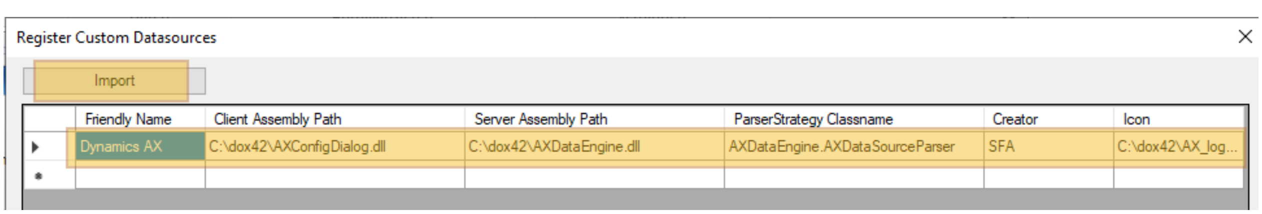
A dox42 Custom Data Source enables access to specific data sources or systems, additionally to the already predefined data sources.

# Installation for the dox42 Office Add-Ins

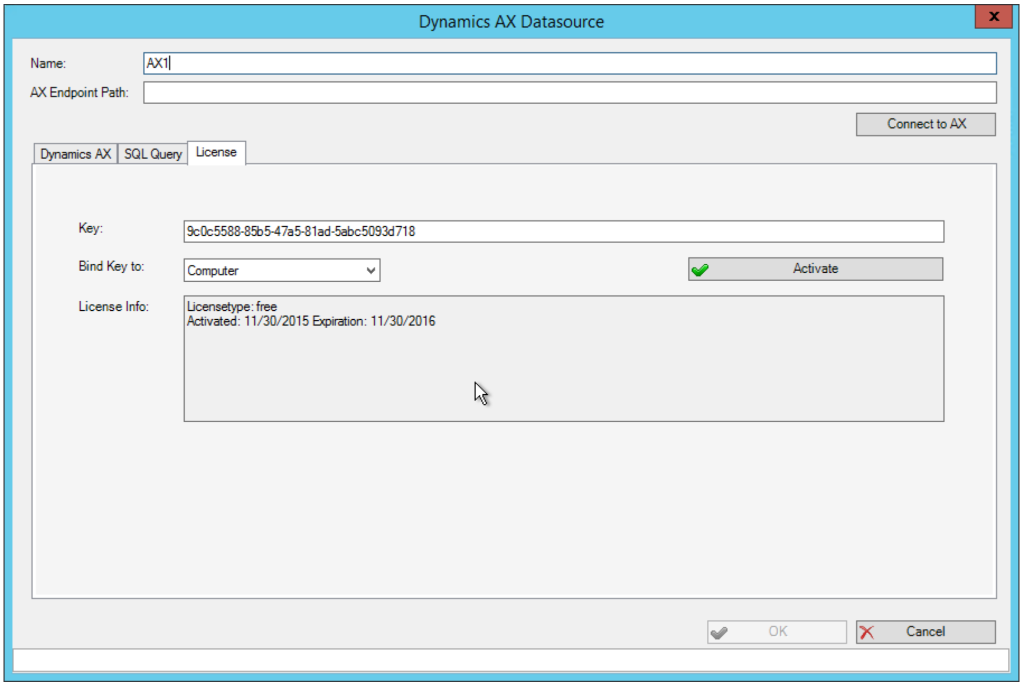
Dox42 report designer users need access to Word, to the dox42 client add-in, and to the dox42 AX client add-in extension.

Importing the Config for the AX Addin (AXDataSourceImport.config) as Custom Data Source:



Activating the license for the dox42 AX Client:




## System Requirements

- Activated dox42 Word and dox42 Server
- Microsoft Dynamics AX 2012 R3 CU9

## Installation of dox42 AX for dox42 Server

In order to use the Custom AX Data Source on the dox42 Server, please copy the following Assemblies in the /Bin directory on your dox42 Server:

- AXDataEngine.dll

Now the Custom AX Data Source has to be registered in the **web.config** of your server.

```
<customDataSources>

        <add key="AXDataEngine.AXDataSourceParser"
              value="AX Custom DS;c:\dox42Server\bin\AXDataEngine.dll;"/>

</customDataSources>
```

Furthermore, please insert the license key in the **web.config**:

```
<appSettings>

        <add key="LicenseFilePath" value="c:\dox42Server\License" />

        <add key="LicenseKey" value="…"/>
```

```
        <add key="dox42AXLicenseKey" value="…"/>

        <add key="MailServer" value="smtp.1und1.de"/>

        <add key="dox42ServerMail" value="mydox42server@dox42.com"/>

        <add key="MailServerPassWord" value="…"/>

        <add key="SendErrorsToMail" value="mymonitoring@dox42.com"/>

        <add key="Certificate" value="c:\ dox42Server\Certs\Certificate.pfx"/>

        <add key="CertificatePassWord" value="…"/>

        <add key="CheckServiceHeader" value="true"/>

        <add key="UserPermissionODBCConnectionString"
              value="Driver={SQL Server Native Client 10.0};Server=…;Database=…;Uid=…;Pwd=…;"/>

        <add key="CheckUserPermissionSQL"
              value="SELECT COUNT(*) FROM Users where Username =
              '%username%' and Password = '%password%'" />

        <add key="ThrowExceptionOnError" value="false"/>

</appSettings>
```

For further details slso take a look at the dox42 Server Documentation, available at our resource page: [https://www.dox42.com/Resources?filter=.dokumentation&search=%2Fserver%2Fi](https://www.dox42.com/Resources?filter=.dokumentation&search=%2Fserver%2Fi)

# Installation of dox42 AX Model on AX 2012 R3 CU9

Start the **Microsoft Dynamics AX Management Shell** as Administrator. At the command prompt enter

Install-AXModel –File <PathTo_dox42.axmodel>

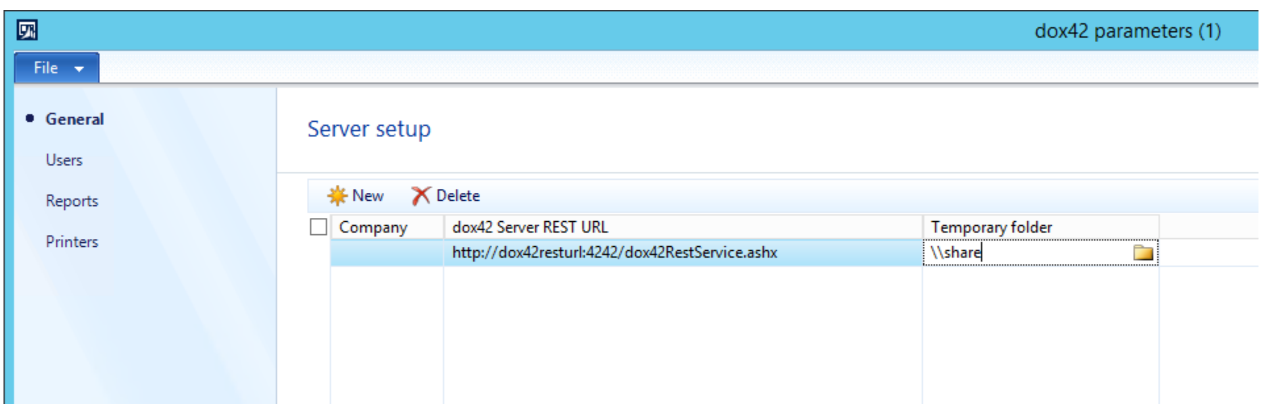
to import the model into the default AX model store. To continue the installation, enter **Y** when you are prompted with “The model is not signed. Are you sure you want to install this model (Y/N)?”.

For further information on how to import an .axmodel file go to: [https://technet.microsoft.com/en-us/library/hh352314.aspx](https://technet.microsoft.com/en-us/library/hh352314.aspx).

Restart the AOS service, and start the AX 2012 client afterwards. Please follow the upgrade checklist. Important steps are:

- Compile the model in X++
- Generate a full CIL
- Synchronize the data dictionary
- Deploy the web service called **SFADox42MSOfficeServices**

Under **System administration > Setup > Business intelligence > dox42 > dox42 parameters**, select the **General** tab page.
Create a new Server setup record by clicking on **New**.
Under the column **dox42 Server REST URL**, enter the path to the dox42 Server service. At **temporary folder**, enter the directory which AX and dox42 will use, to store temporary files.

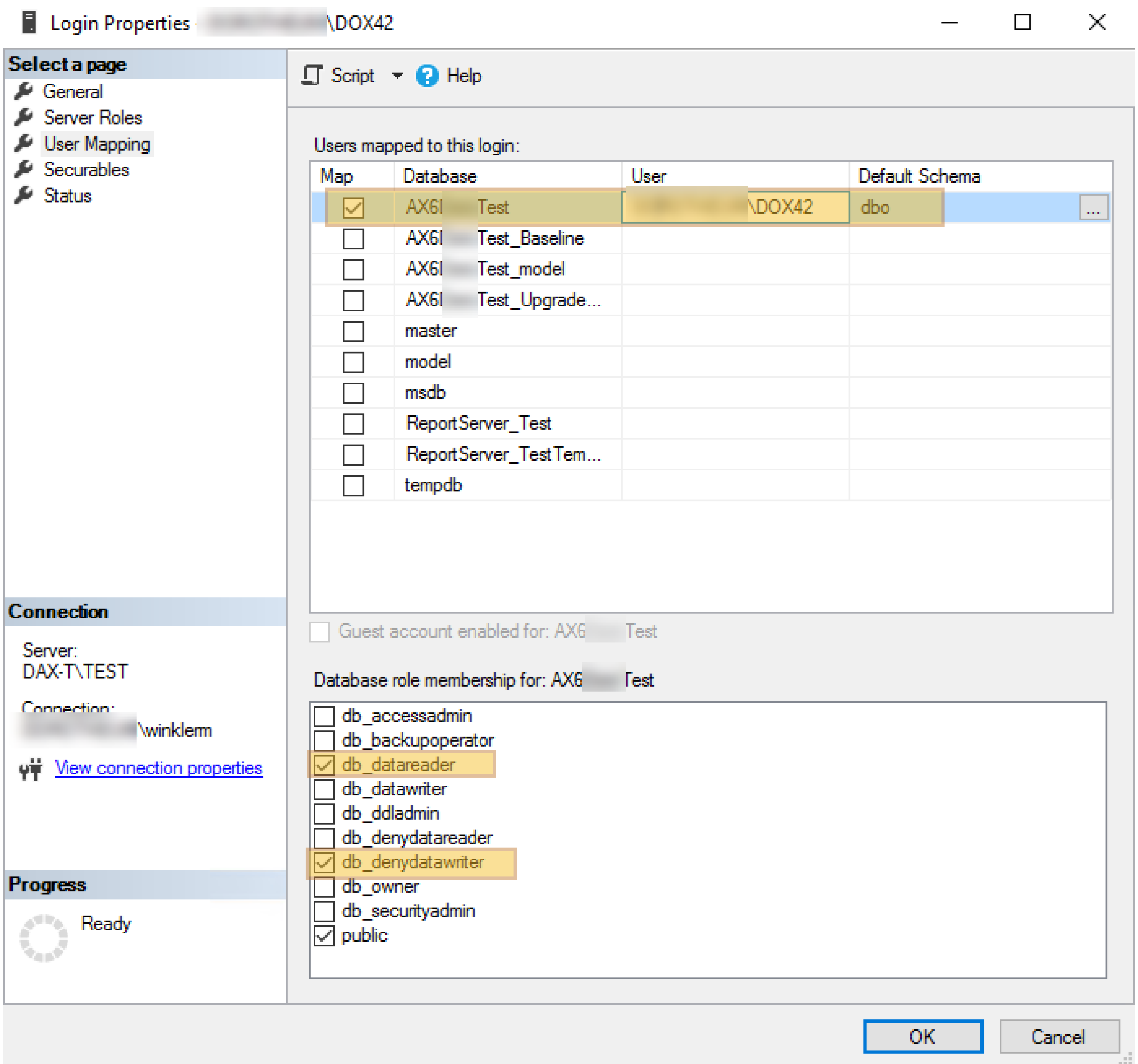


# Implementation of dox42 within AX

## Preparations

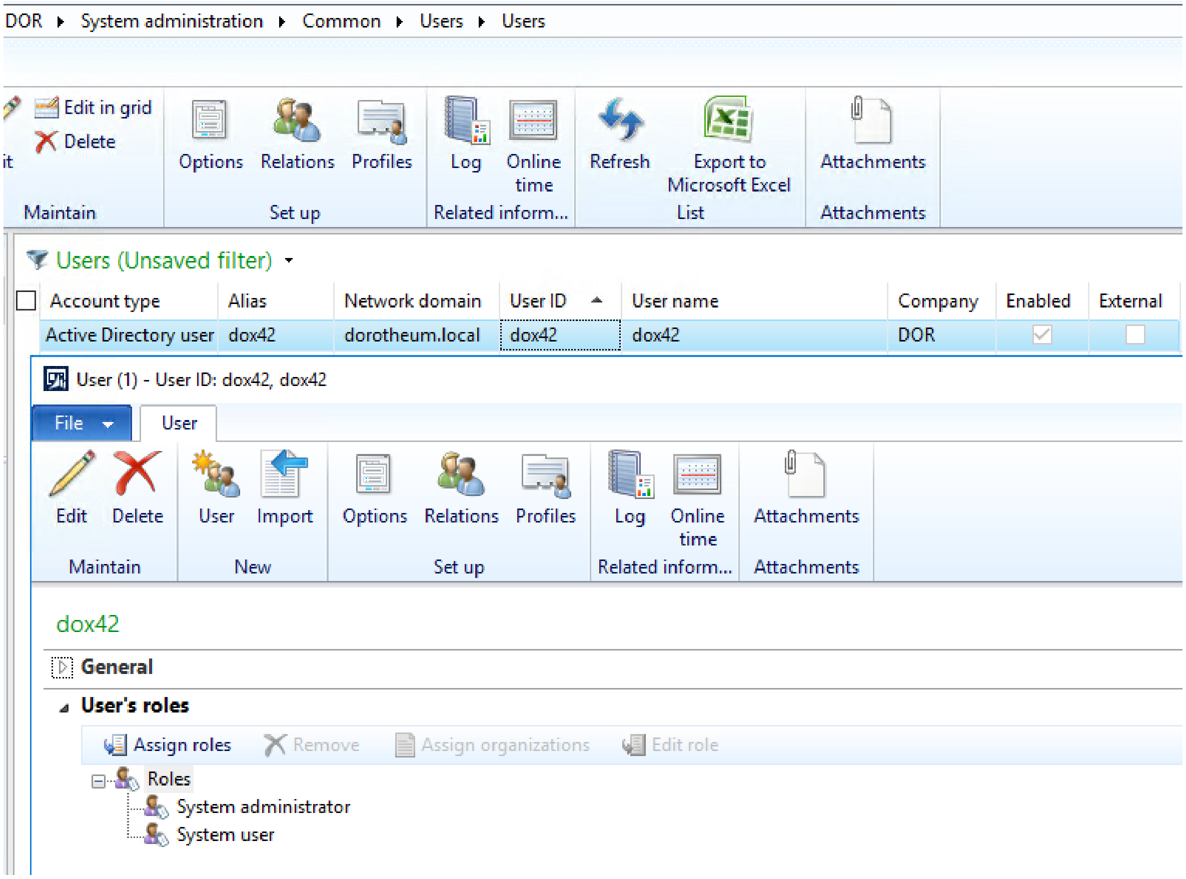
Report designer and the dox42 service user need these access rights:

