Paper SM07

Automated creation of a mapping specification for a pooled database

Thomas Wollseifen, Syneos Health, Germany

ABSTRACT

Provision of SDTM/ADaM data as part of integrated analyses is an important part in preparing the submission of data and reports from clinical studies to authorities. Integrated databases could change over time. I would like to introduce a SAS macro that is used for the creation of a mapping description of several studies that also takes changes to the metadata / data over time into account. A simple macro call creates an Excel file based on the study data and metadata from the pooled database. Mapping suggestions and information from the content of variables is included in the mapping description. Earlier descriptions of studies are automatically integrated into the Excel file. Changes to the metadata are also taken into account. The completed mapping description can finally be used to create a define.xml. It also serves as a template for creating the mapping programs.

Mapping descriptions are an important part of the creation of pooled databases. There are certainly approaches to map data automatically using metadata. This can be achieved quite easily, especially with SDTM, since in SDTM the data is not derived in a complex manner and no additional analysis summary variables are added. In the case of ADaM data, however, it becomes more difficult to create this automatically due to the summary data or derived variables that are dependent on different domains. Of course, programs/macros exist that do a semi-automatic mapping.

For documentation purposes, it is important to have a description of the mapping. Programmers need a specification when starting with mapping. Define.xml and the Reviewer's Guide also need mapping documentation.

In addition to SDTM and ADaM, legacy studies still exist that need to be mapped to an integrated database. The presented macro can create a mapping description based on the pool metadata and study data. It can included existing mapping descriptions and can handle changes in the metadata. New studies can be added automatically.

The following SAS macro call shows different studies with an SDTM data structure and a legacy study for which an assignment description is being created.

```
%create_meta_docu_xls(
    project=pharmaCompany Medication xyz
   ,libmeta=meta
   ,libextmeta=extmeta
   ,studyid=        10001#10002#1007
   ,pathstudydata=    &path10001
                     #&path10002
                     #&path10007
   ,inadsl       =  DEMO#dm#dm#
   ,domains=        ae#ce#cm#da#de#dm#ds#dv#ex#relrec#sc#se#sv#ta#te#ti#ts#tv#vs
   ,pathoutfile=    <outputpath>
   ,outfilename=    mapping_documentation_&date..xlsx
   ,standard=SDTM
   ,mapto         = %str(10001: #event->ce#med->cm#demo->dm#dispo->ds#dosing->ex#lab->lb)
  ,previousfile=mapping_documentation_2021-04-16.xlsx
);
```

First, an information sheet with the summary of the mapping of the different domains is displayed (Fig. 1).

	A	B	C	D
1	Domains Pool	SN_10001	SN_10002	SN_10007
2	AE	AE	AE	AE
3	CE	EVENT->CE	not available or not assigned	CE
4	CM	MED->CM	CM	CM
5	DA	DA	not available or not assigned	DA
6	DE	DE	not available or not assigned	not available or not assigned
7	DM	DEMO-DM	DM	DM
8	DS	DISP-DS	DS	DS
9	TA	TA	TA	TA
10	TE	TE	TE	TE
11	TI	TI	TI	TI
12	TS	TRIAL->TS	TS	TS
13	TV	VISITS->TV	TV	TV
14	VS	VITALS-VS	VS	VS

Fig. 1 Info Sheet with Mapping Assignment

In the following figure (Fig. 2) the mapping of a particular domain (DA) is shown. Colours indicate whether a variable exists in the pool metadata (green), does not exist (red) or has yet to be derived (beige). If a domain was not delivered at all, it is marked with light blue ('Information not delivered'). If substantial changes have been made in the description of an existing variable definition, it is automatically marked with dark yellow. This makes it easier for the programmer to define the mapping.

