



Manual

Data export from ProMISe 4 to SPSS

Datum	Naam	Actie	Definitief
10-07-2014	S. van Rooden	Opgesteld	
	S. van Rooden	Update en uitbreiding	
31/10/2016	M.Scholten	Gecheckt tbv introductie ProMISe4, niets gewijzigd.	
29/11/2016	M.Scholten	Kleine tekstuele wijzigingen (ProMISe3 in ProMISe4 gewijzigd)	
02/11/2017	M.Scholten	Toevoeging aanpassen syntax in het geval van lange tekstvelden in het bestand	
29/10/2018	M. Roza-Scholten	Toevoegen uitleg om met Access te werken na installatie Windows Office 2016	
11/11/2019	M. Roza-Scholten	Extra uitleg over 32 en 64 bits versies van Access en SPSS en info over applicatie 'Access Runtime 2016 – Install'	
2/11/2020	J Manniën	Tekstuele aanpassingen	2/11/2020



1 Table of Contents

1	Table of Contents.....	2
2	Introduction.....	3
3	Export settings in ProMISe	3
3.1	Purpose of export	3
3.2	Database type.....	4
3.3	SPSS version to use.....	4
3.4	External names for variables	4
3.5	Short and long variable names	4
3.6	SPSS data reader.....	4
3.7	Folder to store ZIP file.....	5
3.8	Export data file	5
4	Create SPSS .sav file.....	6
4.1	Extract data from ZIP file	7
4.2	SPSS Syntax (.SPS) file	7
4.3	Long text fields.....	8
4.4	Table structure.....	9
5	Combine SPSS .sav files.....	10
5.1	Combine tables with more than 255 variables	10
5.1.1	Combine tables with more than 255 items in SPSS by the default syntax in the export....	10
5.1.2	Combine tables with more than 255 items in SPSS by a syntax in general.....	10
5.2	Combine various tables	11
5.2.1	Combine various tables by the default syntax in the export	11
5.2.2	Combine various tables by a syntax in general.....	11
6	Restructure SPSS .sav files	12
	Appendix I: specifying the correct SPSS data reader.....	14
	Appendix II: workaround to use Access with Windows Office 2016	16



2 Introduction

In this manual the procedure of exporting data from ProMISe to SPSS is described.

Additionally, the final chapter of this manual describes how various tables in your export can be combined to one dataset that can be used for analysis.

3 Export settings in ProMISe

In ProMISe go to Topic Export:



FIGURE 1 TOPIC EXPORT IN PROMISE

3.1 Purpose of export

The purpose of your export (FIGURE 2) can be a 'back up' or a 'report' of your data. A 'back up' is an export of (a selection of) your data, which may consist of more than one table. A 'report' is an export of a report/query that you created in ProMISe, in which data of more than one table can be combined.

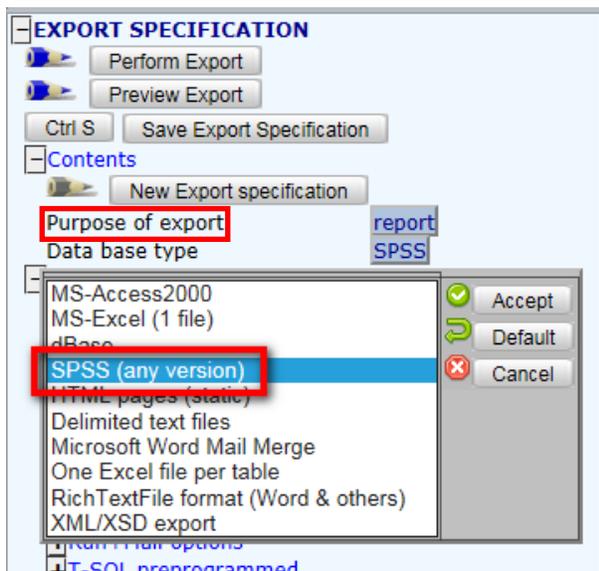


FIGURE 2 EXPORT SPECIFICATION IN PROMISE



3.2 Database type

Choose 'SPSS (any version)' (FIGURE 2) as the "data base type". Run a preview (button 'preview export') to make sure that you have the correct Filters activated and the data are indeed what you expect them to be.

Take a look at the possible parameters for the conversion:

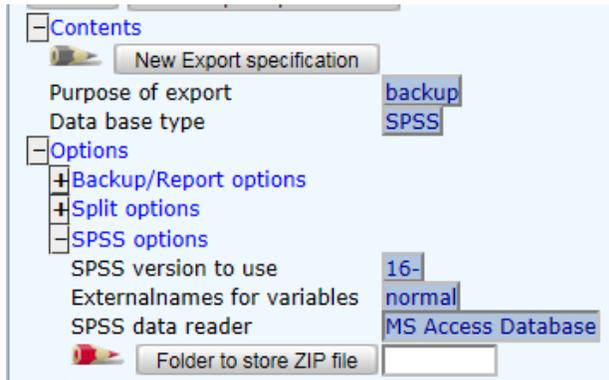


FIGURE 3 SPSS SPECIFICATION IN PROMISE

3.3 SPSS version to use

The default **SPSS version to use** is 16 – (16 or higher; FIGURE 3). If you use an older version of SPSS than 16, you will have to specify your version once and modify it only when you upgrade SPSS.