- AX access
- SQL Server read access on AX database

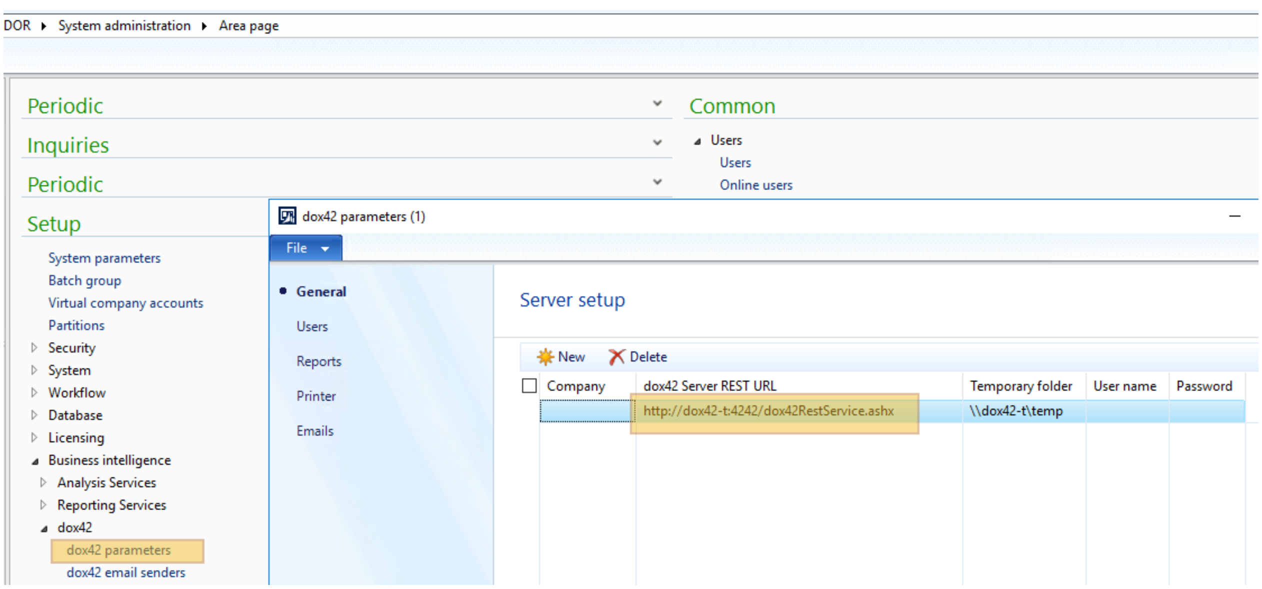


## Configuration

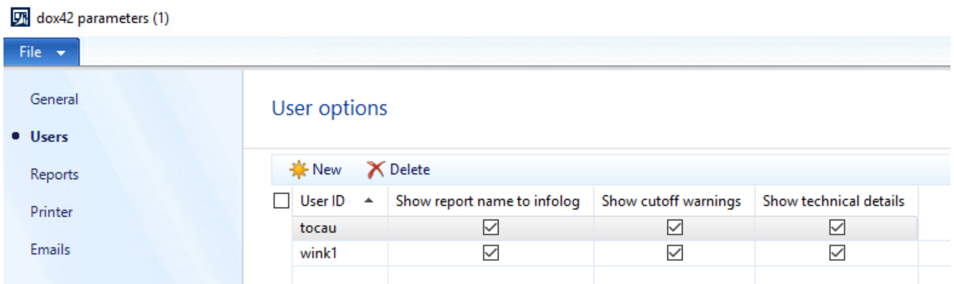
dox42 user das access rights within AX:



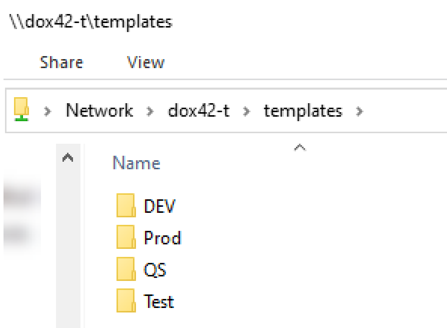

Normally, all companies have the same dox42 service:



For report designers, we show the name of the SSRS report in an infolog (so that we know the technical report name).

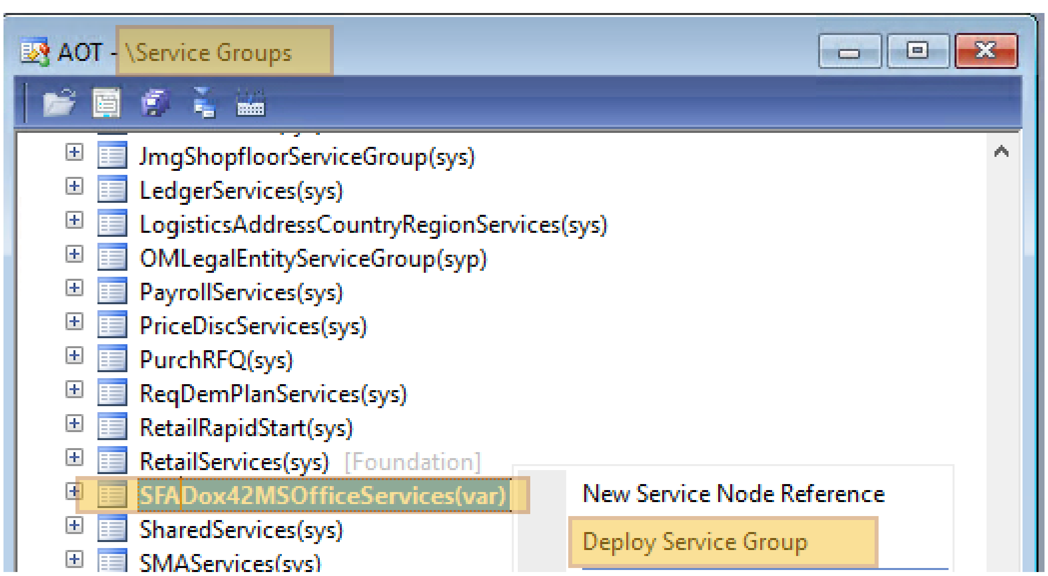



The report templates should be in folder a structure per AX system:

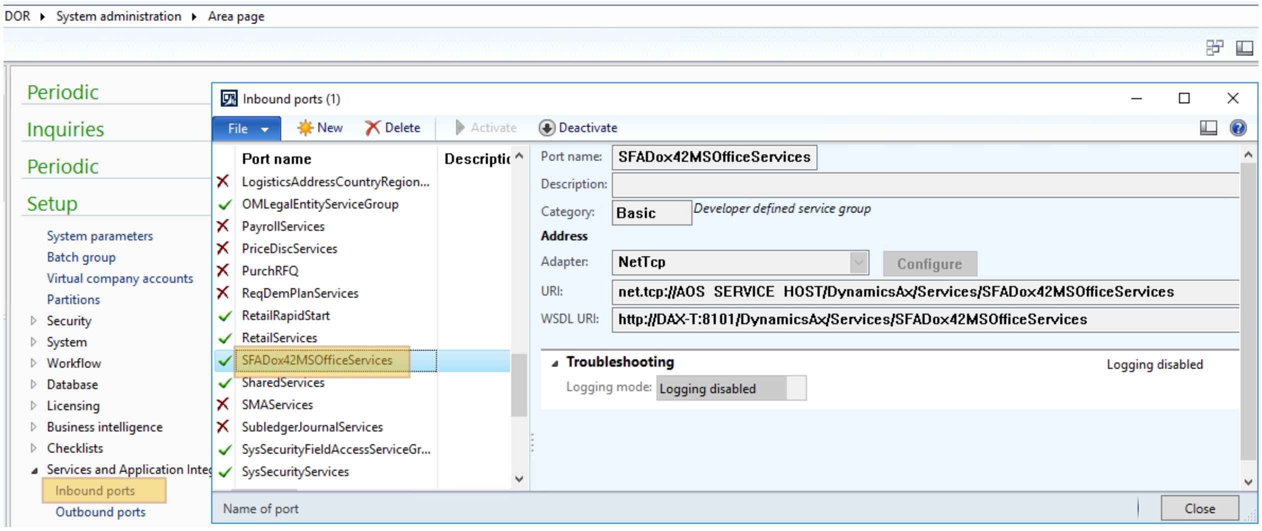



The communication between AX and dox42 should be via inbound port (normally 8201).

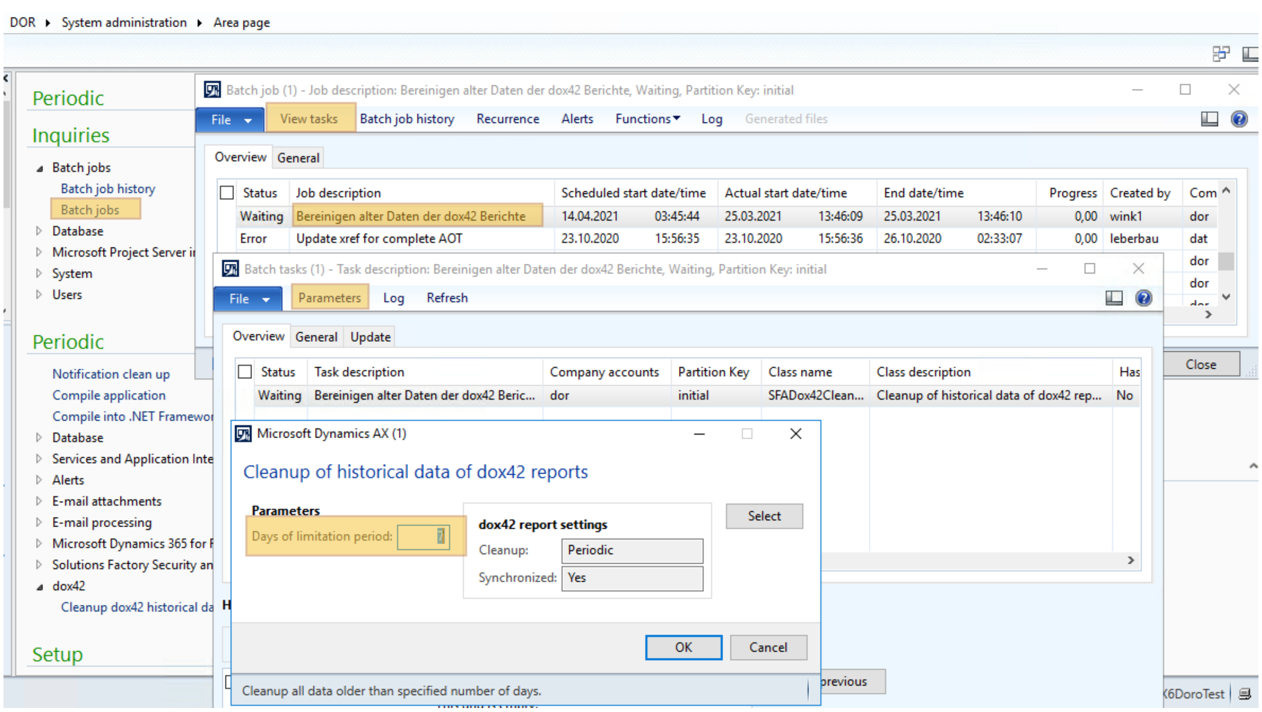
To ensure that you see the incoming port, the service group SFADox42MSOfficeServices needs to be deployed.



And the inbound port needs to be active:




A batch-job needs to be set up to delete all (old) data – e.g. anything older than 7 days:



## Templates and update processes

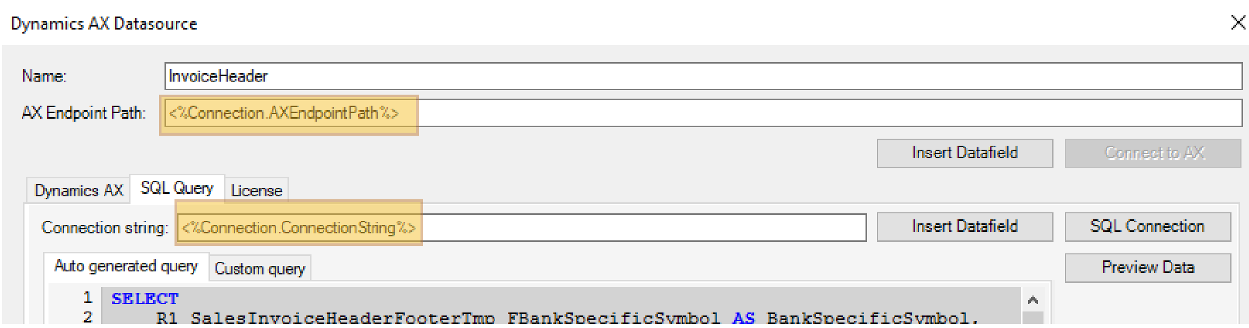
In the templates folder for each environment, keep a file Connection.xlsx:

- The reports can be developed without difference between Dev/Test/QS/Prod
- The delivery of reports from Dev -> Test -> QS -> Live without change to the report / to the Datamap of the report can simply be done via word file copy

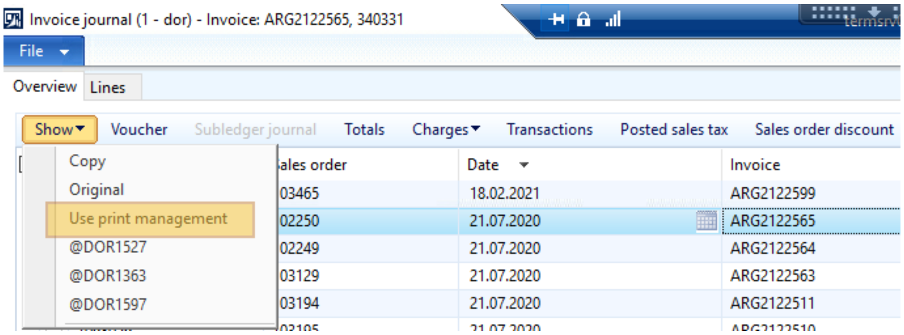
These are the settings:

- Test:
    - Cell A1: Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=AX6DBTest;Data Source=TestDBServer\TestDBServerInstance
    - Cell B1: net.tcp://testAOSserver:8201/DynamicsAx/Services/SFADox42MSOfficeService (Take care, 8201, not 8101!)