	A	B	C	D	E	F	G	H	I	J
1	DOMAIN	VARIABLE	VARNUM	LABEL	TYPE	CODELIST	TARGET	SN_10001	SN_10002	SN_10007
2	DA	DAREASND		Reason Not Performed				Information not delivered by study	Information not delivered by study	DA.DAREASND,Variable not in metadata,contains only missings
3	DA	DASTAT		Completion Status				Information not delivered by study	Information not delivered by study	DA.DASTAT,Variable not in metadata decided not to map
4	DA	DPAGEID		Datapage ID				Information not delivered by study	Variable is dropped	Information not delivered by study
5	DA	RECPOS		Record Position				Information not delivered by study	Variable is dropped	Information not delivered by study
6	DA	VISITDY		Planned Study Day of Visit				Information not delivered by study	Information not delivered by study	DA.VISITDY,Variable not in metadata
7	DA	STUDYID	1	Study Identifier	C		ADS SDTM	Data not provided	DA.STUDYID	Set to '10007' Study ID, original study ID is not used.
8	DA	DOMAIN	2	Domain Abbreviation	C	DOMAIN	SDTM	Data not provided	DA.DOMAIN	DA.DOMAIN,Codelist ist different;$
9	DA	USUBJID	3	Unique Subject Identifier	C		ADS SDTM	Data not provided	DA.USUBJID	DA.USUBJID
10	DA	SUBJIDN	4	Subject Identifier for the Study (N)	N		ADS	Data not provided	DA.SUBJIDN	numeric part of USUBJID without study ID part
11	DA	DASEQ	5	Sequence Number	N		ADS SDTM	Data not provided	DA.DASEQ	DA.DASEQ
12	DA	DAGRPID	6	Group ID	C		ADS SDTM	Data not provided	DA.DAGRPID	Not populated
13	DA	DAREFID	7	Reference ID	C		ADS SDTM	Data not provided	DA.DAREFID	DA.DAREFID
14	DA	DASPID	8	Sponsor-Defined Identifier	C		ADS SDTM	Data not provided	DA.DASPID	DA.DASPID
15	DA	DASMNUM	9	Study Medication Number	C		ADS SUPP	Data not provided	DA.DASMNUM	Not populated
16	DA	DATESTCD	10	Short Name of Accountability Assessment	C	DATEST	ADS SDTM	Data not provided	DA.DATESTCD	DA.DATESTCD,Codelist ist different;$ codelist DATEST is applied and new/different codes were requested/added.
17	DA	DATEST	11	Name of Accountability Assessment	C		ADS SDTM	Data not provided	Decode of DATEST	DA.DATEST
18	DA	DACAT	12	Category of Assessment	C		ADS SDTM	Data not provided	DA.DACAT	DA.DACAT, remapped according to ZDXYL codelist:
19	DA	DACATN	13	Category of Assessment (N)	N	ZDACAT	ADS	Data not provided	DA.DACATN	derived from DACAT according to ZDABCD codelist
20	DA	DACATO	14	Category of Assessment, Other	C		ADS SDTM	Data not provided	DA.DACATO	Not populated

Fig. 2 Excel file of domain and studies

There are also mapping suggestions if variables did not appear in the metadata. The decision as to how a variable should be mapped is made using the appropriate distance between variable names, labels and the type of variable. Variable names, labels and the type are compared with existing variables in the metadata and a score is then used to decide how the variable should be mapped.

Introduction to data flow in clinical studies

In clinical studies, data is recorded in CRFs via EDC systems. Additional data from laboratories or other sources is added. The data structure is converted from the source structure (raw database) into SDTM and then into ADaM. SDTM or ADaM data can also then be used for integrated databases (IDB), whereby e.g. the individual SDTM study is mapped directly into SDTM of the IDB. The IDB then serves as the basis for reports (e.g. ISS or ISE). The structure or mapping used is described in the reviewer's guide and the metadata and assignments are shown in the define.xml. Fig. 3 shows the flow of data in clinical studies from the source data to the IDB and analyses and submission to authorities.

The mapping from one structure to another is controlled manually (i.e. by program), semi-automatically, or fully automatically (e.g. via a metadata repository). In all cases, a structure A is converted into a structure B. This must then also be documented. SDTM and ADaM consist of different domains (such as AE, DM, CM in SDTM or from ADAE, ADSL, ADCM etc. in ADaM), which divide the individual sections of the CRF and the data into meaningful topics. In the various domains, there are variables with labels and formats that contain the respective data.