NOTE: It might be a problem if Access and SPSS do not have the same bits. Check that for both programs the 32 bit (called x86 in SPSS) or the 64 bit (x64) version are installed on your computer. In Access you can find the number of bits if you go to File, Account, About Access in a blank Access document. In SPSS you can find it on the License Authorization Wizard.

3.4 External names for variables

This option is 'normal variable names' by default (FIGURE 3). In some projects external names are defined. In that case, you may choose to select 'use external names'.

3.5 Short and long variable names

In each project can be defined whether short or long variable names are exported from ProMISe to SPSS, as well as short or long value labels. This selection cannot be defined or changed here.

3.6 SPSS data reader



The option **SPSS data reader** (FIGURE 3) you will need to specify only once (until you install another Windows version) and is usually already correct (MS Access Database). See **Appendix I** for more information on this.

3.7 Folder to store ZIP file

When you export the data, you will get a zip file, containing an Access file (.MDB file) with the data and an SPSS syntax (.sps file). Running this syntax will create an SPSS .sav file containing your exported data. As in the syntax the location of your data files is specified, you will have to specify this location in advance in this field. For instance like FIGURE 4: **D:\ANALYSES\MYTHESIS\MYPAPER** (screenshots in Windows 7).

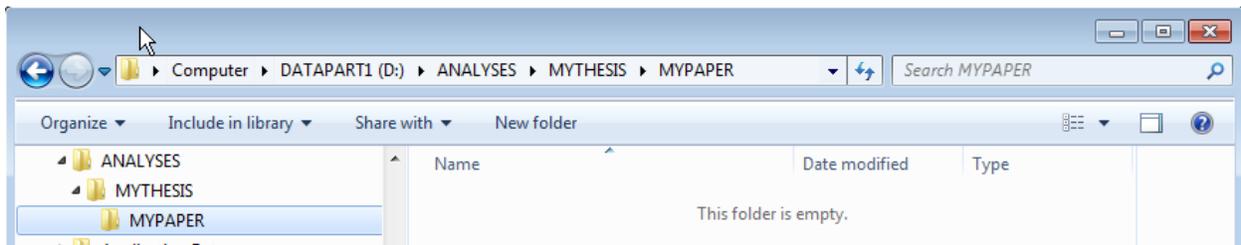


FIGURE 4 EXAMPLE OF FOLDER WHERE ZIP FILE IS STORED

You need to specify this location in ProMISe **in advance** (FIGURE 5) and you must later on store the ZIP file here.

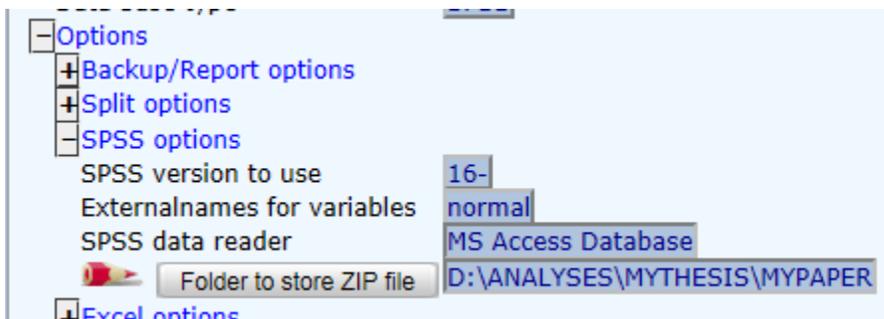


FIGURE 5 FOLDER TO STORE ZIP FILE

3.8 Export data file

Now you can run the export job ('Perform Export', FIGURE 6), with optional Item Filters, Record Filters or Population filters; possibly based on an Advanced Query et cetera.



FIGURE 6 PERFORM EXPORT IN PROMISE

When it is ready, download the ZIP file and store it in the folder specified:

Click on the link (paper/floppy icon)

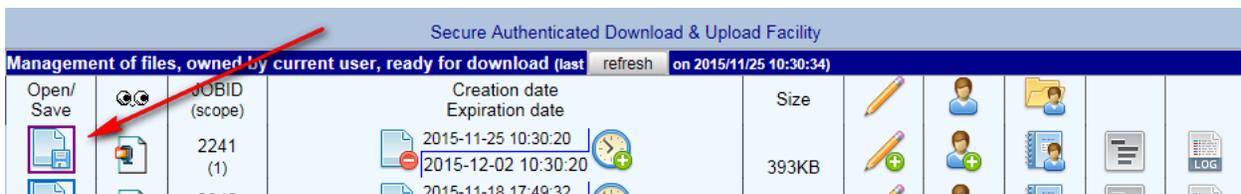


FIGURE 7 SELECT EXPORT FROM THE DOWNLOAD QUEUE

And choose the option 'save as':



FIGURE 8 SAVE EXPORT AS...

Choose the correct folder:



FIGURE 9 CHOOSE CORRECT FOLDER TO STORE DOWNLOAD

And click [Save].



FIGURE 10 MESSAGE AFTER DOWNLOAD HAD COMPLETED

4 Create SPSS .sav file



4.1 Extract data from ZIP file

Go to the folder where you have stored the zip file. Make sure you will unzip the file in the same folder! If you use winzip, for instance, select the option “Extract to here”. Please verify that the folder with the extracted data has the same name as your ProMISe username, that is the first part of the ZIP file name, up to the underscore (FIGURE 11).