This way, these settings in the dox42 client are possible:



## Example (sales invoice)



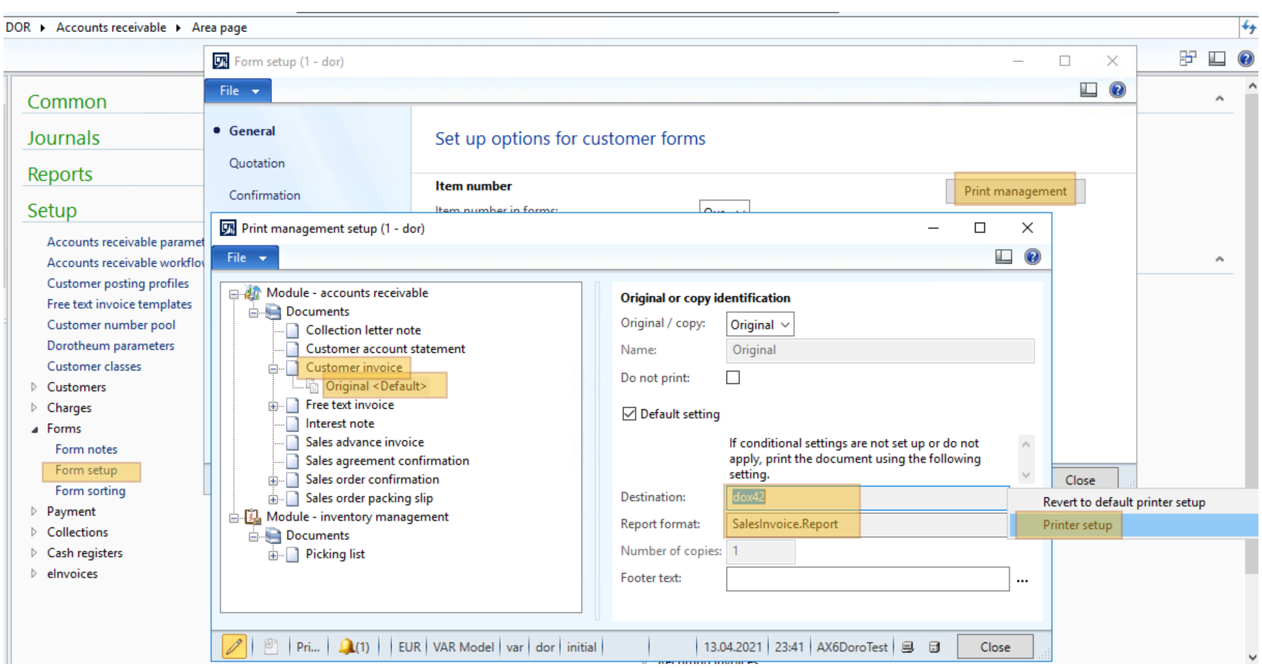

Result:

ARG2122565

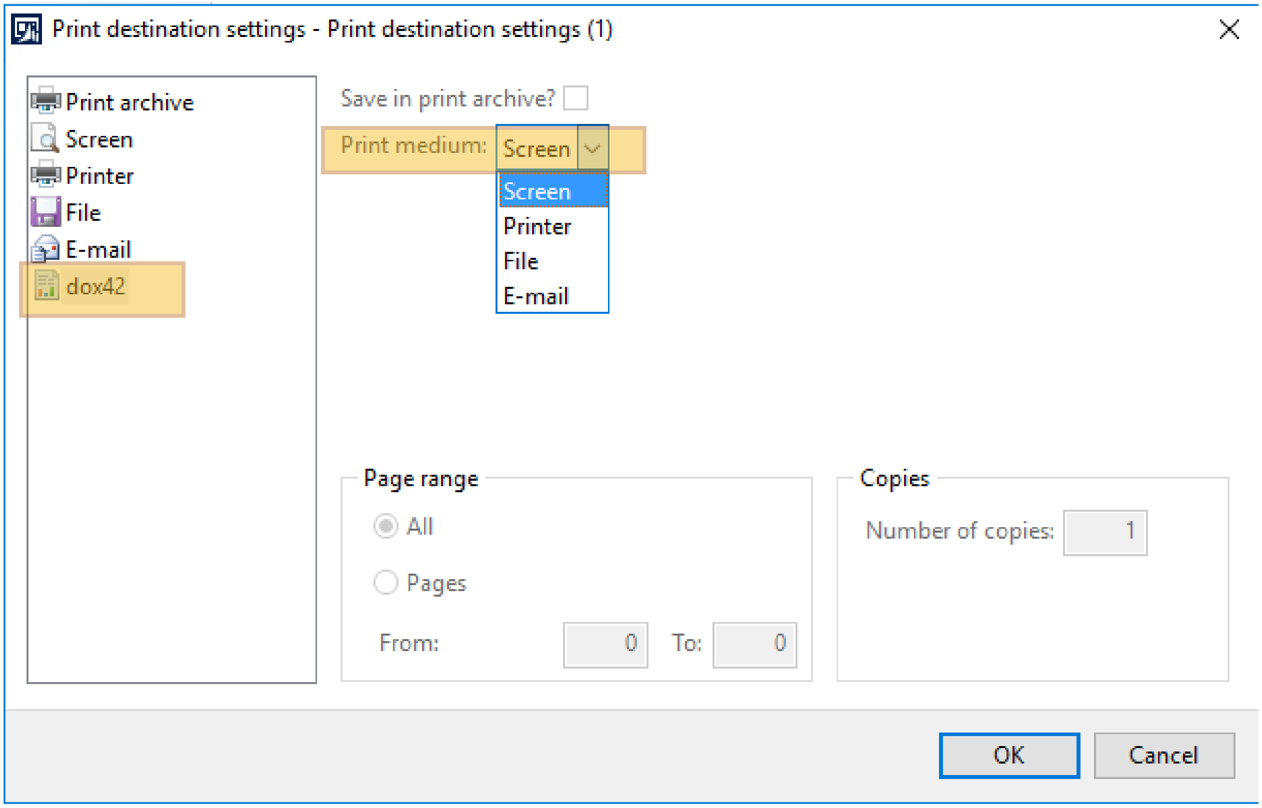
| | | | | | |
|---|---|---|---|---|---|
| 150701 | Menge: 1,00<br>Variante :<br>1277733 | 1,00 | Stk | 62,50 | EUR |

| | | | | | |
|---|---|---|---|---|---|
| | | | AT-U20 | 12,50 | EUR |

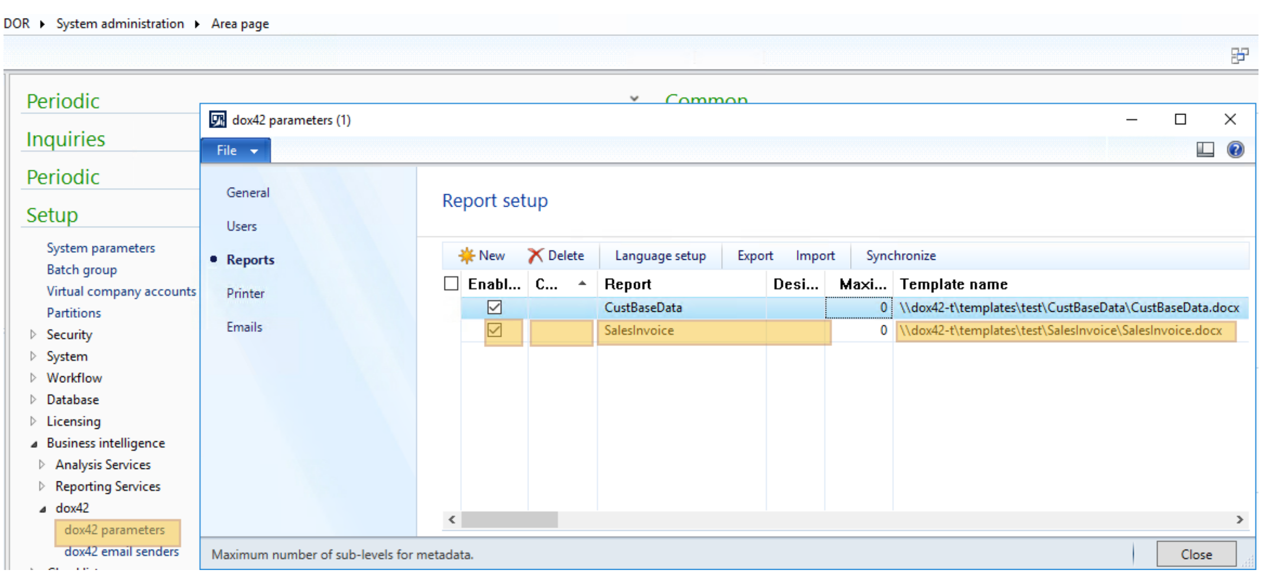
To print a dox42 report, a print dialog is needed. In case of so-called form letter reports, these settings normally come from the Print management.



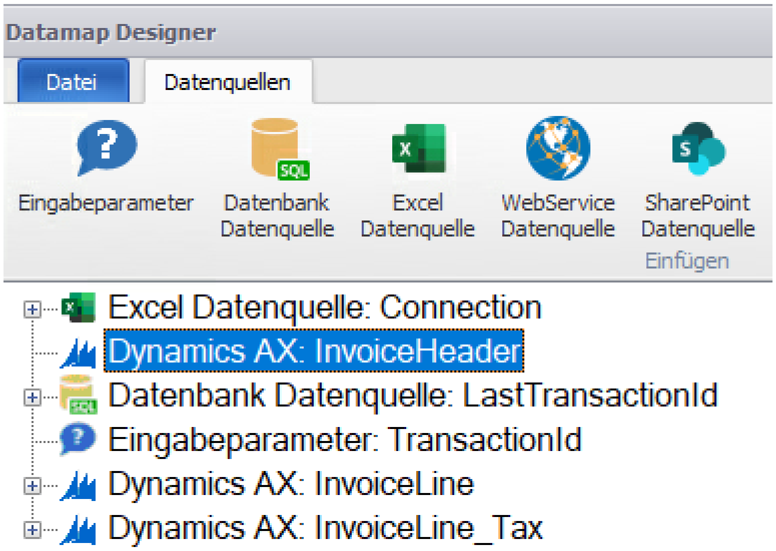
The dox42 option then has targets Screen, eMail, Printer, Fileshare.




The dox42 option is available, if the report (actually, the report design) has been set up in the dox42 parameters for the specific company (or, for all companies), and activated:

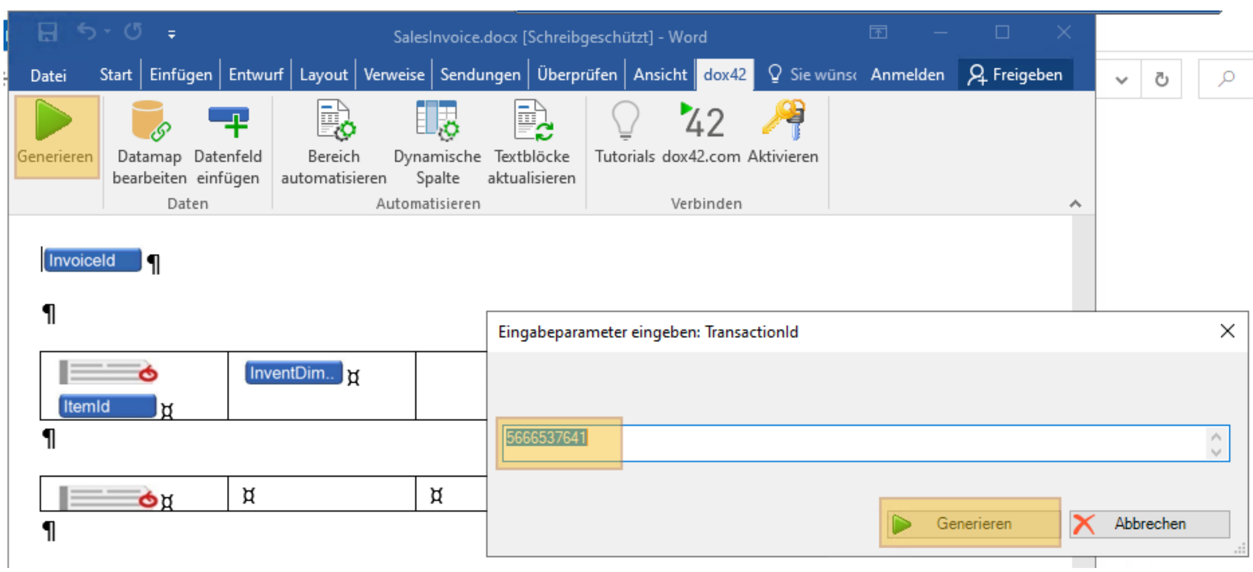


In our example the template uses this data map:




This data map allows:

1. Preview of the last-printed data via Generate




2. Simple copy of the template from Test to Prod:

    a. Same Setup in AX dox42 parameters in test and prod (only use \prod\ path instead of \test\ path):

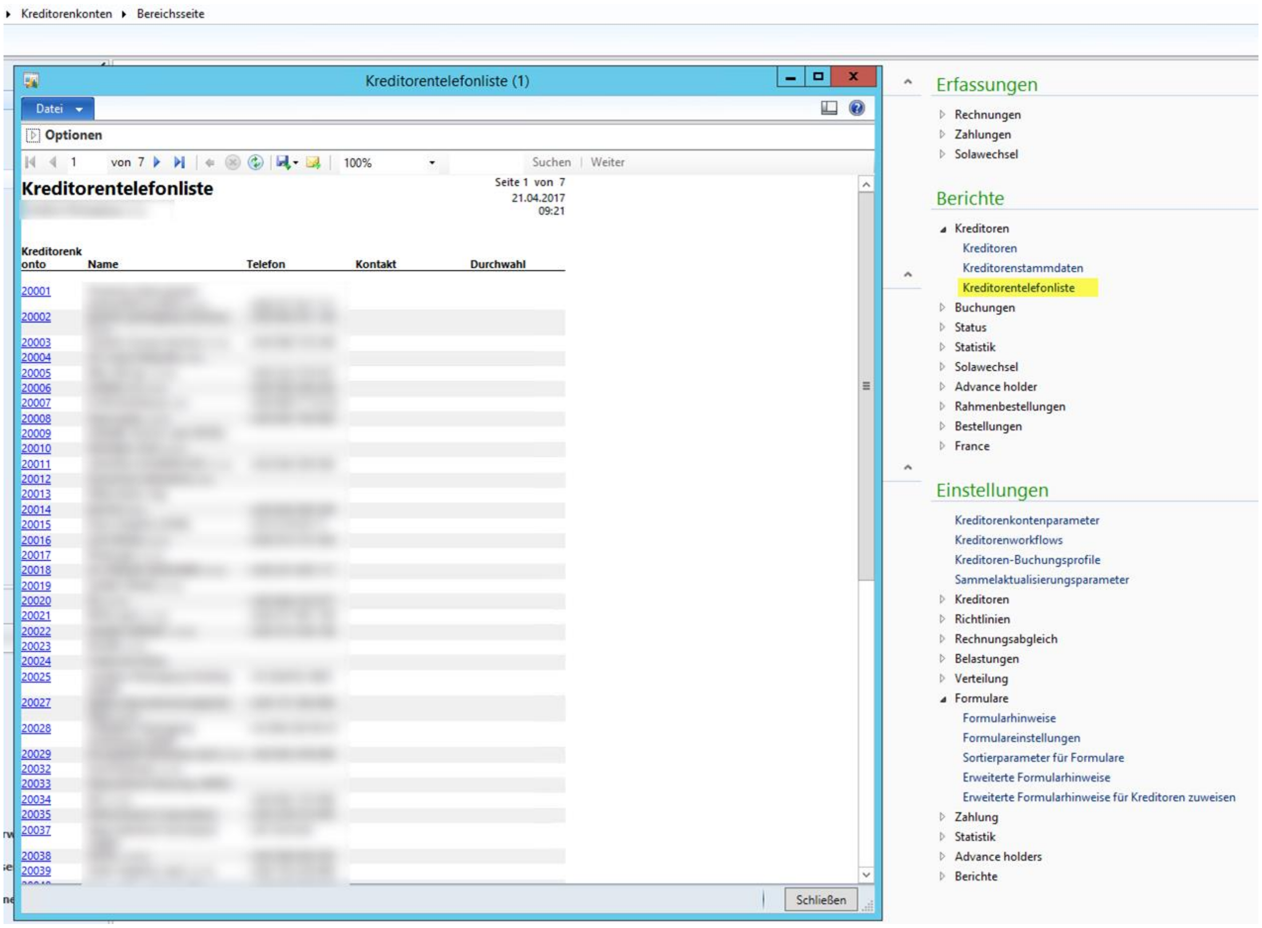
    b. Copy SalesInvoice folder from Test to Prod:

    The Connection.xlsx file, which is available in each base folder (Dev, Test, QS, Prod) then ensures access to the correct AX and to the correct database.