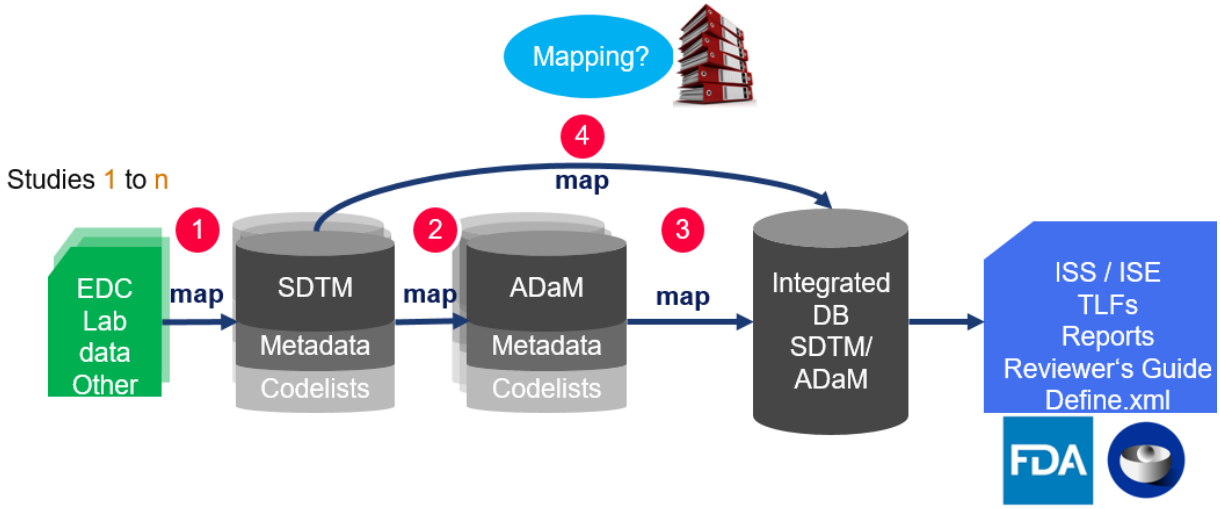


Fig. 3 Mapping in clinical studies

It is important that the mapping is documented at the respective transitions. An Excel file, for example, in which the metadata of the various domains is created as individual sheets with the variables, formats, and labels, is ideal for this purpose. For the various studies, you can create a column in the sheets that shows the mapping description of the respective study, domain and variable as the smallest structure.

We are also going to present a macro **%m_create_metadata_docu_xls** that simplifies the process of documentation for the programmer. The macro creates an Excel file that shows the metadata of the IDB and the data of the studies together. For each variable an assignment has either already been entered or has to be entered by the programmer. This is then highlighted in colour in the Excel file. We come to the details of the macro and the generated Excel file in the next sections.

The macro also takes into account various events of an IDB that can occur during the lifecycle of an IDB.

Features of the study flow that influence the IDB

- Study data is added at different timepoints
- Mapping might be changed due to new requests
- New variables and domains could be added
- Structure of the IDB might change
- SDTM or ADaM IDB
- Maintenance of the IDB (MedDRA updates, WHO-DD updates, corrections)
- Authority requirements

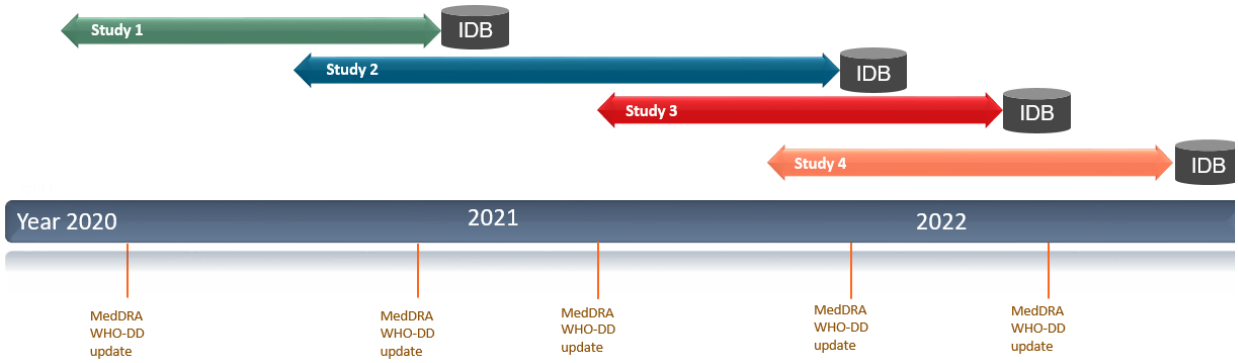


Fig. 4 Features of the study flow that influence the IDB

The respective properties and events may influence the mapping, the structure of the IDB or the documentation of the mapping. Fig. 4 shows the rough course of studies and where there is an influence on the IDB.

The macro %m_create_metadata_docu_xls creates an Excel file with sheets for the respective domains. In Fig. 5 we see a section of a sheet for the AE domain. On the left side, it shows the domain's variables and their label, type, length and associated codelist as the target structure. The target structure shows the metadata of the respective domains of the IDB with their variables. In addition, there are columns with the mapping description for the studies. Colours indicate whether a variable was found (green), not found (beige), or does not appear in the metadata of the domain at all (red). Details of the macro will be shown in later sections.