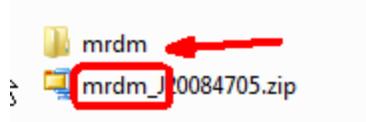


FIGURE 11 ZIP FILE AND EXTRACTED FOLDER WITH CORRESPONDING USERNAME

4.2 SPSS Syntax (.SPS) file

Open the subfolder (in the example it is called mrdm) and you will see the 2 files required to get your data into SPSS: MS ACCESS (.MDB) file and SPSS Syntax (.SPS) file (FIGURE 12).

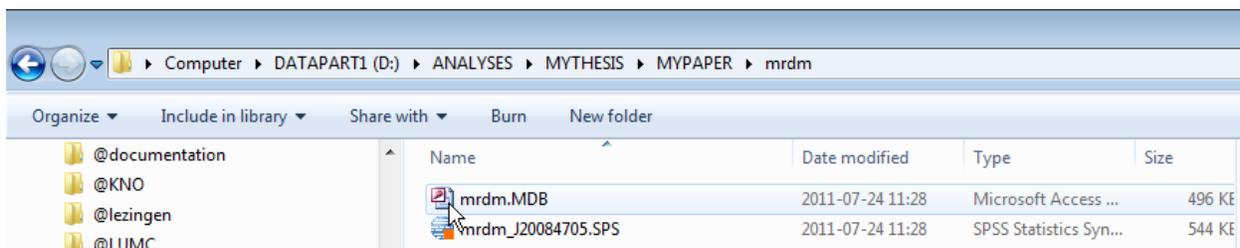


FIGURE 12 THE .MDB FILE AND THE .SPS FILE IN THE EXTRACTED SUBFOLDER

Open the SYNTAX file:

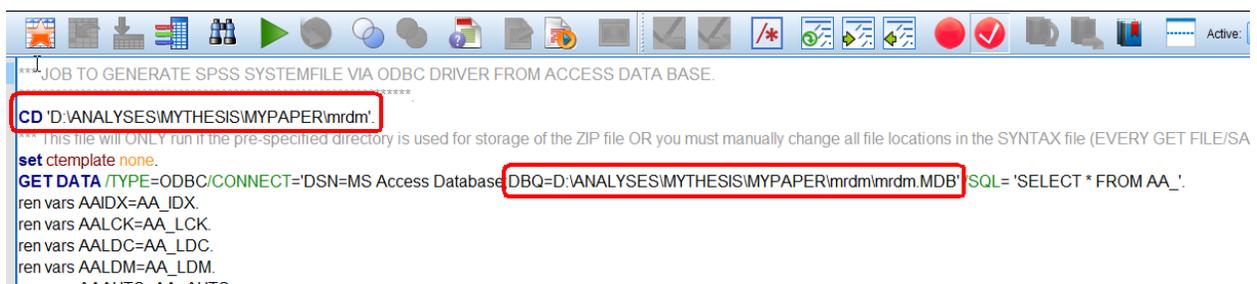


FIGURE 13 SYNTAX FILE OF PROMISE EXPORT TO SPSS

You will see that in the syntax file the paths are already defined: ProMISe has to actually write the syntax file for you and first of all it has to execute a so-called ‘Change Directory’ command (CD) to force SPSS to work in the subfolder with your username attached to the specified path. And secondly, it has to instruct the correct ODBC driver (here ‘MS Access Database’) as well as the location of the .mdb file. These locations are derived from the information that was entered at the ‘folder to store the ZIP file’



(see paragraph 3.7). If you inadvertently use the wrong name, you can also change it to the correct name directly in the SPSS Syntax file.

You can now RUN it (choose option 'Run → All'; FIGURE 14) and it will produce the relevant SPSS files. All in the current folder.

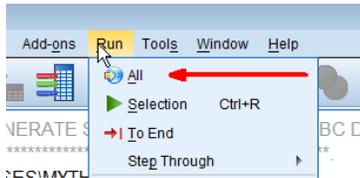


FIGURE 14 SYNTAX: RUN ALL

NOTE: If you have Office Windows 2016 installed on your computer you cannot use Access automatically in the way you used to do. For a workaround, see **Appendix II**, which you only have to run once.

4.3 Long text fields

If you want to export long text fields ('strings' with a length of over 255 characters) to SPSS, running the syntax can be very slow and lead to failure. This can be prevented by changing the SPSS syntax file a little before running it.

Open the syntax, in the line after the line with 'SET CACHE 1.' (usually line 14) you will find two lines which start with 'GET DATA', the first is in full colour, the second starts with ** and is grey-coloured and inactive:

```
14 SET CACHE 1.  
15 GET DATA /TYPE=ODBC/CONNECT='DSN=MS Access Database;DBQ=H:\temp_downloads\mscholten_MC_GLOBAL  
16 ** GET DATA /TYPE=ODBC/CONNECT='DSN=MS Access Database;DBQ=H:\temp_downloads\mscholten_MC_GLOB
```

You have to inactivate the first one by inserting a * (then it turns grey) and activate the second line by deleting the **. The second line will now be in full colour:

```
14 SET CACHE 1.  
15 * GET DATA /TYPE=ODBC/CONNECT='DSN=MS Access Database;DBQ=H:\temp_downloads\mscholten_MC_GLOBAL  
16 GET DATA /TYPE=ODBC/CONNECT='DSN=MS Access Database;DBQ=H:\temp_downloads\mscholten_MC_GLOBAL_c
```

The line in full colour ends with

```
'SELECT * FROM AA_'ASSUMEDSTRWIDTH=32767.
```

Which means the syntax assumes all strings to have a maximum width of 32767 characters. If you are sure the maximum length of text fields in your file is lower, you can change it to a lower number (e.g. 1000) to increase speed.