# Process for replacing an SSRS report with dox42

Assumption, we want to replace the report Vendor accounts -> reports -> vendors -> vendor phoner list.

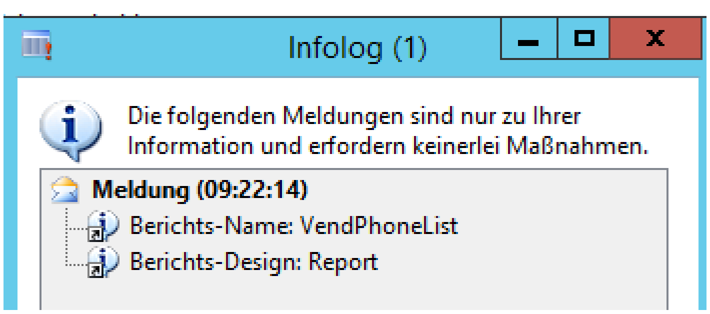
Current report design:




1. First task – Identify technical name of report.

This is needed as we need to know which report we are actually replacing.

To do this, simply open the existing SSRS report – an infolog is displayed with the technical report name and the report design:



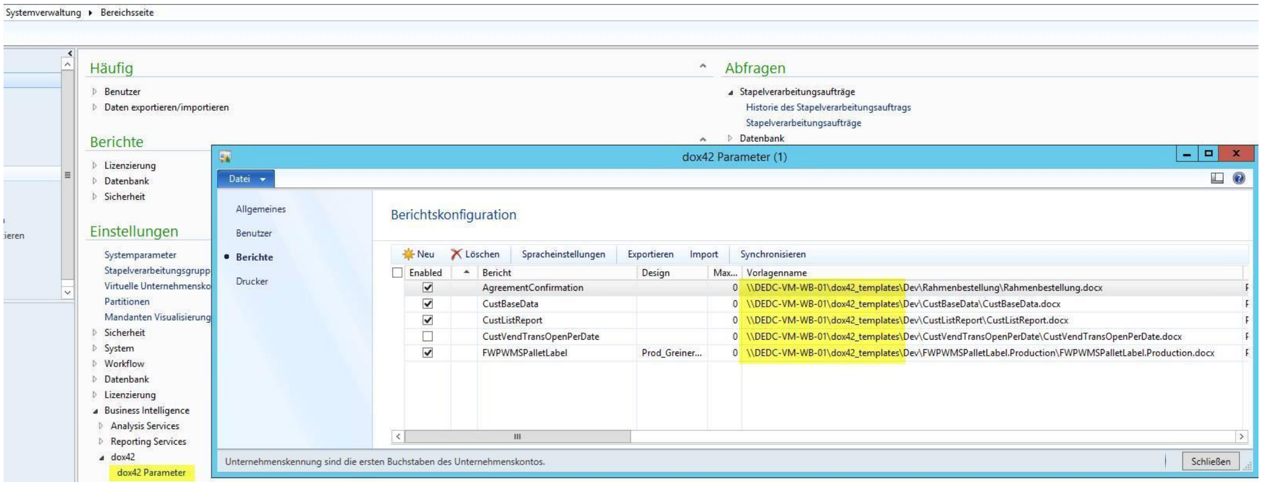

(Most SSRS reports have only 1 report design.)

2. Second task – Create new dox42 Word report template

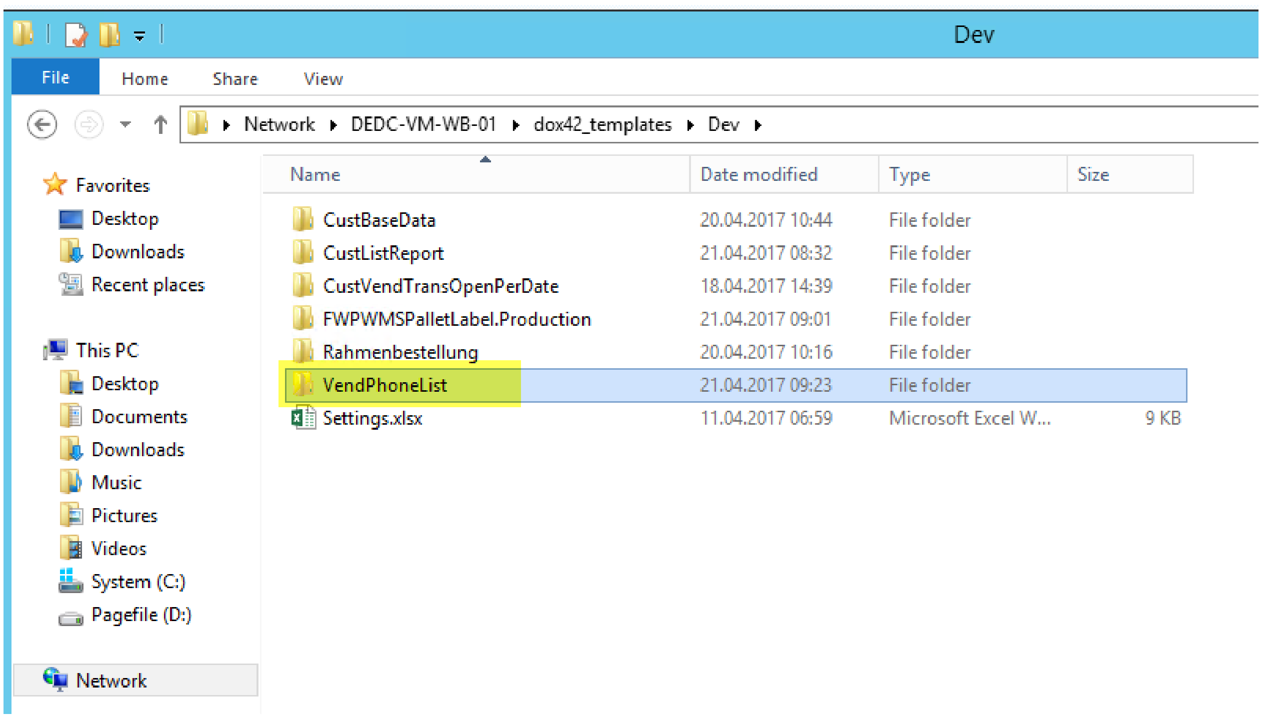
Either copy existing template and change settings, or start from scratch.

Here, we start from scratch:

a. Template folder is in the dox42 parameters:

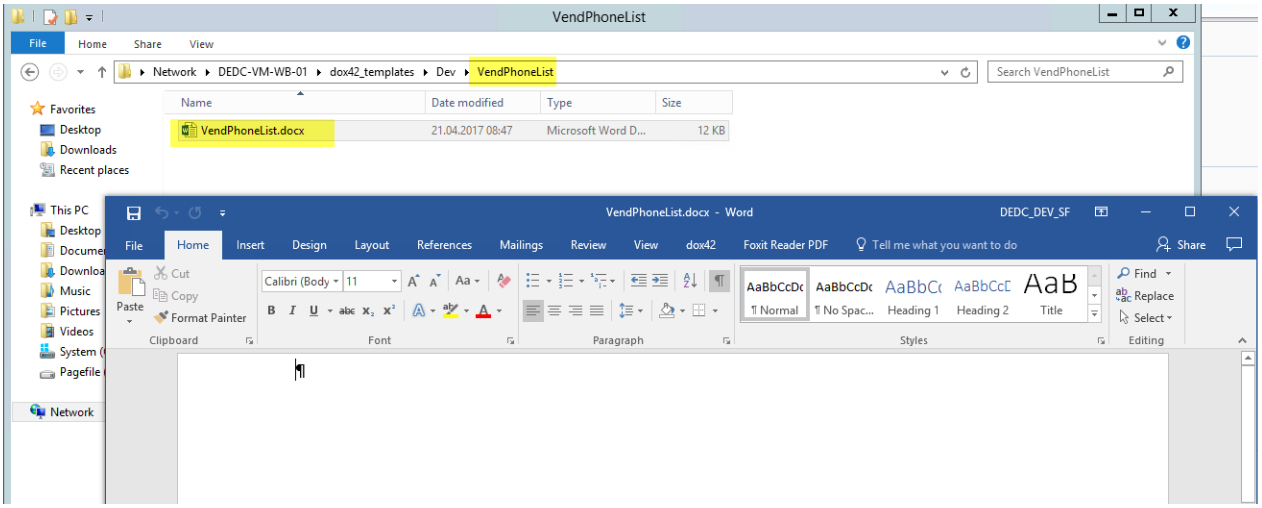



b. Create new subfolder in this template folder, e.g. with technical name of the report:

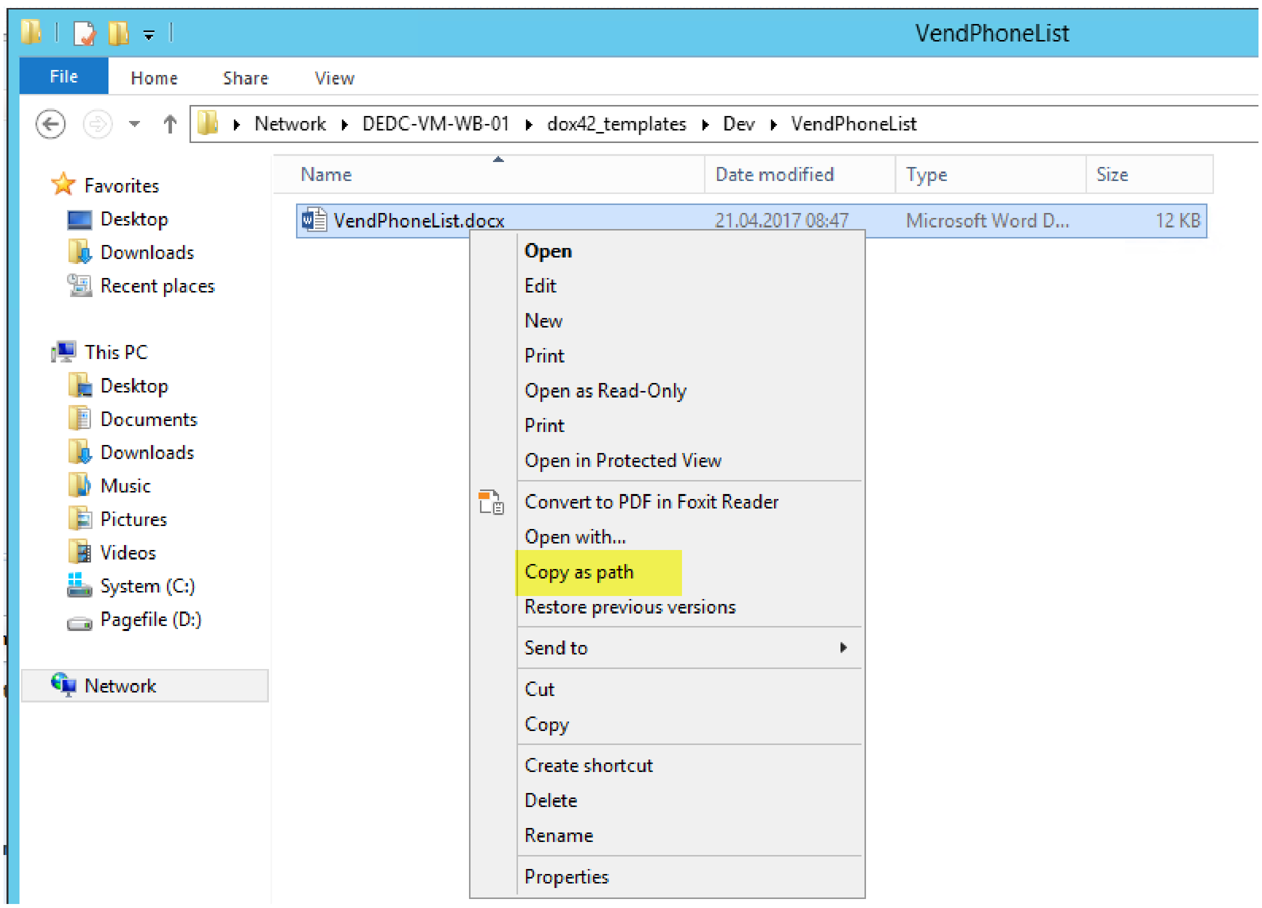



c. Create new (empty) Word document in this subfolder, ideally the template name matches the technical report name

Result: New Word template in new folder




d. Next step: Take file name incl. path into clipboard (Shift-Rechtsklick->“copy as path“)



3. Create new line in the dox42 parameters, using the path to the Word file – we are replacing the SSRS report by our Word template

Important settings:

   a. Enabled: needs to be set – this way, our new, emply, not-yet-working report template is activated

   b. Company: If empty, the SSRS report is replaced by the dox42 report in all companies

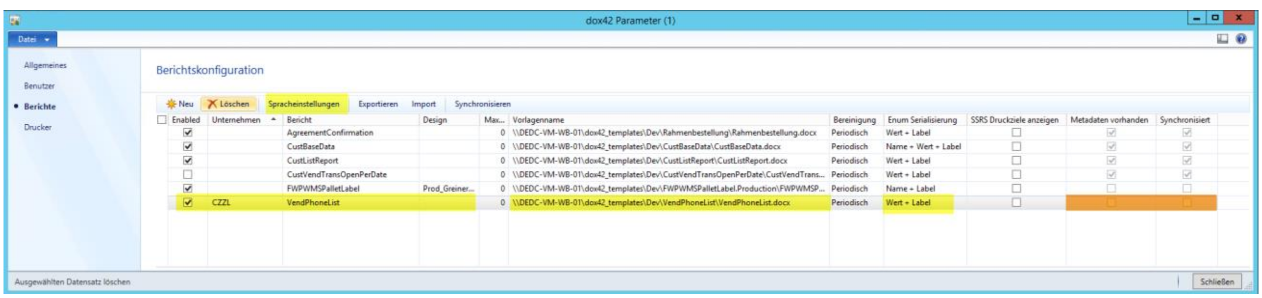
   c. Report: Technical report name

   d. Design: If empty, all designs of the SSRS report are replaced by the dox42 report. In most cases, there is only 1 report design, and even if there are more, many times all report designs may be replaced by the same dox42 template.

   e. Template name: This is the path+document which we still have in our clipboard (see step above)

   f. Enum Serialization: please use Value+Label or Name+Value+Label. This way, Enums like SalesStatus are shown in the report data in easy-to-understand form (e.g. status of sales order – SalesStatus 2 has Name „Delivered“, Label „Geliefert“). Within the template, we recommend to work with the enum name.

   g. Language settings: dox42 report templates are not multi-language by default – instead, we suggest to duplicate the report design if several languages are needed (it is also possible to work with a labels Excel, but this way, the WYSIWYG of the Word template suffers)