	A	B	C	D	E	F	G	H	I	J	K
1	DOMAIN	VARIABLE	VARNUM	LABEL	TYPE	LENGT	CODELIST	BSP_TARGET	SN_1000	SN_1001	SN_1002
24	AE	AESHOSPP							Variable does not exist	Variable does not exist	SUPPAE.QVAL where SUPPAE.QNAM eq. AESHOSPP
25	AE	AESHOSPR							Variable does not exist	Variable does not exist	SUPPAE.QVAL where SUPPAE.QNAM eq. AESHOSPR
26	AE	AESOSP							Variable does not exist	Variable does not exist	SUPPAE.QVAL where SUPPAE.QNAM eq. AESOSP
35	AE	AETRLPRC							Variable does not exist	Variable does not exist	SUPPAE.QVAL where SUPPAE.QNAM eq. AETRLPRC
71	AE	STUDYID	1	Study Identifier	C	10		OAD ADS SDTM		AE.STUDYID	AE.STUDYID,Length>meta data definition
72	AE	DOMAIN	2	Domain Abbreviation	C	2	DOMAIN	OAD SDTM		AE.DOMAIN	AE.DOMAIN,Codelist is different:,$
73	AE	USUBJID	3	Unique Subject Identifier	C	27		OAD ADS SDTM		AE.USUBJID	AE.USUBJID
74	AE	SUBJIDN	4	Subject Identifier for the Study (N)	N	11		OAD ADS		AE.SUBJIDN	
75	AE	AESEQ	5	Sequence Number	N	8.1		OAD ADS SDTM		AE.AESEQ	AE.AESEQ
76	AE	AENYN	6	Any AEs? (N)	N	3	NY	OAD		AE.AENYN	
77	AE	AEGRPID	7	Group ID	C	20		OAD ADS SDTM			
78	AE	AEREFID	8	Reference ID	C	30		OAD ADS SUPP		AE.AEREFID	

Fig. 5 Resulting Excel Sheet wit domain AE

As already indicated, the macro %m_create_metadata_docu_xls can react to certain events in the lifecycle of an IDB. Adding a variable to the IDB metadata is no problem for the macro %m_create_metadata_docu_xls. When the macro is called again, the variable is inserted at the corresponding position in the domain or sheet. The user can then enter or adapt the corresponding mapping description. Additional domains in the metadata of the IDB are not an obstacle either. Here you can simply add the respective domain in the macro call. New studies can also be easily added in the macro call and the corresponding columns in the Excel file are automatically added by the macro. An Excel file that has already been created can also be used as a source file at any time. New information is then automatically added via the macro %m_create_metadata_docu_xls. Later we will see some examples of the various possibilities of the macro, but first let's take a look at the structure of the macro. Fig. 6 shows the structure and sequence of the macro % create_meta_docu_xls.

Macro %m_create_metadata_docu_xls

The macro runs through the various domains and studies via do-loops. If several source domains are mapped to a single target domain, another do-loop runs through the source domains. Within the innermost loop, data checks of the respective domain of a study are carried out against the metadata of the target structure. At the end, the results are linked via an SQL join and sheets are created for each domain. Possible earlier versions of the Excel file can also be used as input and entries that have already been changed manually are added to the result file. Changes to the target structure can also be taken into account. Additional variables or new domains are simply inserted. The current target structure is always taken into account here and the order of the variables in the domain sheets is then automatically determined and new variables are placed in the correct position.