Make sure you save the altered syntax before running it in case you have to run it again in the future.



4.4 Table structure

The type and number of .sav files you will get, depend on the structure of your project and export. When you export on the basis of a Query (Report), you will only see one SPSS data file (typically called 0.sav). However, if you exported a “backup”, you will see as many files as there are tables in your relational structure of the project (FIGURE 15 shows an example).

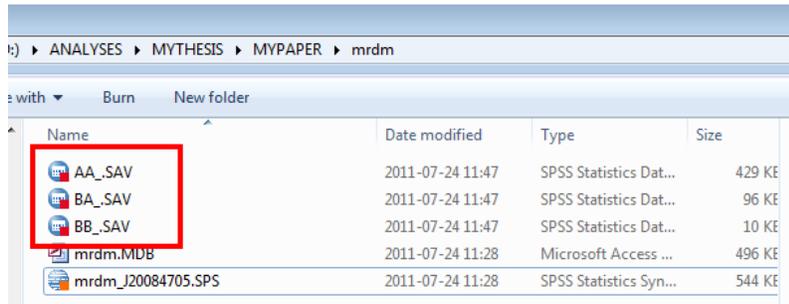


FIGURE 15 EXAMPLE OF .SAV FILES AFTER RUNNING THE SYNTAX

You will find a main table, usually patient-table (AA_.SAV). Based on the project structure, you may also find sub-tables (BA_.SAV, BB_.SAV) or even tables on lower levels (sub-subtables; CA_.SAV, CB_.SAV, etc or sub-sub-subtables DA_.SAV, DB_.SAV). An example of a table structure is shown in Table 1.

PHYSICAL TABLE TAG	LEVEL	BRANCH	PARENT TAG	TOTAL TAG	INDEX EXPRESSIONS	TABLE DESCRIPTION
AA_	1	0		AA_	ID, IDAA	PATIENTS
BA_	2	0	AA_	AA_BA_	ID, IDAA, IDAABA	ASSESSMENTS
BB_	2	1	AA_	AA_BB_	ID, IDAA, IDAABB	DIAGNOSES
BC_	2	2	AA_	AA_BC_	ID, IDAA, IDAABC	TREATMENTS
CA_	3	0	BC_	AA_BC_CA_	ID, IDAA, IDAABC, IDAABCCA	DRUGS
BD_	2	3	AA_	AA_BD_	ID, IDAA, IDAABD	QUESTIONNAIRES
BE_	2	4	AA_	AA_BE_	ID, IDAA, IDAABE	ADVERSE EVENTS

TABLE 1 AN EXAMPLE OF A TABLE STRUCTURE

All tables are indexed by key variables:

AA_ = ID and IDAA

BA_ = ID, IDAA, IDAABA

BB_ = ID, IDAA, IDAABB

BC_ = ID, IDAA, IDAABC

.....etc.



CA_ = ID, IDAA, IDAABX (with X depending on the table of which CA_ is a sub table), IDAABXCA
CB_ = ID, IDAA, IDAABX, IDAABXCB
...etc.

The combination of key variables makes every record unique and indicates the relational structure. It also offers the possibility to combine the various tables in a logical manner (see chapter 0).

In some exports you will find, for example, a BA_ and a BA_1 table. This will be the case when the table consists of more than 255 variables. Access will split the dataset into two tables (or more tables with every 255 variables extra). These tables will be combined in SPSS to one dataset after running the syntax (see paragraph 5.1).

5 Combine SPSS .sav files

Separate tables in Access will be exported to separate SPSS tables. Below is described how the tables are combined in the syntax, to understand what happens and to deduct how data is combined in the SPSS syntax. You may want to adapt the syntax if you want to create other combinations, rename datasets etc...

It may also be that you want to combine various tables; after you have exported them from the ProMISe database (It is also possible to combine tables in ProMISe, see the manual ProMISe data retrieval).

5.1 Combine tables with more than 255 variables

5.1.1 Combine tables with more than 255 items in SPSS by the default syntax in the export

If you have exported more than 255 items from a table, these are split in Access, and will be combined in SPSS by running the commands at the end of the syntax (an example) in FIGURE 16.

```
SORT CASES BY RECORD_.  
**** We combine this extension file with the first one:  
MATCH FILES FILE='0.SAV'/TABLE=*/BY RECORD_.  
SORT CASES BY ID,IDAA.  
SAVE OUT='0.SAV'.
```

FIGURE 16 EXAMPLE OF SYNTAX STATEMENTS WHERE SPLITTED TABLES ARE REUNITED IN SPSS

5.1.2 Combine tables with more than 255 items in SPSS by a syntax in general

Tables with more than 255 variables that were split in access can be easily combined with the variable 'RECORD_'. The number in this variable corresponds to the same records in de separate tables (e.g. BC_.sav and BC_1.sav). You can use the following syntax:



```
match files files 'XX_.SAV'/table='XX_1.SAV'/by RECORD_.
```

Before running this syntax *XX* needs to be replaced by the name of the table you want to combine (e.g. BA, BC, CA, etc.).