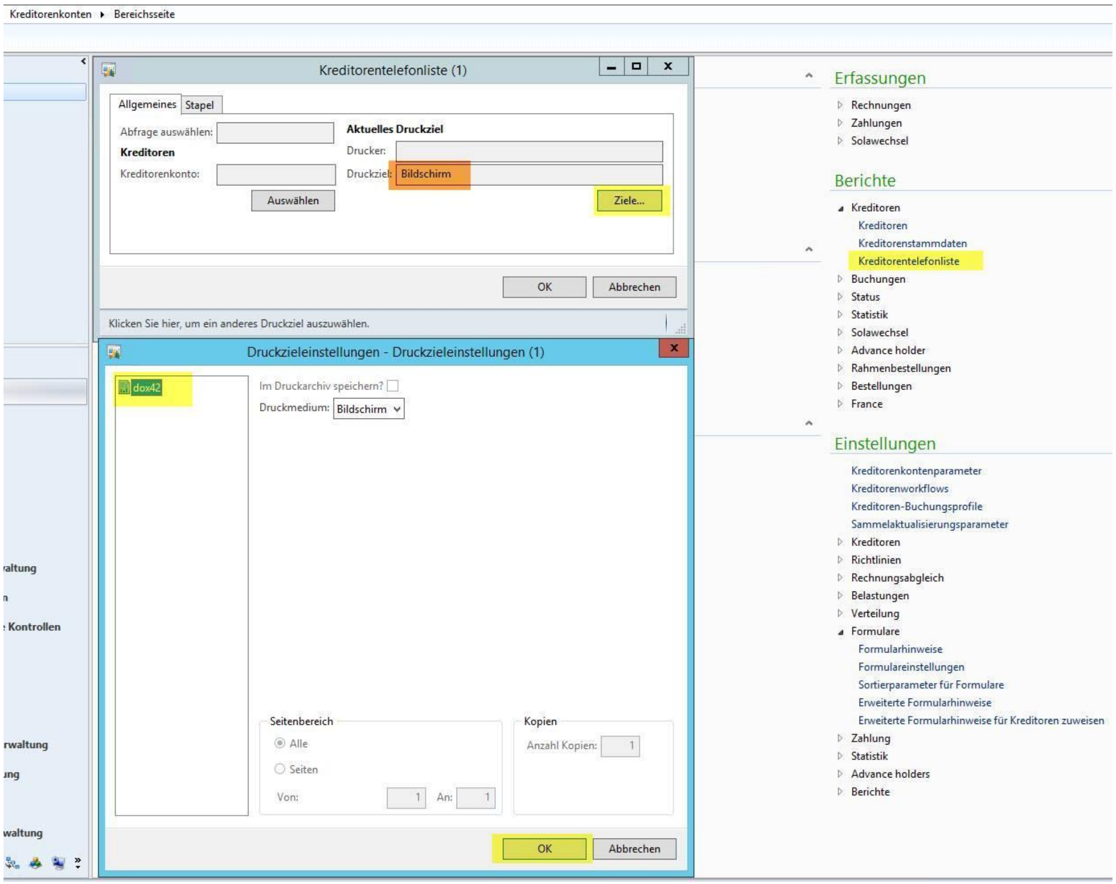


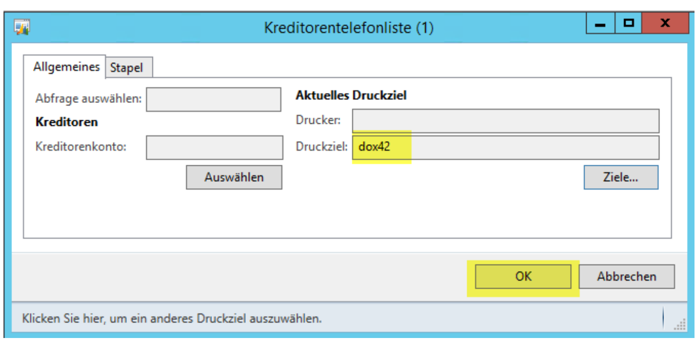

Very important are the checkboxes “metadata available” and “synchronized” – these, we set in the following steps.

4. Create the report metadata (dox42 needs to know which data sources and fields are behind the report).

We are done if the checkbox „Metadaten available“ in the dox42 parameters is active.

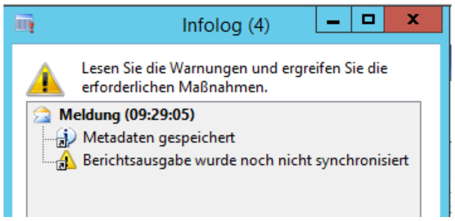
   a. To achieve this, simply print the report again, this time, with the new print target dox42:



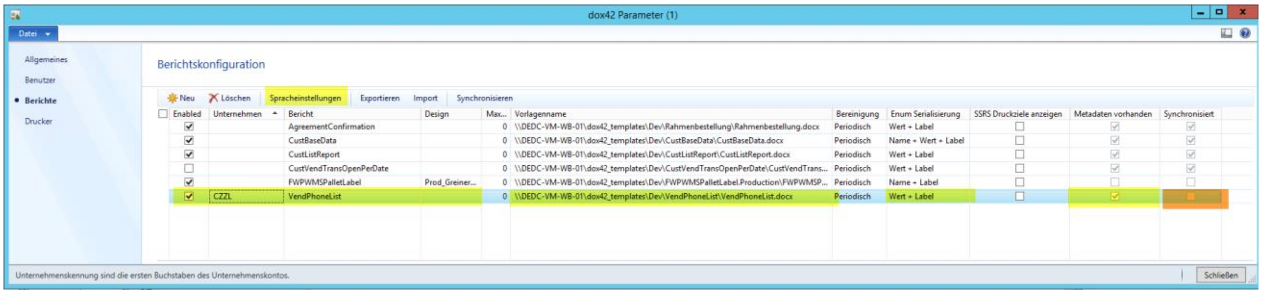




After the OK, we get these messages in the infolog

- Metadata saved – OK
- Report output not yet synchronized (this is our next step)

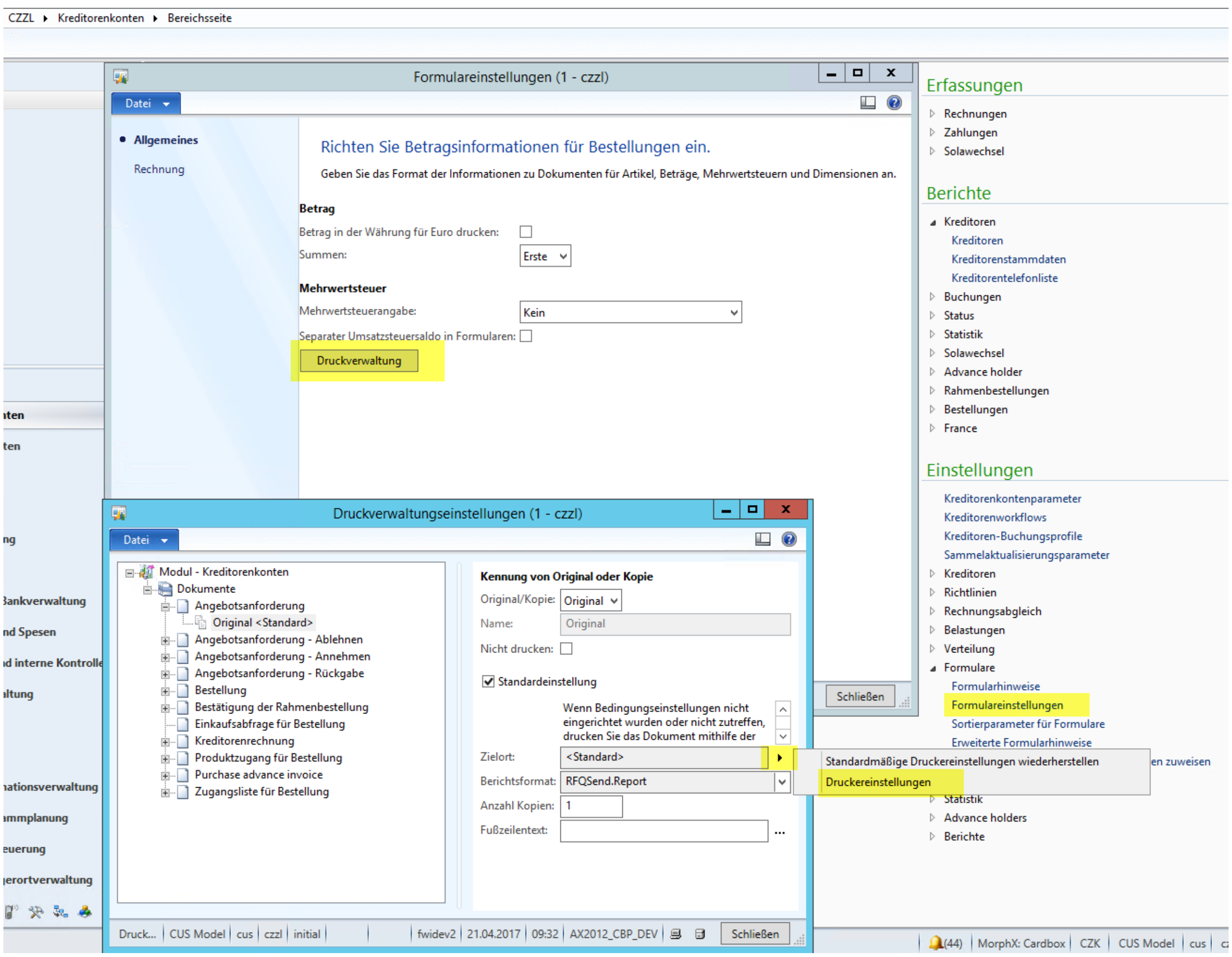



Now, checkbox „Metadata available" is active in the parameters:



In several (relevant) special cases, AX does not ask for a print target – then, we cannot chose the print target dox42 instead of standard (SSRS) print target screen.

In many of these cases, the print target is selected by the settings in the form setup (print management):

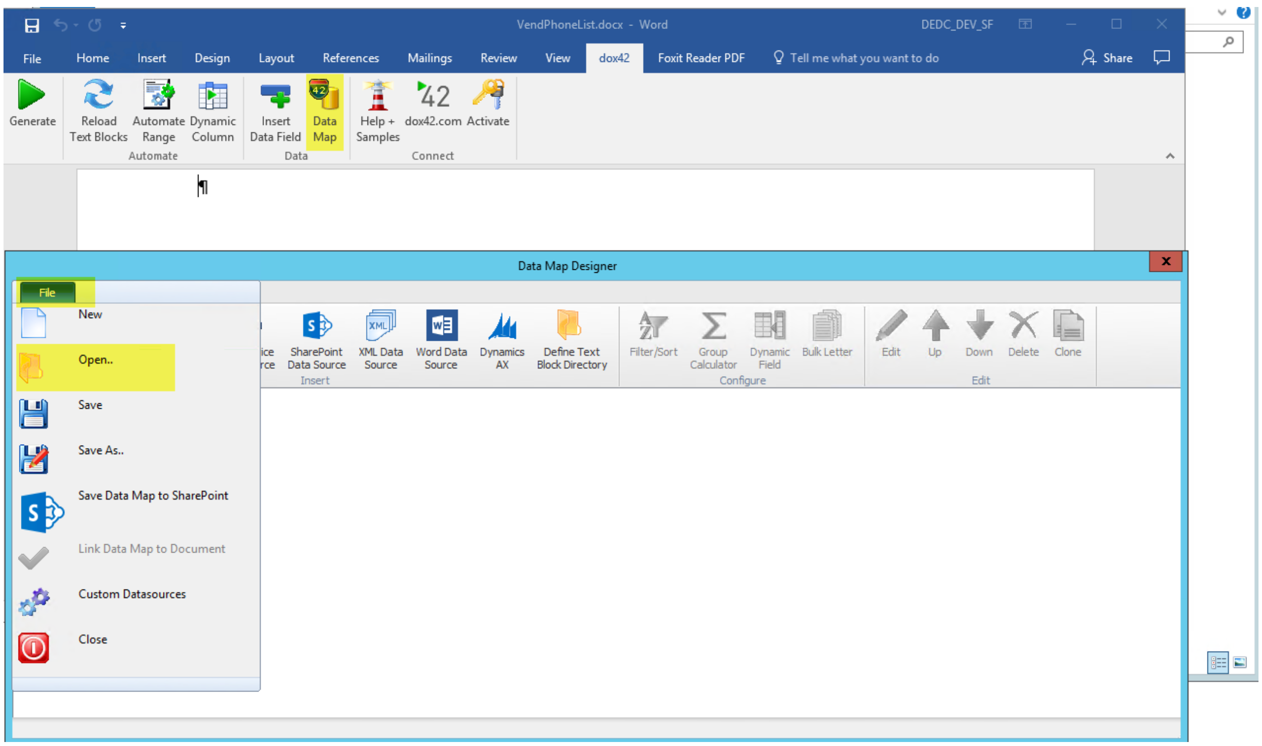


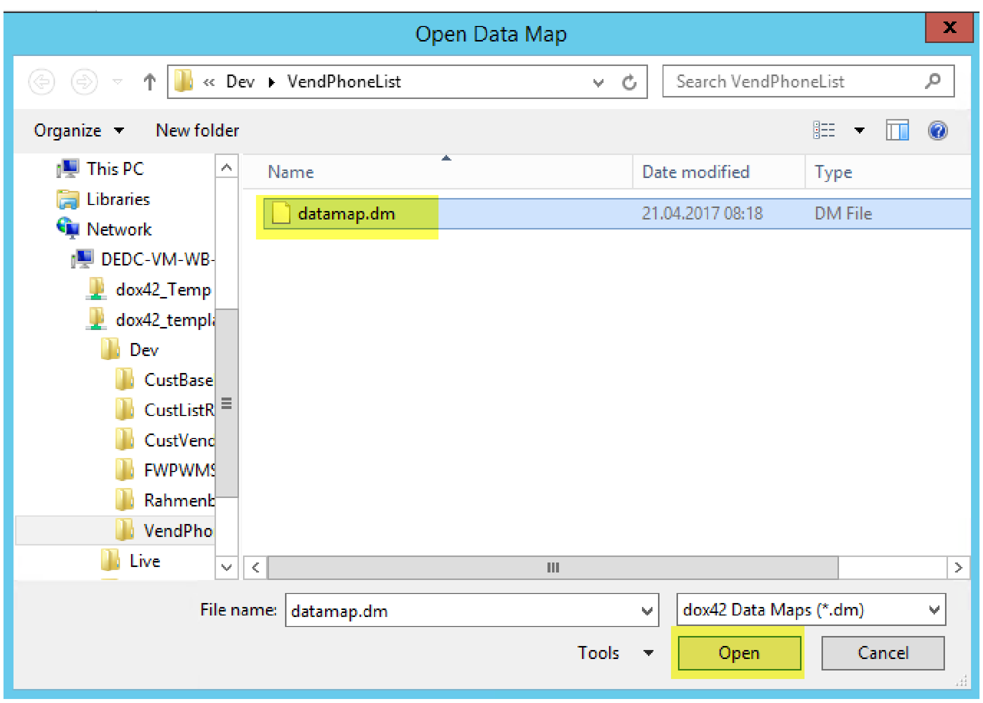
5. Next, we need to synchronize, so that dox42 can create temporary dox42 tables for the report data.

To achieve this, we need to update the data map in dox42, and create a minimal report template.