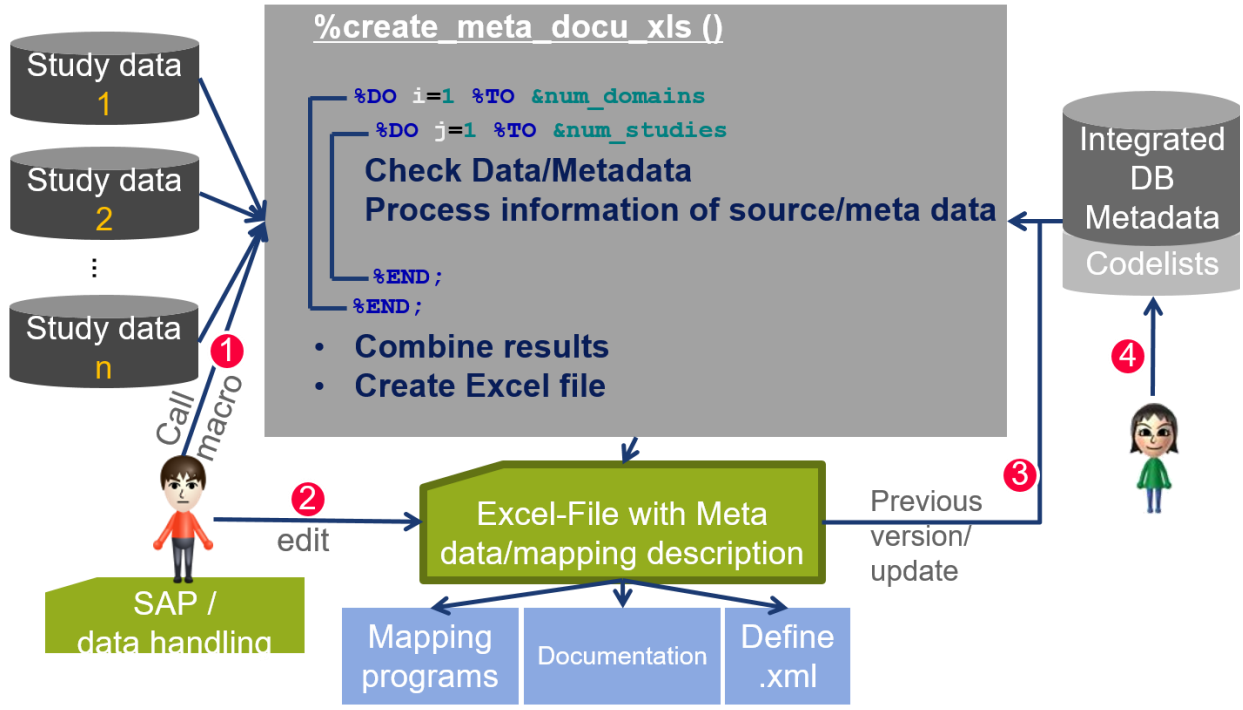


Fig. 6 Macro %create_meta_docu

In Tab. 1 we see the most important macro parameters of %create_meta_docu_xls. The project name *PROJECT* is shown in the Excel file. The library *LIBMETA* for the metadata of the IDB must be specified. The metadata defines the target structure. The various studies for which documentation is to be created are specified in the *STUDYID* parameter, separated by #. Using the same order, the *PATHSTUDYDATA* parameter is used to defined the paths to the source data of the individual studies, likewise separated by #. Either ADAM or SDTM can be entered via the parameter *STANDARD*. This is required for internal checks according to certain criteria with regard to the CDISC structure.

```
%macro create_meta_docu_xls(
       project                          =      /*Project name */
     , libmeta                          =      /*Library (name) to meta data*/
     , studyid                          =      /*<study 1>#<study 2>#<study 3>#...*/
     , pathstudydata                    =      /*<path 1>#<path 2>#<path 3>#... */
     , standard                         =ADAM  /*Standard SDTM|ADAM, default=ADAM*/
     , domains                          =      /*<domain 1>#>domain 2>#...*/
     , basedat                          =      /*Name of the ADSL or DM dataset*/
     , mapto                            =      /*Mapping definition */
     , pathoutfile                      =      /*path to output file */
     , outfilename                      =      /*name of output Excel file *.xlsx*/
     , previousfile                     =     /*name of previous mapping document */
     , includesuggestion                =N     /*mapping suggestions (Y/N)*/
     , suggestion_threshold             =75    /*threshold for suggestion costs*/
     ...
  /*further parameters for colours and standard text entries.*/
```

Tab. 1 Macro definition %create_meta_docu_xls

The *DOMAINS* parameter specifies a # delimited list of domains which should be displayed in the Excel file.

BASEDAT specifies which basic data set is to be taken into account in the macro. This may be necessary if variables such as populations or treatment groups have been added to the domains. These can then be filtered out in the macro or are then marked in the generated Excel file. The *BASEDAT* parameter can be specified separately for each study using #. Or it can be left blank, in which case either DM (for SDTM) or ADSL (for ADaM) is used.

The *MAPTO* parameter is important for the mapping assignment. The mapping from source to target is defined here. Each study can map single or multiple source domains to a target domain. Supplemental data (SUPP-data) from SDTM can also be entered here. SUPP-data is then transposed in the macro so that the corresponding generated variables can be displayed. These are processed in the macro and added to the source data of the respective domain. However, it is made clear that this variable comes from the respective SUPP domain. *MAPTO* is used as follows:

MAPTO = STUDYID1: (SOURCE_1 SOURCE_2 … SOURCE_n) | (SOURCE1 SUPPDOMAIN1) → TARGET_i # **/**

STUDYID2: (SOURCE_1 SOURCE_2 … SOURCE_n) | (SOURCE2 SUPPDOMAIN2) → TARGET_i # **/**

…

STUDYIDn: (SOURCE_1 SOURCE_2 … SOURCE_n) | (SOURCE3 SUPPDOMAIN3) → TARGET_i #

The mapping areas of the respective target domains are separated by the # symbol. The studies are indicated with $STUDYID_i$:. Each section on a study is followed by a forward slash. The assignment is called again in the macro and translated into the corresponding mapping in a do-loop. The structure of the source domains (and SUPP domains) are analysed and assigned depending on the target domain.