5.2 Combine various tables

5.2.1 Combine various tables by the default syntax in the export

If you have exported information from various tables in a backup from ProMISe, these will be combined by running the commands at the end of the syntax (an example) in FIGURE 17.

```
*** In order to COMBINE several .SAV files with their respective parent file(s), you may use the following suggested syntax.  
*** These are examples how to create for each table the combination with all its parents.  
  
get file='AA_.sav'.  
save out='PATIENTS.sav'.  
get file='BA_.sav'.  
match files file='*/table='AA_.sav'/by ID,IDA.A.  
save out='HIP_OPERATIONS.sav'.
```

FIGURE 17 EXAMPLE OF SYNTAX STATEMENTS WHERE DIFFERENT TABLES ARE COMBINED

In this example, the AA_.sav file will be saved as an SPSS dataset that is named 'PATIENTS.sav' (This was the original name of the AA_ table in ProMISe). Secondly, all data in the BA_.sav file is combined with the data of the AA_.sav file (every row contains a record from the BA_ table, with information from the AA_ table added. The file with the combined data is named 'HIP_OPERATIONS.sav' (this was the original name of the BA_ table in ProMISe).

5.2.2 Combine various tables by a syntax in general

Different tables can be combined with the key variables that you will find in every table that you have exported from ProMISe. You can use the command MATCH FILES in SPSS. Below you will find an example of a syntax in which a top-table (AA_, usually patient table) is combined with a sub table (BA_, which could contain surgeries, treatments, questionnaires, etc.).

```
* First indicate the correct directory of your files.  
cd 'H:\MyDocs\.....'.
```

```
* In order to run the command MATCH FILES, data need to be sorted on the key variables.  
get file='BA_.sav'.  
sort cases by id, idaa, idaaba.  
save outfile='FU.sav'.
```

```
* similar for the second table:.  
get file='AA_.sav'.  
sort cases by id,idaa.
```



```
save outfile='patients.sav'.
```

** finally both tables are combined with the following command:*

```
MATCH FILES /FILE='FU.sav'  
/TABLE='patients.sav'  
/BY ID IDAA.  
EXECUTE.  
save outfile='patient_and_followup.sav'.
```

You can add the commands DROP and KEEP to specify the variables that you want to exclude or include from your dataset. Replace the last row in the syntax by

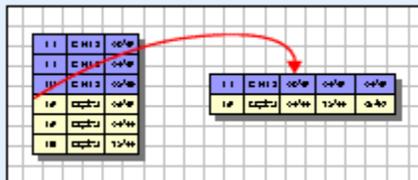
```
save outfile='patient_and_followup.sav'/DROP=XY.
```

(where XY is the name of the variable that you want to exclude).

6 Restructure SPSS .sav files

Whether and how your data need to be restructured depends on the analyses you want to perform in SPSS. Below you will find an introduction on how you can restructure your dataset in SPSS.

You can restructure your database in several ways. The most common way is when you have a database with several rows per patient (e.g. several hospital visits, with each row representing one visit) and you want to rearrange the database into one row per patient (FIGURE 18).



Restructure selected cases into variables

Use this when you have groups of related cases that you want to rearrange so that data from each group are represented as a single case in the new data set.

FIGURE 18 RESTRUCTURE A DATASET FROM CASES TO VARIABLES

You can either use the following syntax:

```
SORT CASES BY ID IDAA.  
CASESTOVARS  
/ID=ID IDAA  
/GROUPBY=INDEX.
```

Or you can choose in the Menu bar 'Data → Restructure' and follow the steps (FIGURE 19).



Cases to Variables: Select Variables

Data from case groups in the current file will be restructured into single cases in the new file.

Choose variables that identify case groups by moving those variables into the Identifier Variable list. Optionally you can also choose Index Variables.



The variables that remain in the list of Variables in the Current File either contain data that vary within a case group or data that do not vary.

A variable with data that vary will become a group of new variables in the restructured file. A variable with data that do not vary will be copied into the new file.

Variables in the Current File:

- Geboortedatum van de patiënt (jjjj-m...
- Geslacht van de patiënt [GESLACHT]
- Datum waarop de patiënt is opgeno...
- Consult [CONSULT_DATUM]
- Baseline of Follow Up Consultrecor...
- Anamnese [Anamnese]
- De observatie dat de patient klachte...