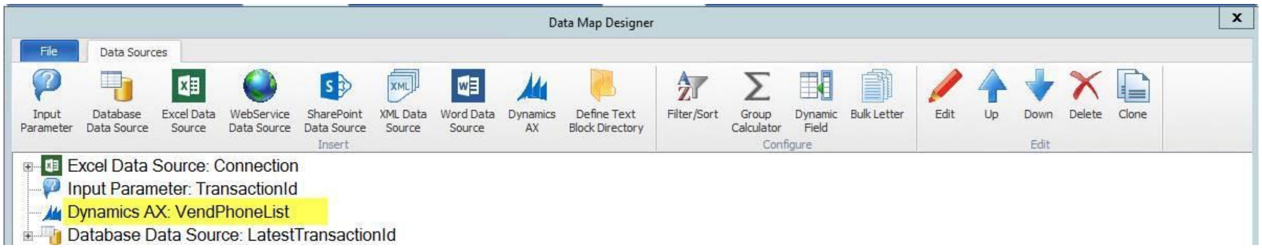
Generate a new report data map like this (or copy an existing one and change it later):

   a. Open existing / create new data map in Word template:



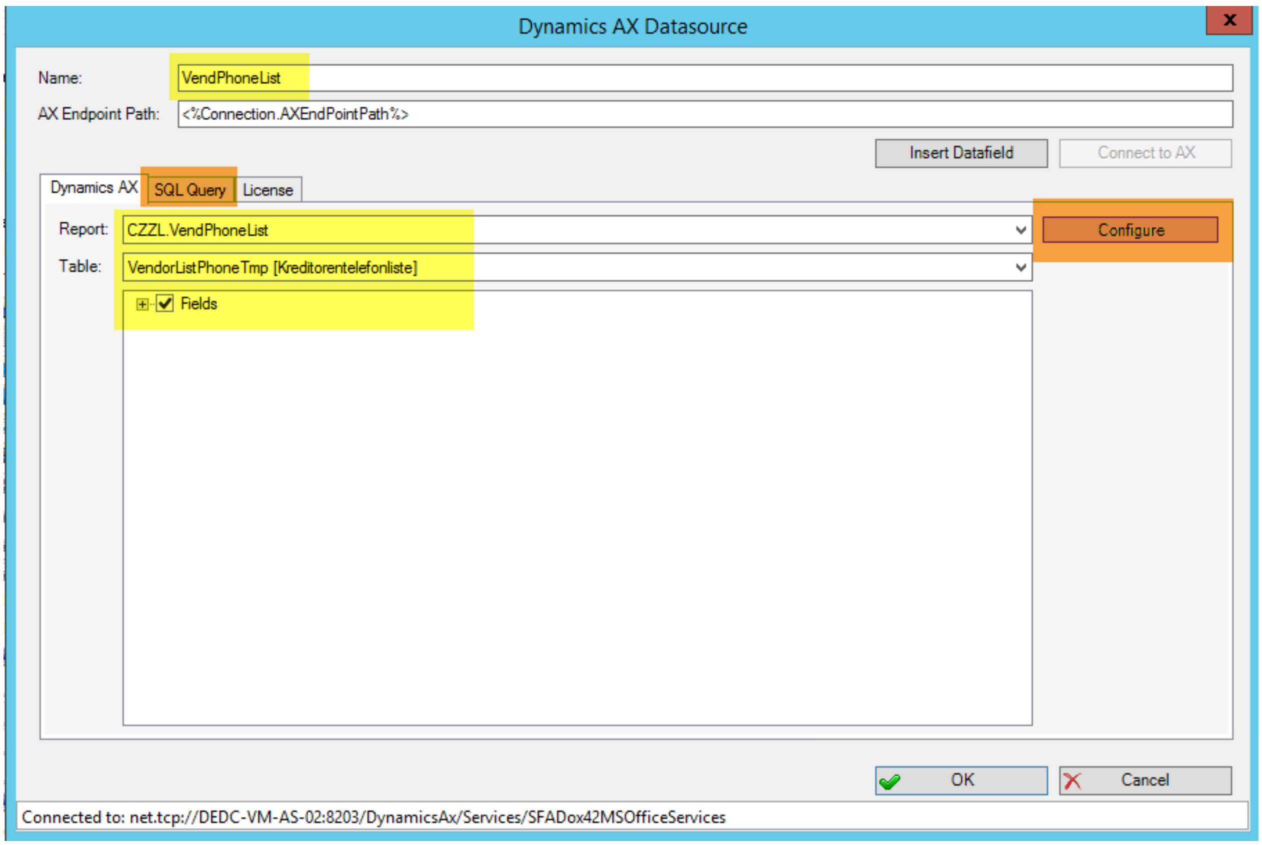




Result should look like this:



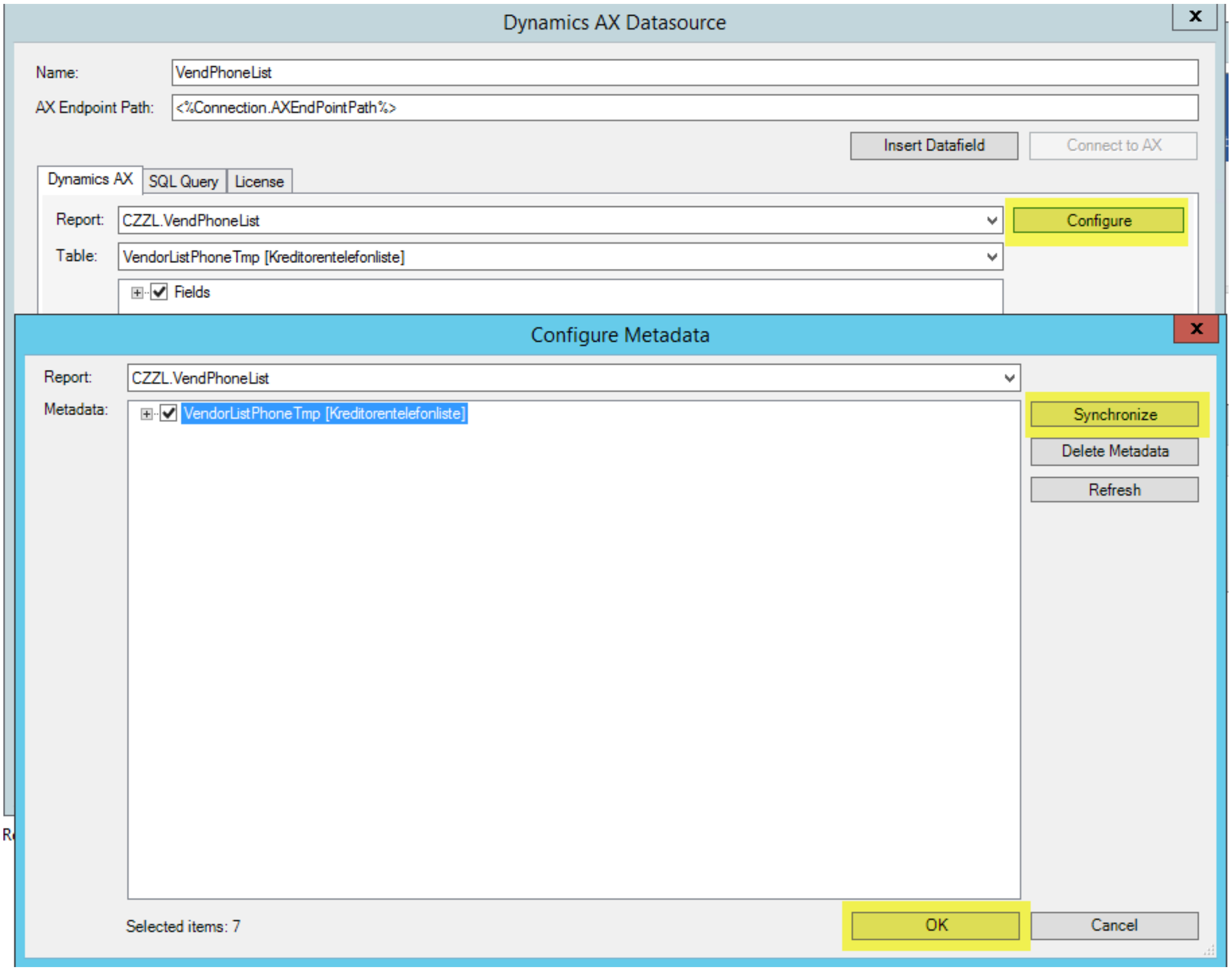

These should be the settings:

b. Dynamics AX data source

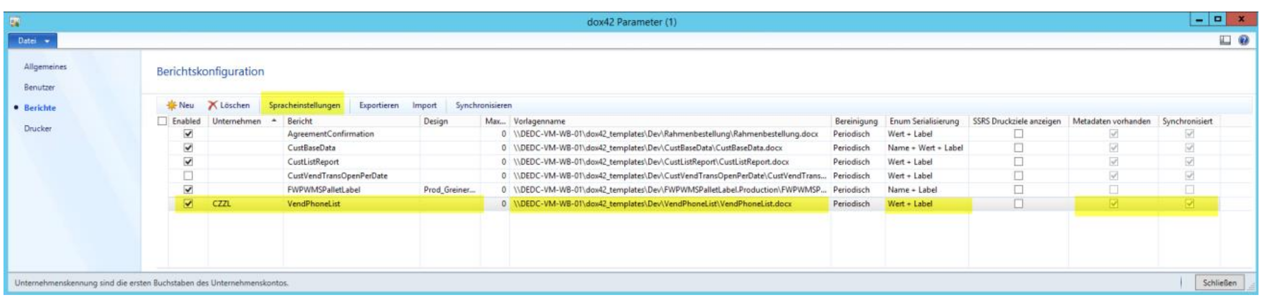



- Report: In the Report-Dropdown, we can select our new report (CZZL.VendPhoneList). If the report is missing ->the metadata has not been synchronized in the environment behind the endpoint path.
- Table: Select the table with the data (we need one dox42 Dynamics AX datasource per report table)
- Fields: Set all checkboxes
- Name: VendPhoneList (Name of the datasource)

In Configure -> click Synchronize



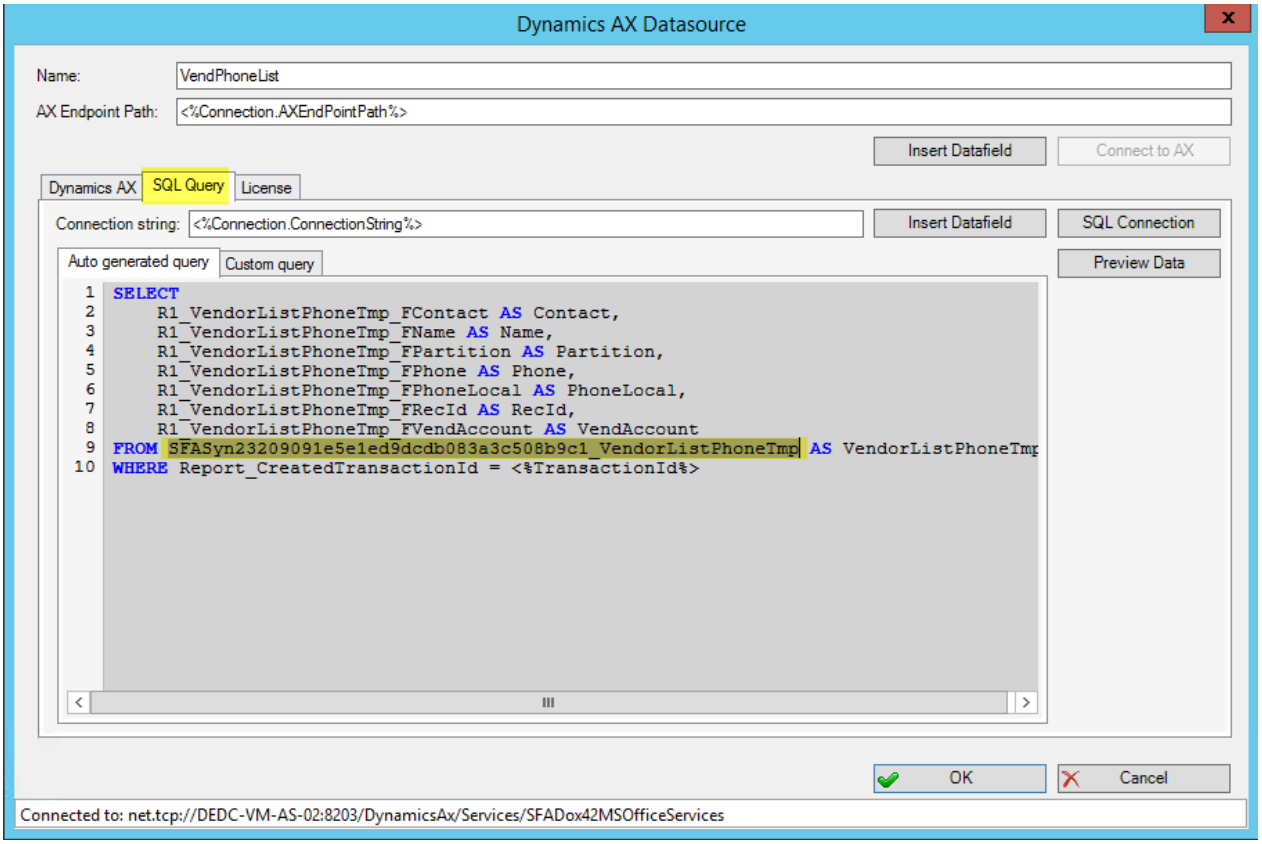

In the dox42 parameters, checkbox „Synchronized“ is now active:



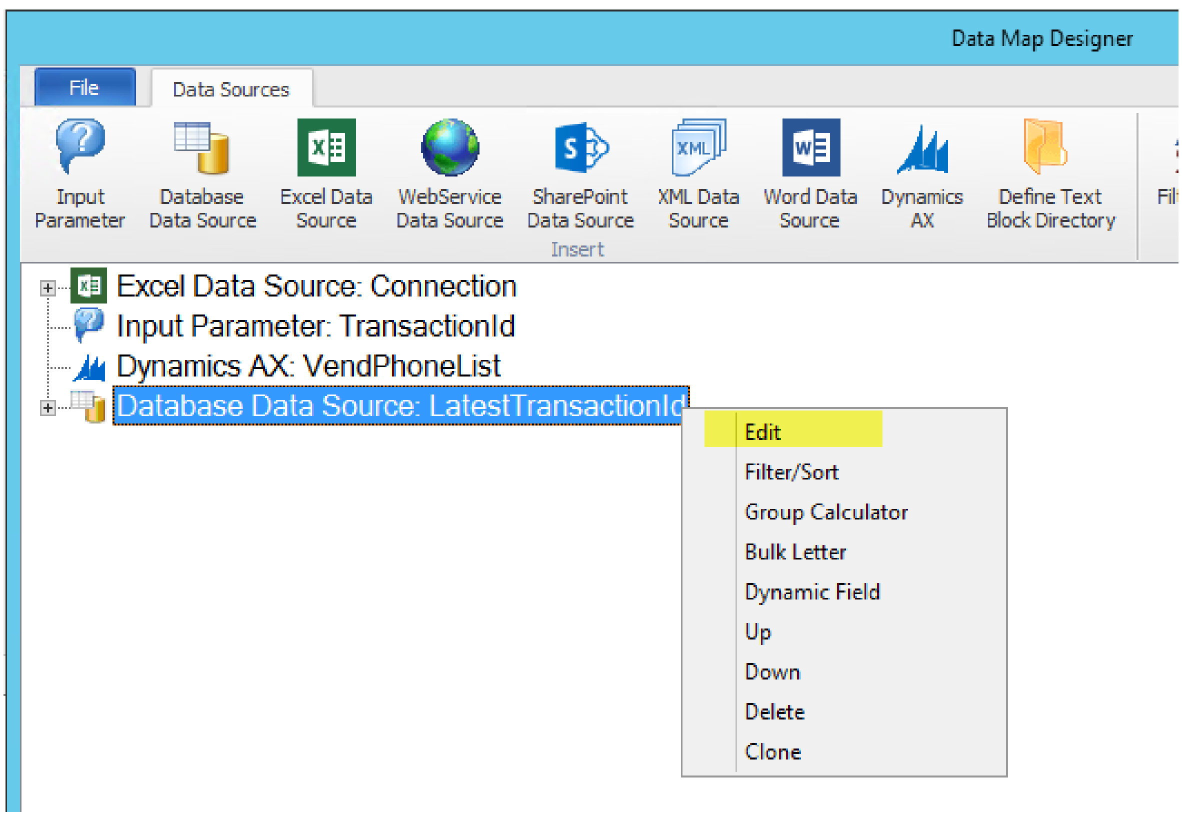

Now a temporary table has been created in the database.