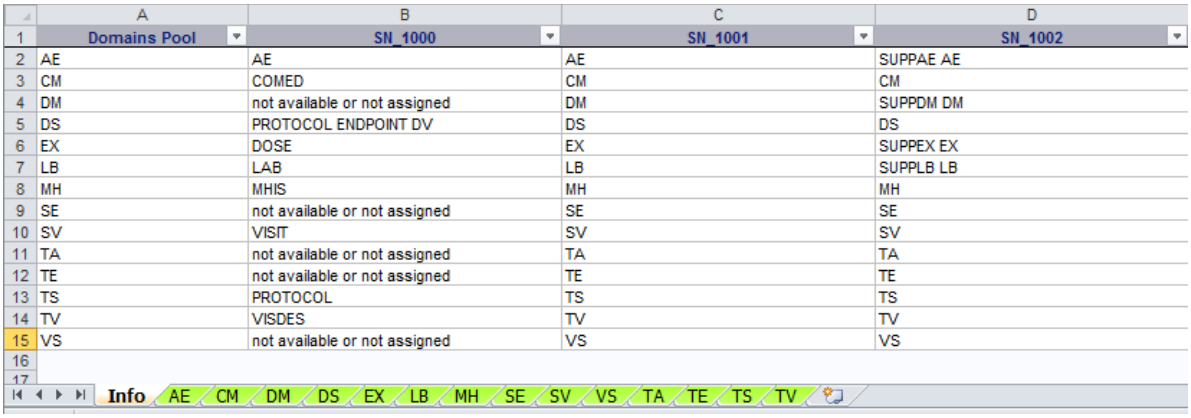
The following example (Tab. 2) shows a mapping assignment with three studies and multiple domains. The *MAPTO* parameter shows the various options available for mapping from source to target.

```
%create_meta_docu_xls(
  project       = Drugname
 ,libmeta       = meta
 ,studyid       = 1000#1001#1002
 ,pathstudydata = &source1 #&source2 #&source3
 ,basedat       = PATINFO#DM#DM#DM
 ,domains       = AE CM DM DS EX LB MH SE SV VS TA TE TE TS TV VS
 ,pathoutfile   = &outpath./docum
 ,outfilename   = drugname_mapping_documentation_&date._test.xlsx
 ,standard      = SDTM
 ,mapto         = 1000: COMED->CM#(PROTOCOL ENDPOINT DV)->DS#DOSE->EX#LAB->LB#
                        MHS->MH#
                        VISIT->SV#PROTOCOL->TS#VISDES->TV
                  /1001:
                  /1002: (SUPPAE AE)->AE#(SUPPDM DM)->DM#(SUPPEX EX)->EX#
                         (SUPPLB LB)->LB

);
```

Tab. 2 Example call %create_meta_docu_xls

The result of the mapping assignment can then be seen in the Excel file in the info sheet (see Fig. 7).



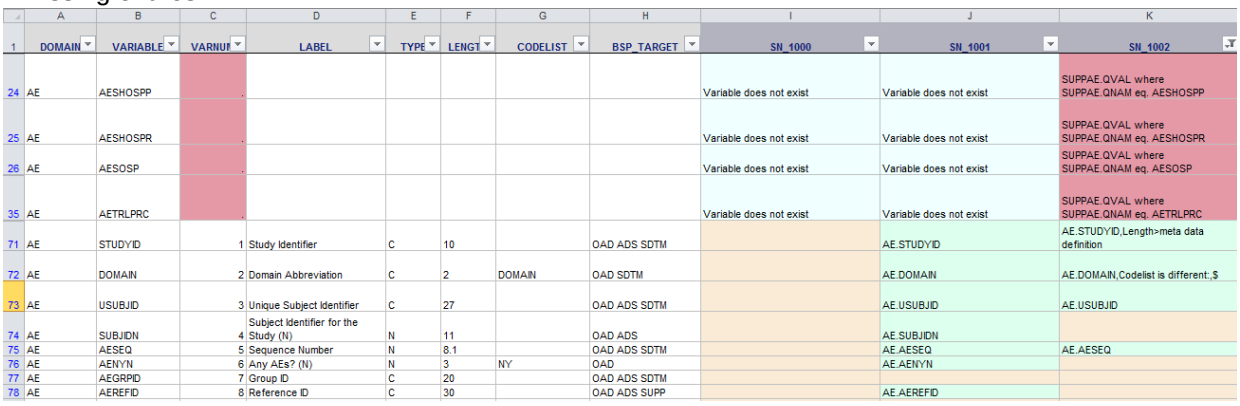
	A	B	C	D
1	Domains Pool	SN_1000	SN_1001	SN_1002
2	AE	AE	AE	SUPPAE AE
3	CM	COMED	CM	CM
4	DM	not available or not assigned	DM	SUPPDM DM
5	DS	PROTOCOL ENDPOINT DV	DS	DS
6	EX	DOSE	EX	SUPPEX EX
7	LB	LAB	LB	SUPPLB LB
8	MH	MHIS	MH	MH
9	SE	not available or not assigned	SE	SE
10	SV	VISIT	SV	SV
11	TA	not available or not assigned	TA	TA
12	TE	not available or not assigned	TE	TE
13	TS	PROTOCOL	TS	TS
14	TV	VISDES	TV	TV
15	VS	not available or not assigned	VS	VS
16				
17				