Identifier Variable(s):

- UMC (anoniem) [ID]
- Dossier (anoniem) [IDAA]

Index Variable(s):

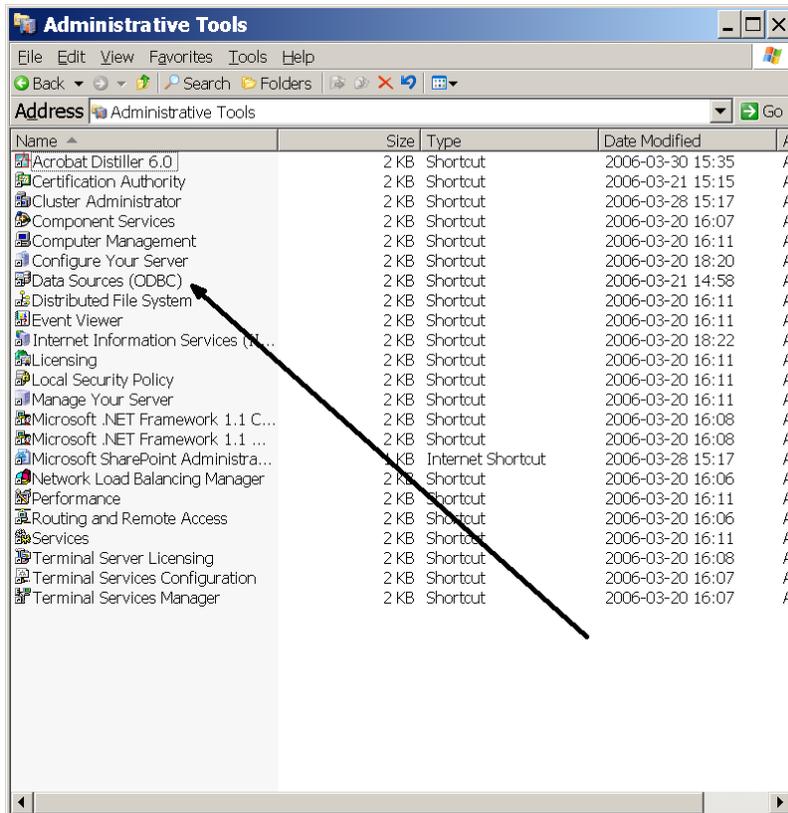
FIGURE 19 WINDOW IN SPSS TO COMPLETE WHEN RESTRUCTURE DATA FROM CASES TO VARIABLES



Appendix I: specifying the correct SPSS data reader.

The “SPSS data reader” is the standard name for the ODBC interface on the majority of PC’s in Europe, especially when they are configured for the English language.

However, for example German installations of any Windows version have the same driver but with a German name! In other words, the ODBC driver (which allows SPSS to read a MS-Access database to get the exported data) has a name which depends on the actual installation of Windows.

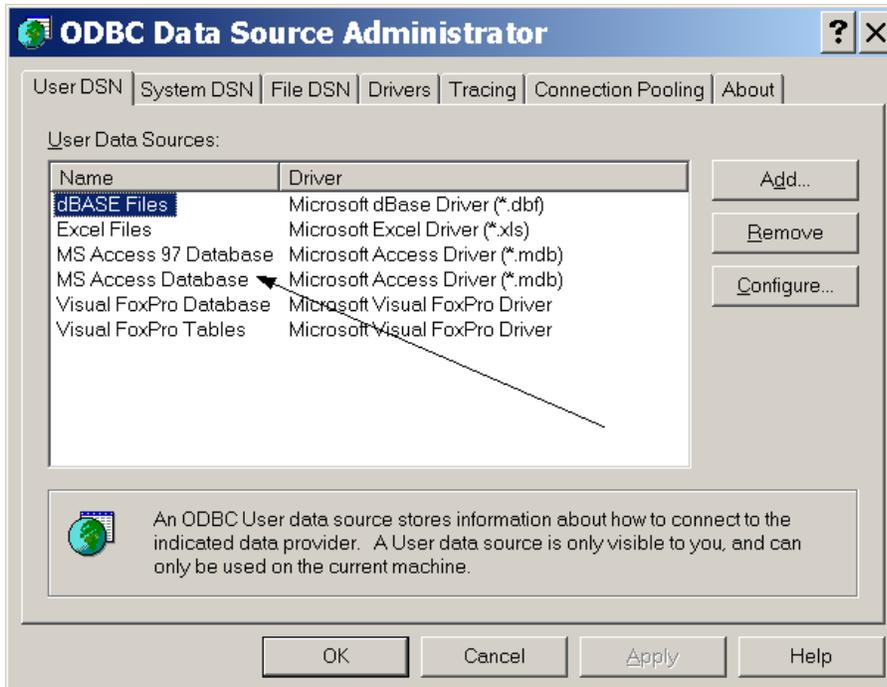


How do you know which name YOUR ODBC driver has? You can do this as follows (again: not needed on the majority of PC’s).

Open the CONTROL PANEL of your Windows version (via “Start”, “Settings”, “control panel” and click on “administrative tools”. You should see this menu in explorer (left).

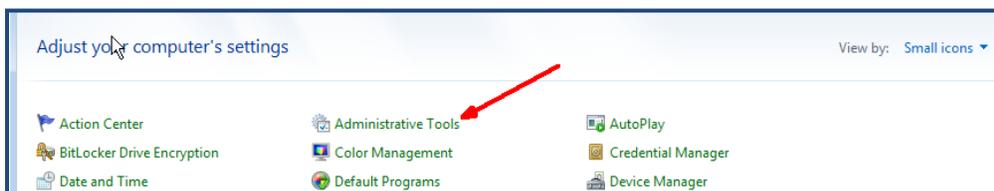
Newer versions of Windows show a slightly different layout (below).