c. Identify name of temporary table

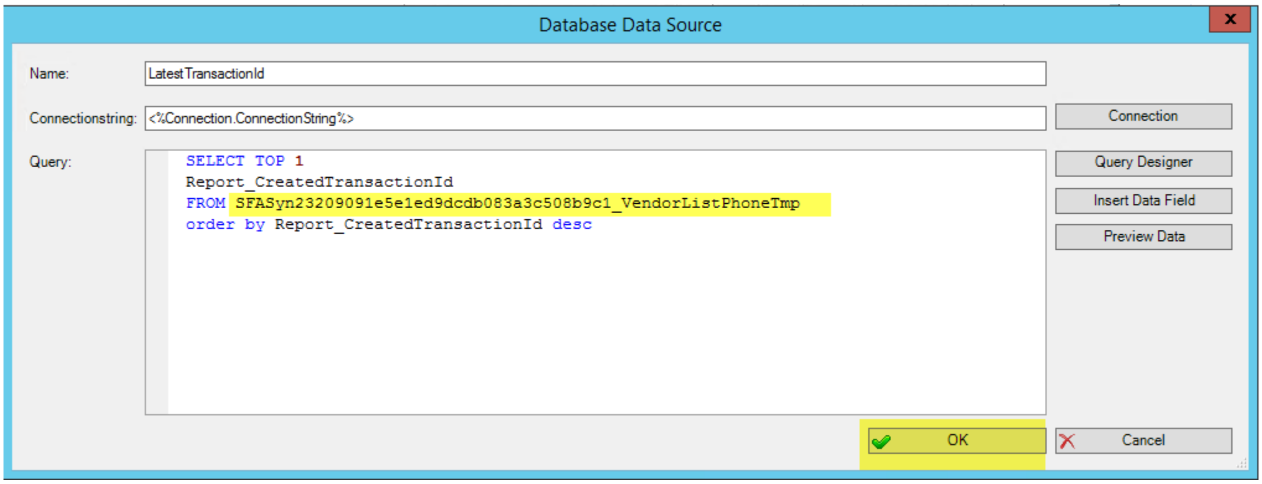
The name of the name of temporary table can be seen in „SQL Query“ tab, after the „FROM“. Copy that to the clipboard, for the main data table of your report:




d. Change database datasource „LatestTransactionId“:

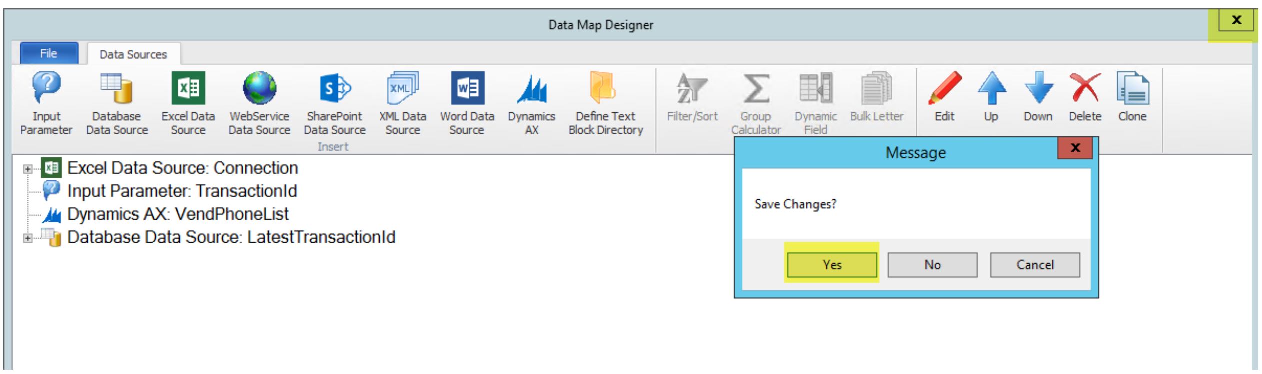


It should look like this:

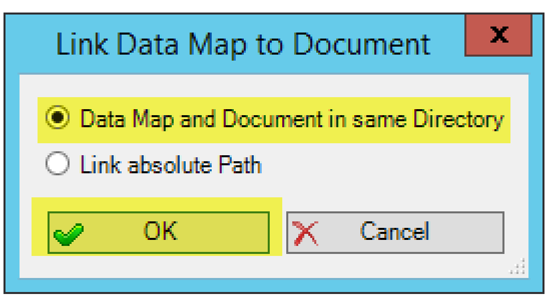



(actually, it should have an order-by-statement like “order by Report_CreatedDateTime desc, Report_CreatedTransactonId”, so that it works correctly in multi-AOS environments)

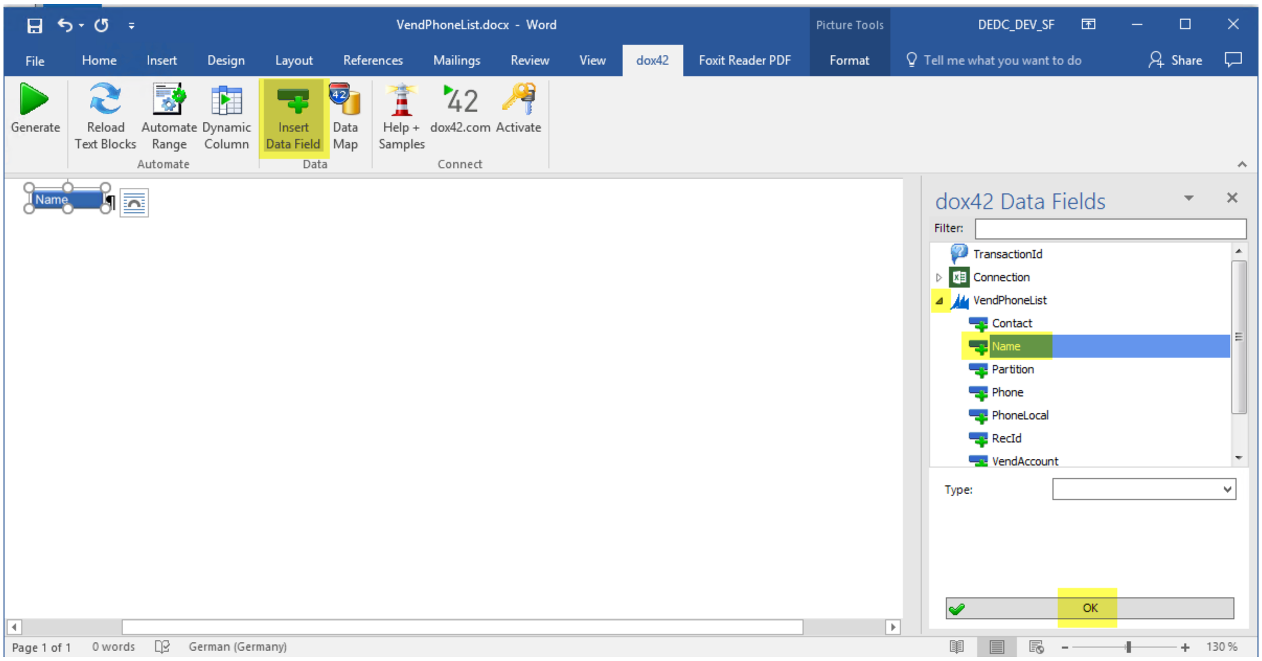
Close data map, save changes:



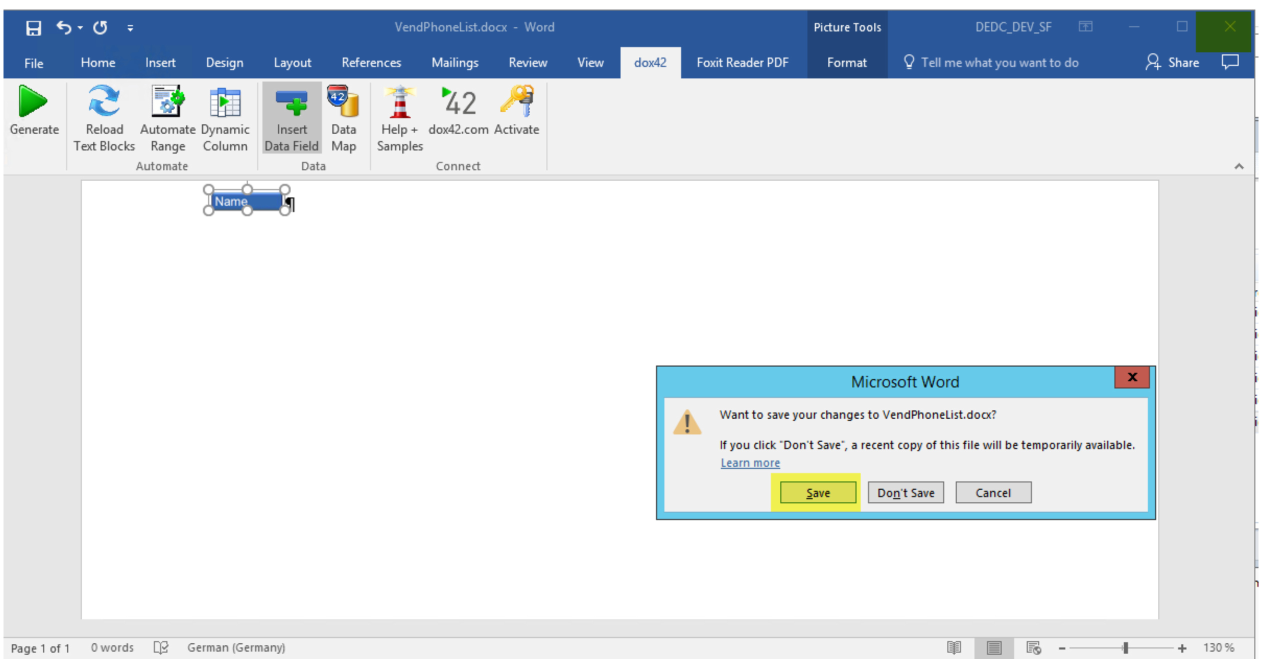

If you get this question, simply confirm:



e. For our example, we need some fields from the main report datasource:



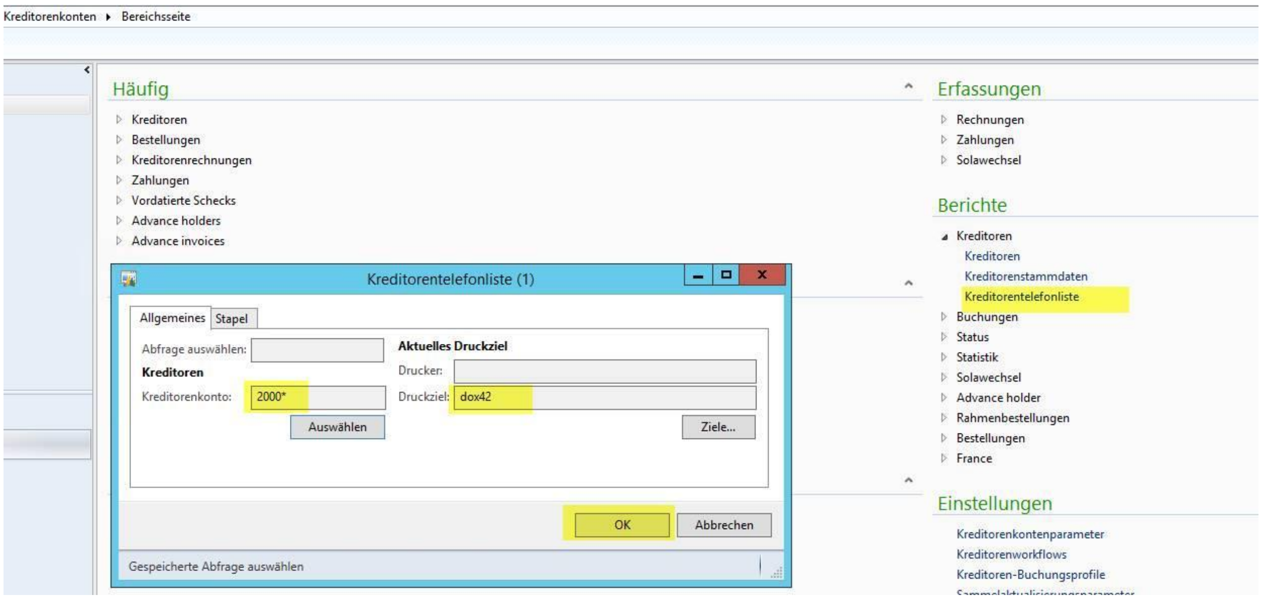

Save the template, close the template in Word (maybe close Word):



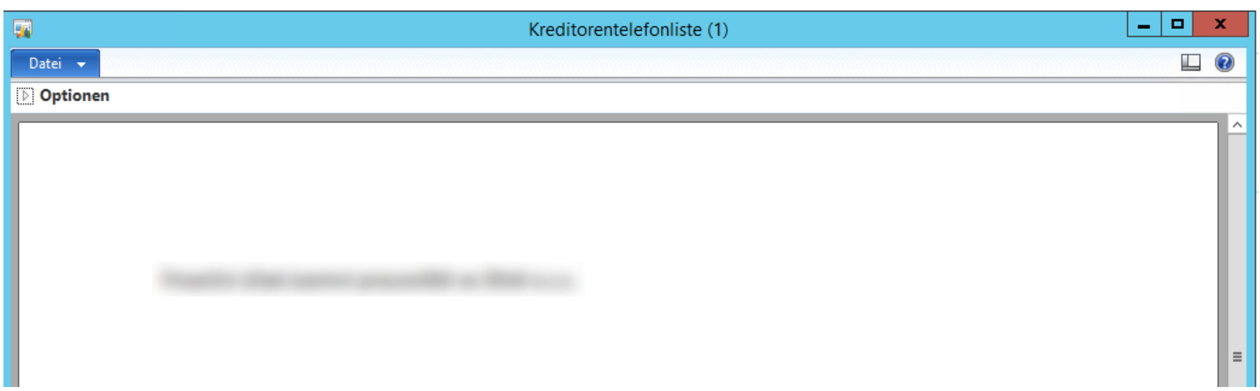
6. Call new report from AX:

We want to see dummy template data, and we want to see the content of the temporary table, so that we then can analyze the report data we get from AX, in dox42, with buttons Preview-Data and Generate

a. Call report in AX

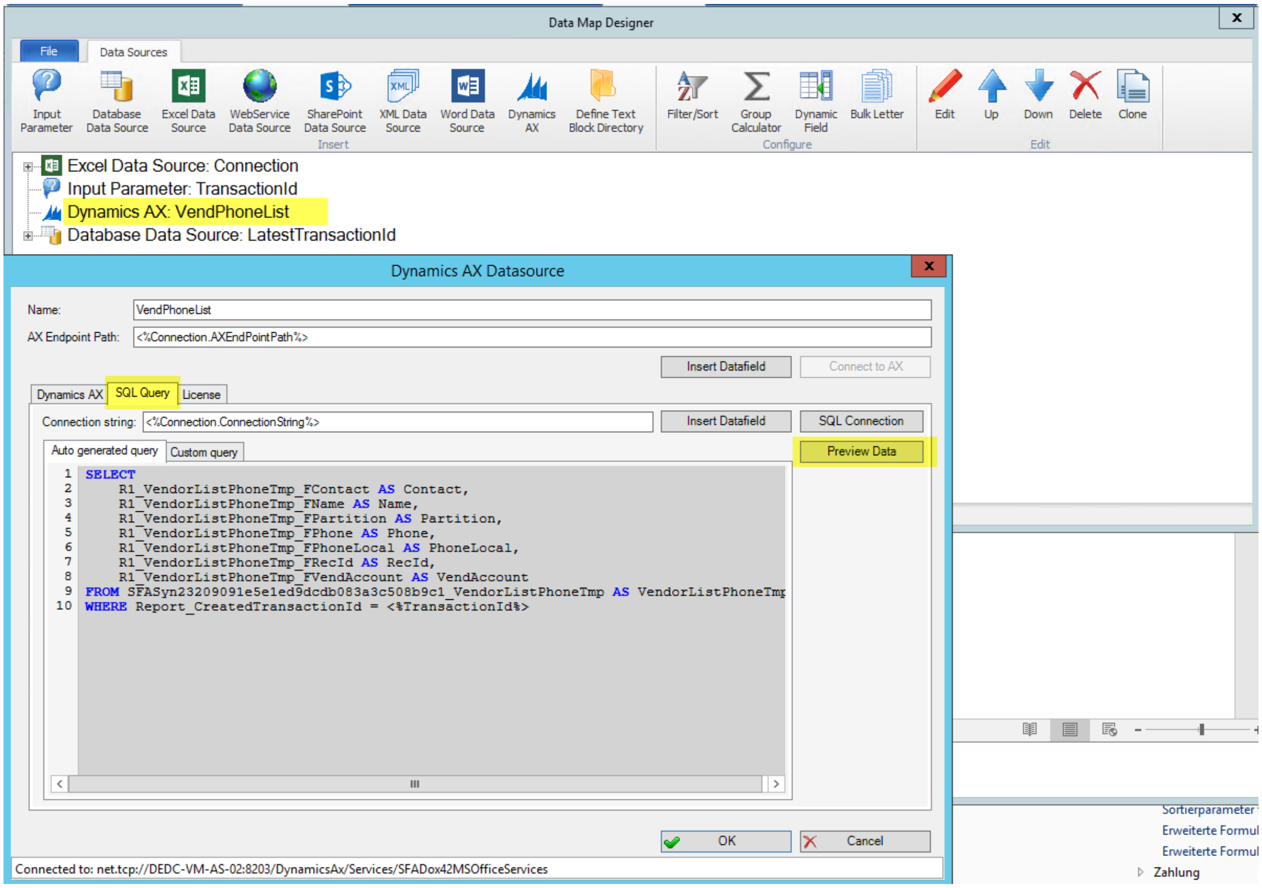



We get this result:

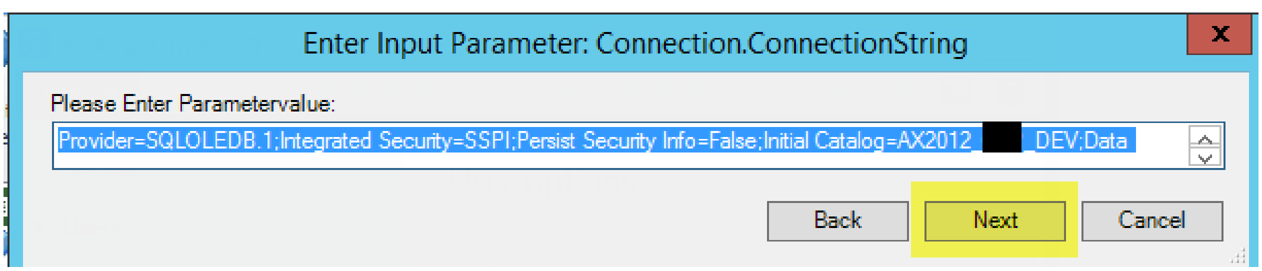


7. From now on, we have this possibility in the dox42 client:

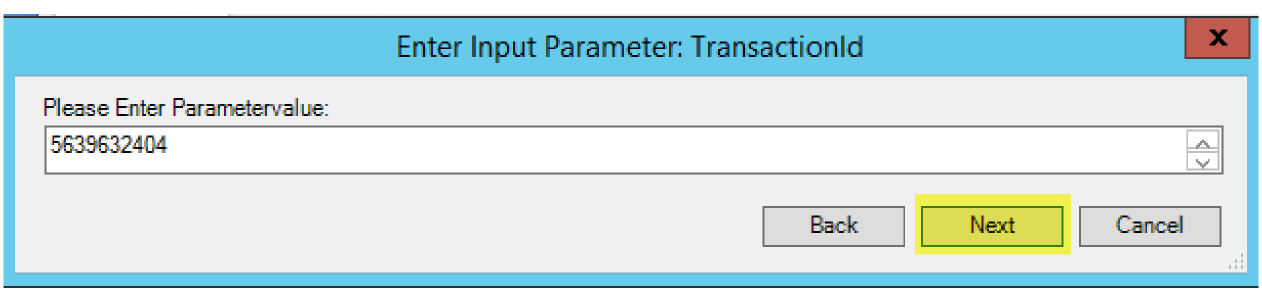
   a. In SQL Query -> Preview Data, we can check the data of the last report execution




Click on Preview Data shows the database connection:




Followed by the LatestTransactionID:



(If the latest LatestTransactionID is empty, then the Word-Template has not been called in the last few days - the temporary report data tables are deleted every 7 days, see setup of batch job)

Afterwards, we get the data behind the report, which we then can analyze:

(9 records)

| | Contact | Name | Partition | Phone | PhoneLocal | RecId | VendAccount |
|---|---|---|---|---|---|---|---|
| ▸ | | [illegible] | 5637144576 | [illegible] | | 1328 | 20001 |
| | | [illegible] | 5637144576 | [illegible] | | 1376 | 20002 |
| | | [illegible] | 5637144576 | [illegible] | | 1424 | 20003 |
| | | [illegible] | 5637144576 | | | 1472 | 20004 |
| | | [illegible] | 5637144576 | [illegible] | | 1520 | 20005 |
| | | [illegible] | 5637144576 | [illegible] | | 1568 | 20006 |
| | | [illegible] | 5637144576 | [illegible] | | 1616 | 20007 |
| | | [illegible] | 5637144576 | [illegible] | | 1664 | 20008 |
| | | [illegible] | 5637144576 | | | 1712 | 20009 |

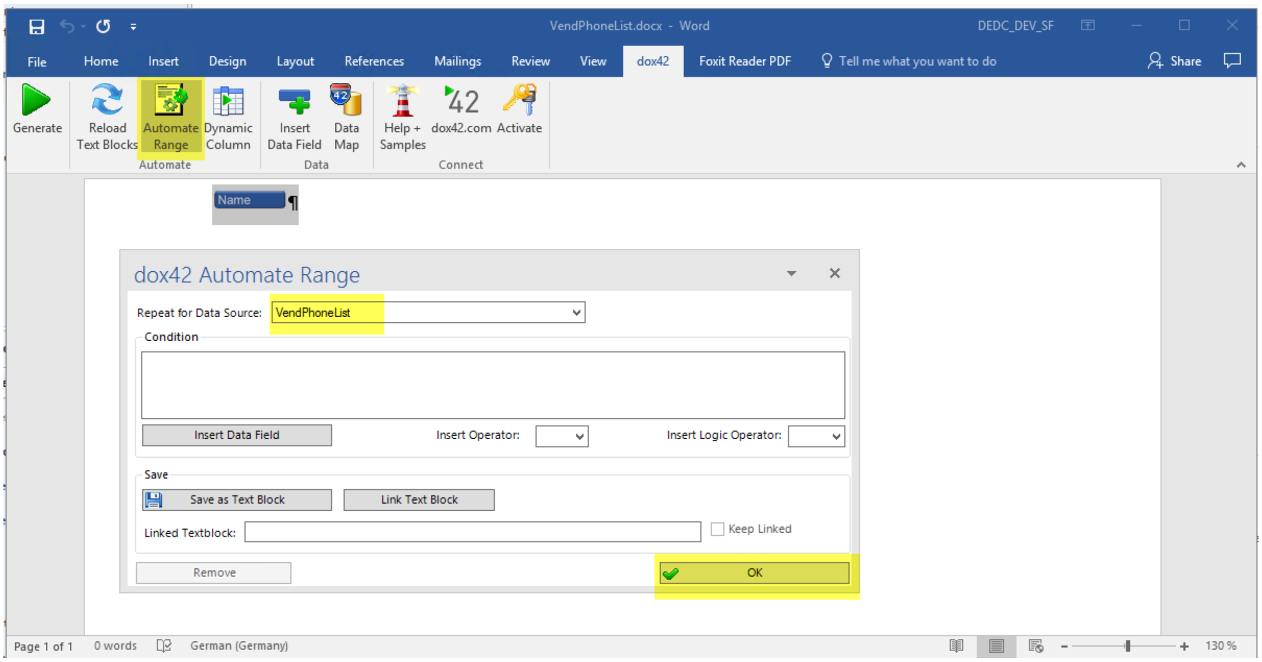
Important:

- Check the number of records – sometimes you get more or less records than expected. Then, find out the reason.
- Check the type fields – normally, AX uses one flat table for header and lines data, or for lines and totals. In these cases, there is normally some type field that explains the difference.

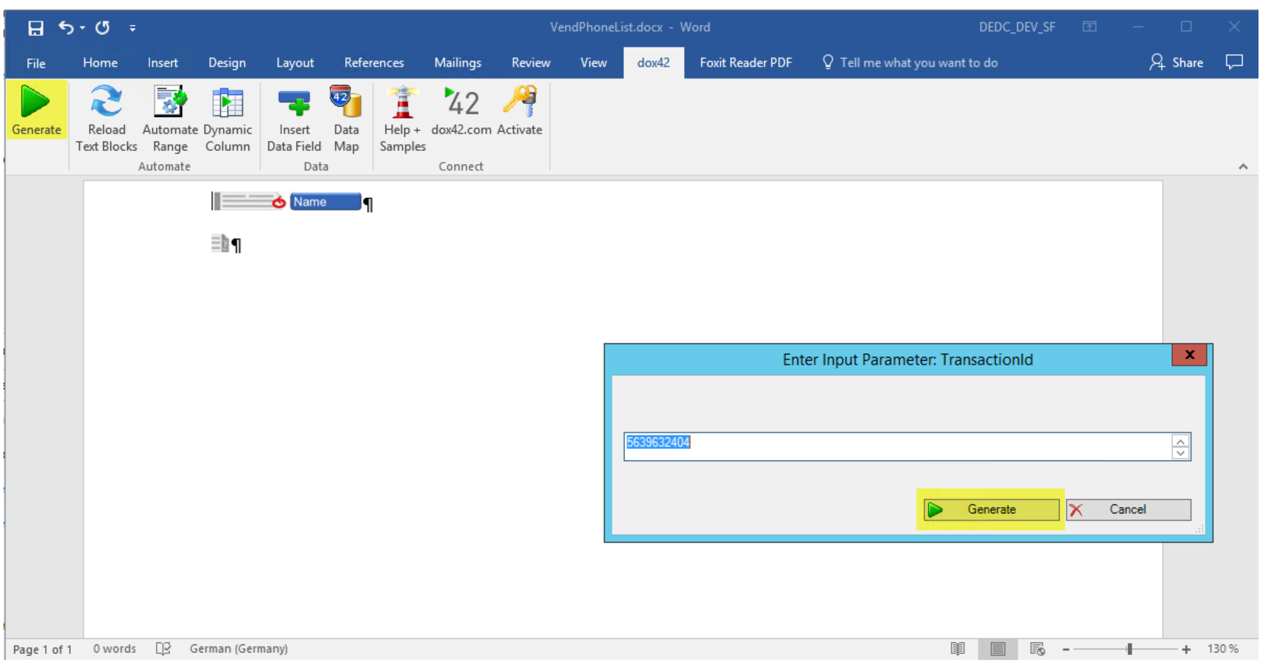
If columns are missing that are needed, 3 possibilities:

- Extend the data basis of the report (so extend the AX query, or extend the AX Report Data Provider class + the AX temporary table)
- Extend the data basis of the report by a data method (display method) on the temporary table
- Use a custom SQL statement (not recommended!!!)

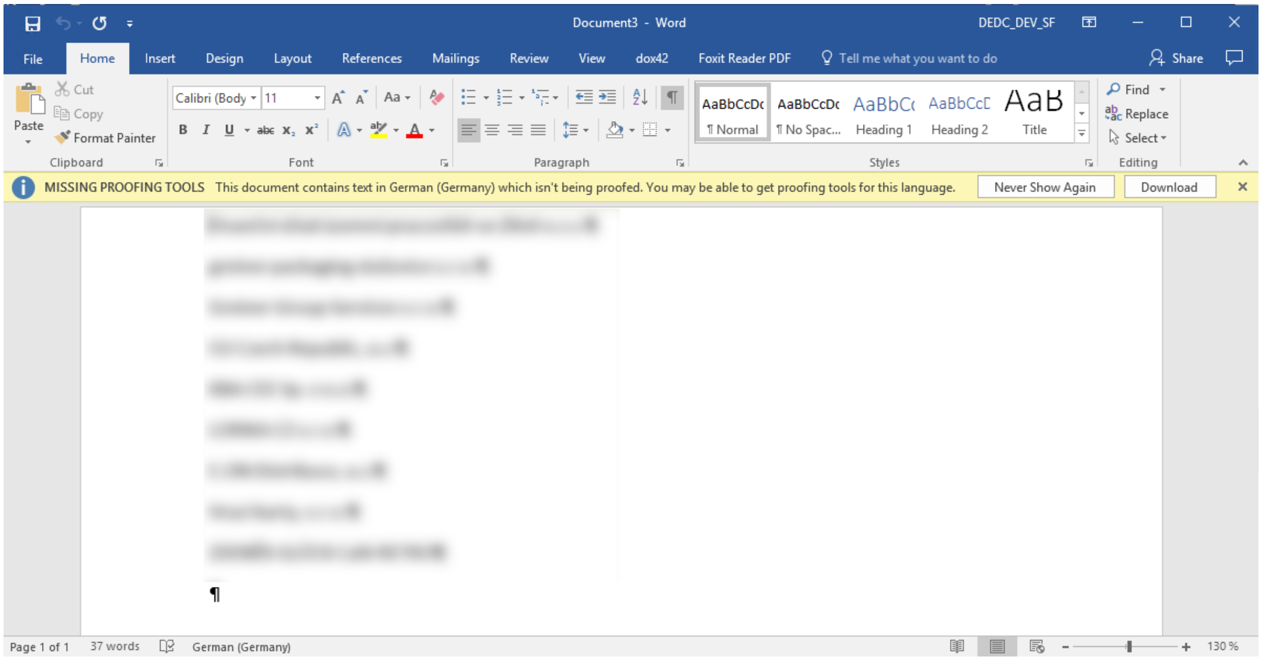
b. In our example, we would now most probably use a repeating automate range:




c. Further tests can now be made via Generate, directly from the dox42 client (data from last report execution is used)



Result: new document (not the template, but the report itself) – and since we implemented the repeater, we already get several lines:




Advantage of Generate button is also that errors are displayed directly in the dox42 client.

If a report is called from AX and errors occur, these are logged in the log file from the dox42 server.