Fig. 7 Resulting Excel file with *Info* sheet and mapping assignment

The %create_meta_docu_xls macro then generates a sheet with the target structure (domain name, variable name, variable sequence number, label, type, length, codelist) for each domain called. A column is created for each study that shows the variables found or the actual mapping. Colours highlight different properties and information about the different variables in the studies:

- Green - found variable
- Beige - missing variable, mapping description must be added
- Red - variable does not belong to the target structure
- Light blue - the variable was not found, it is available in another study

Fig. 8 shows an example of an AE domain with different colours indicating features of the found, not found, or missing entries.

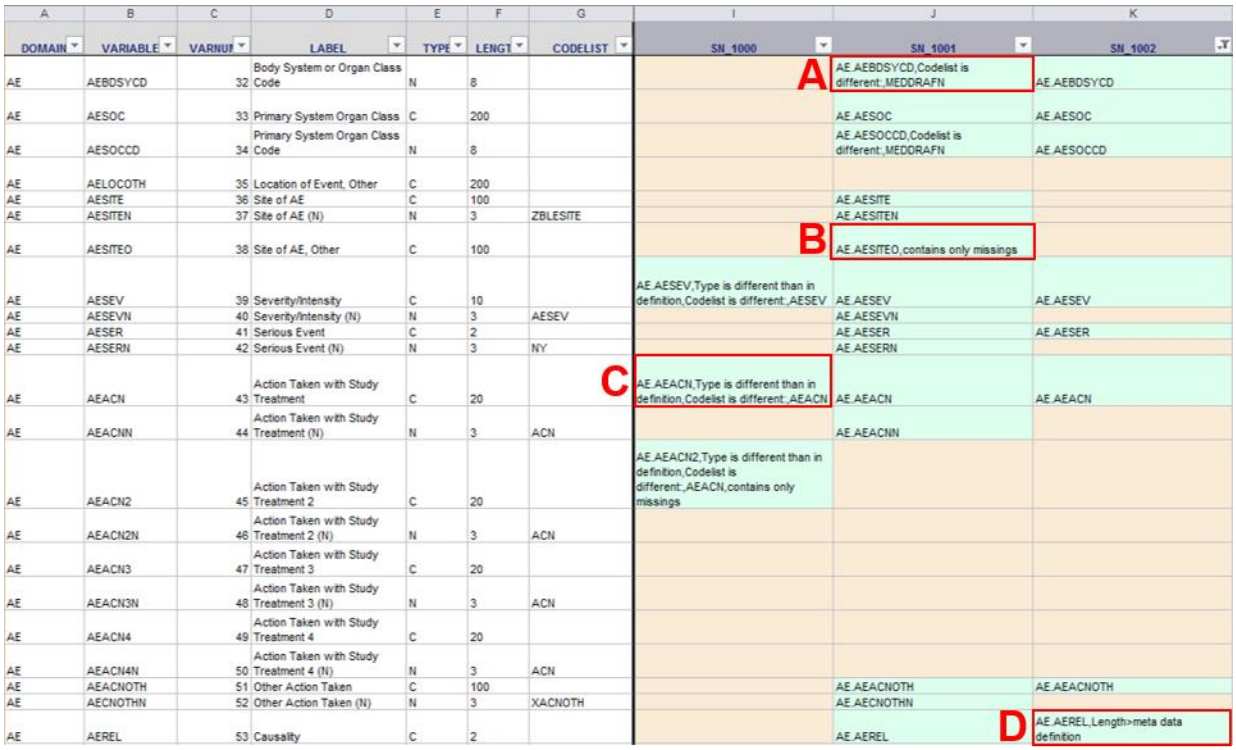


	A	B	C	D	E	F	G	H	I	J	K
1	DOMAIN	VARIABLE	VARNUM	LABEL	TYPE	LENGT	CODELIST	BSP_TARGET	SN_1000	SN_1001	SN_1002
24	AE	AESHOSPP							Variable does not exist	Variable does not exist	SUPPAE.QVAL where SUPPAE.QNAM eq. AESHOSPP
25	AE	AESHOSPR							Variable does not exist	Variable does not exist	SUPPAE.QVAL where SUPPAE.QNAM eq. AESHOSPR
26	AE	AESOSP							Variable does not exist	Variable does not exist	SUPPAE.QVAL where SUPPAE.QNAM eq. AESOSP
35	AE	AETRLPRC							Variable does not exist	Variable does not exist	SUPPAE.QVAL where SUPPAE.QNAM eq. AETRLPRC
71	AE	STUDYID	1	Study Identifier	C	10		OAD ADS SDTM		AE.STUDYID	AE.STUDYID,Length>meta data definition
72	AE	DOMAIN	2	Domain Abbreviation	C	2	DOMAIN	OAD SDTM		AE.DOMAIN	AE.DOMAIN,Codelist is different:,$
73	AE	USUBJID	3	Unique Subject Identifier	C	27		OAD ADS SDTM		AE.USUBJID	AE.USUBJID
74	AE	SUBJIDN	4	Subject Identifier for the Study (N)	N	11		OAD ADS		AE.SUBJIDN	
75	AE	AESEQ	5	Sequence Number	N	8.1		OAD ADS SDTM		AE.AESEQ	AE.AESEQ
76	AE	AENYN	6	Any AEs? (N)	N	3	NY	OAD		AE.AENYN	
77	AE	AEGRPID	7	Group ID	C	20		OAD ADS SDTM			
78	AE	AEREFID	8	Reference ID	C	30		OAD ADS SUPP		AE.AEREFID	

Fig. 8 AE domain with meta data structure and mapping columns per study