Double click on “Data Sources (ODBC)”:



- IF you see an entry called “MS Access Database” spelled exactly like in the ProMISe export menu, you are OK!
- IF you see an entry which ALSO refers to a Microsoft Access Driver (*.mdb) (for example: “MS Access Daten Bank”, then use that **EXACT NAME** also in ProMISe as the option “SPSS data reader”.
- IF you don’t have any such option in this window, click on “Add...” and add a *.mdb driver (following the windows instructions) and give it as a title the SAME as you use in Promise.

Close the Administrator and Control panel.



Now you can run any conversion to SPSS.

A bit of explanation: a conversion to SPSS consists of a regular MS-Access data base (.mdb) and an accompanying SPSS syntax file (.sps). When you run this syntax within SPSS, the very first instruction SPSS gets is the reading of the MS Access data base into SPSS format (memory). For this read-action SPSS relies totally on the Windows supplied so called ODBC driver. Since ProMISe has to construct the syntax beforehand (while running the export) you have to tell ProMISe beforehand how this ODBC driver is called on your PC, otherwise you would have to instruct SPSS to use the correct ODBC driver each time you convert.



Appendix II: workaround to use Access with Windows Office 2016

This workaround has to be run only once. If along the way you get a message that somehow you have installed a 64 bits version as well as a 32 bits version, ask someone from the ICT-department for help.

If you work in the LUMC, you can install this application from the software centre:



Access Runtime
2016 - Install
Microsoft
2016 x86

If you are working elsewhere, use this work around:

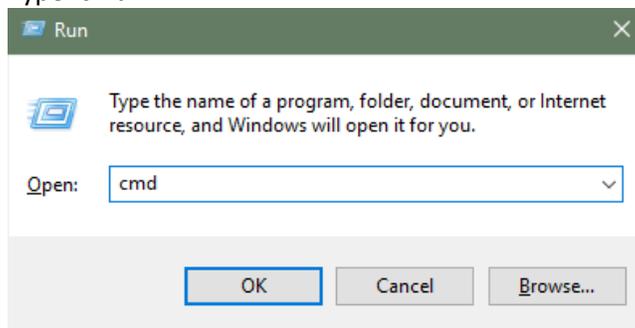
Step 1: download runtime 2016

<https://www.microsoft.com/en-us/download/details.aspx?id=54920>

- Choose the AccessDatabaseEngine.exe (=32 bits version)
- “ save as” to your general download folder (or elsewhere, but then you have to be careful with the location in the next steps)

Step 2: Run with a command prompt:

- Open run-dialog: Windowskey + R
- Type: cmd



- Click ok.
- You'll get a command shell:
 - o Go to the download folder: `cd Downloads`
 - o Start installation met the option “/quiet”: `AccessDatabaseEngine.exe /quiet`



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.16299.492]
(c) 2017 Microsoft Corporation. All rights reserved.

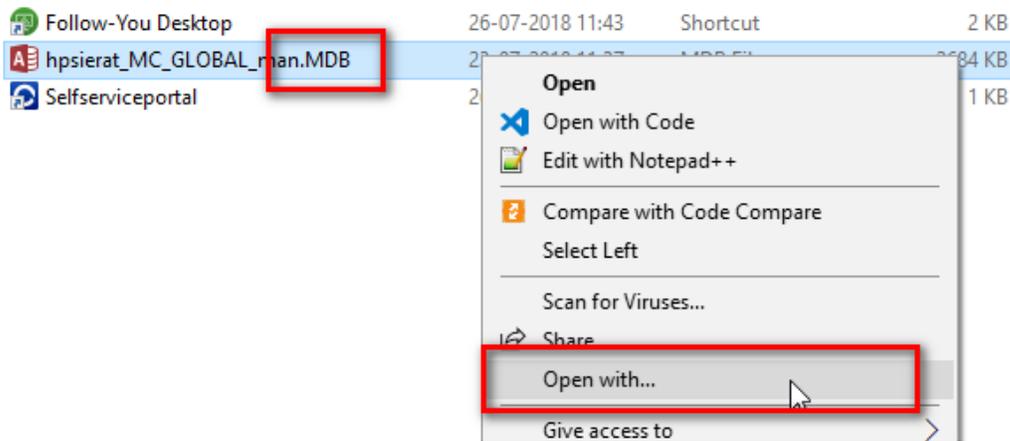
C:\Users\hpsierat>cd Downloads

C:\Users\hpsierat\Downloads>AccessDatabaseEngine.exe /quiet
```

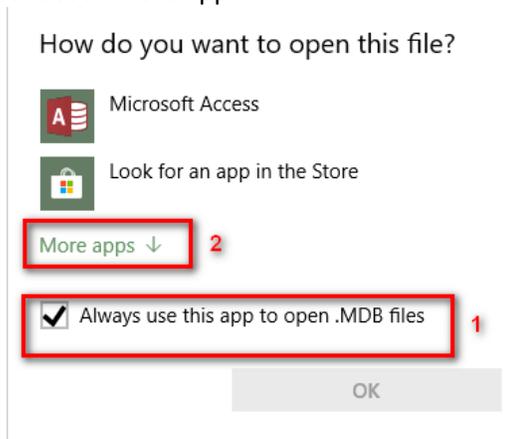
- In the dialogue screen that asks for permission, click ok.

Step 3

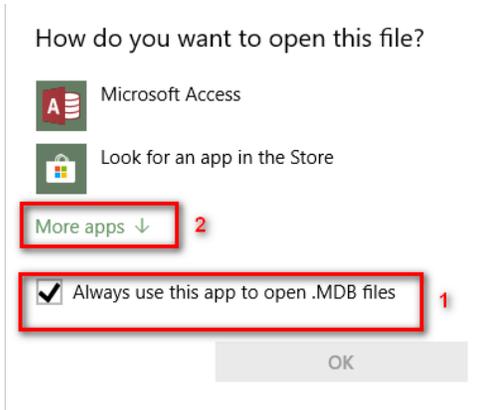
- Click right on a .mdb-file:
- Choose: Open with...



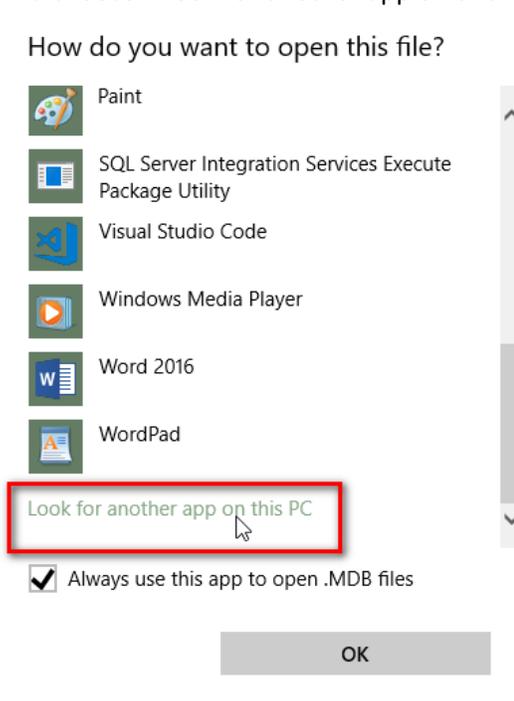
- Choose option "Always use this app to open .MDB file".
- Choose "More apps"



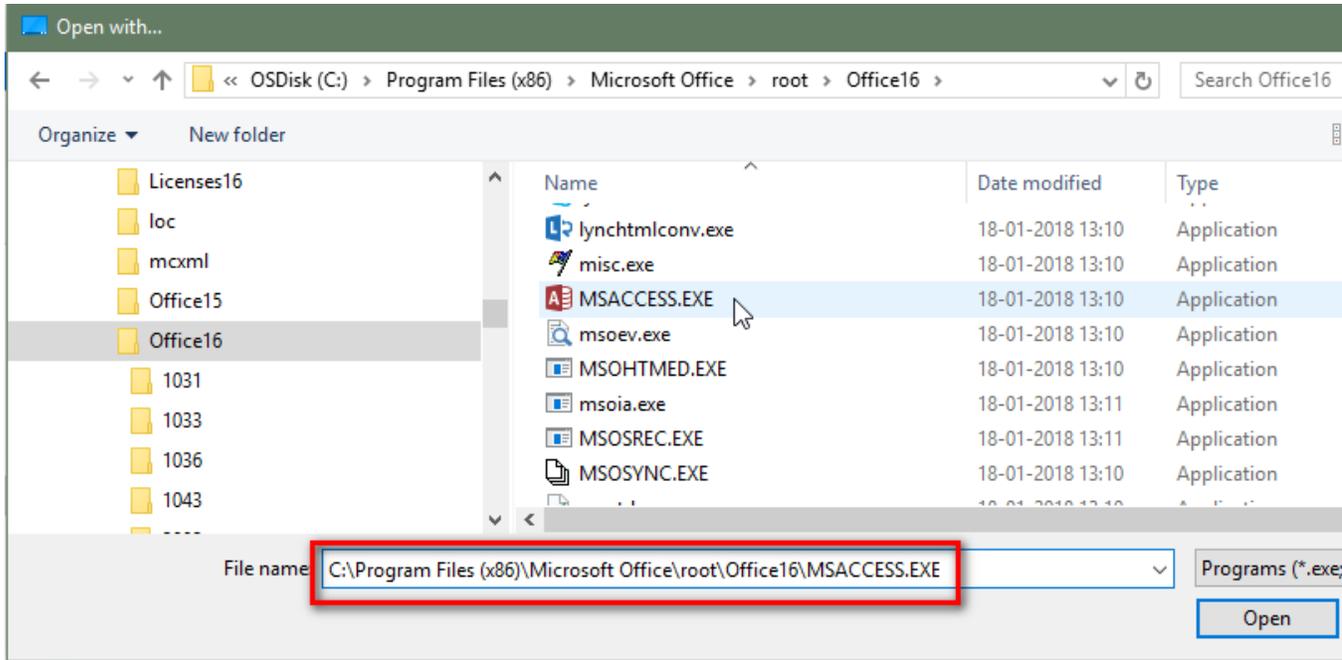
- Scroll to the bottom of the list of programs



- And choose “Look for another app on this PC”



-
- Choose “File name”: C:\Program Files (x86)\Microsoft Office\root\Office16\MSACCESS.EXE



- Click “Ok”
- Now you’ll see the right icon and can open your Access file by double clicking on a .mdb file