The generated Excel file is initially a draft mapping description, any missing information about the assignment - beige fields - must be entered manually. For additional variables found (red) that do not belong to the target structure, it must be decided whether they should be mapped or how they should be handled. For the cells marked in green, it is initially clear that the variable exists. If necessary, changes have to be made here, because sometimes the mapping for an IDB has to be harmonized. Additional information about the variable content helps the programmer to adapt the mapping if necessary.

Fig. 9 shows more details with examples of variables found and information about the content.



DOMAIN	VARIABLE	VARNUM	LABEL	TYPE	LENGT	CODELIST	SN_1000	SN_1001	SN_1002
AE	AEBDSYCD	32	Body System or Organ Class Code	N	8			A AE.AEBDSYCD,Codelist is different:,MEDDRAFN	AE.AEBDSYCD
AE	AESOC	33	Primary System Organ Class	C	200			AE.AESOC	AE.AESOC
AE	AESOCCD	34	Primary System Organ Class Code	N	8			AE.AESOCCD,Codelist is different:,MEDDRAFN	AE.AESOCCD
AE	AELOCOTH	35	Location of Event, Other	C	200				
AE	AESITE	36	Site of AE	C	100			AE.AESITE	
AE	AESITEN	37	Site of AE (N)	N	3	ZBLESITE		AE.AESITEN	
AE	AESITEO	38	Site of AE, Other	C	100			B AE.AESITEO,contains only missings	
AE	AESEV	39	Severity/Intensity	C	10		AE.AESEV,Type is different than in definition,Codelist is different:,AESEV	AE.AESEV	AE.AESEV
AE	AESEVN	40	Severity/Intensity (N)	N	3	AESEV		AE.AESEVN	
AE	AESER	41	Serious Event	C	2			AE.AESER	AE.AESER
AE	AESERN	42	Serious Event (N)	N	3	NY		AE.AESERN	
AE	AEACN	43	Action Taken with Study Treatment	C	20		C AE.AEACN,Type is different than in definition,Codelist is different:,AEACN	AE.AEACN	AE.AEACN
AE	AEACNN	44	Action Taken with Study Treatment (N)	N	3	ACN		AE.AEACNN	
AE	AEACN2	45	Action Taken with Study Treatment 2	C	20		AE.AEACN2,Type is different than in definition,Codelist is different:,AEACN,contains only missings		
AE	AEACN2N	46	Action Taken with Study Treatment 2 (N)	N	3	ACN			
AE	AEACN3	47	Action Taken with Study Treatment 3	C	20				
AE	AEACN3N	48	Action Taken with Study Treatment 3 (N)	N	3	ACN			
AE	AEACN4	49	Action Taken with Study Treatment 4	C	20				
AE	AEACN4N	50	Action Taken with Study Treatment 4 (N)	N	3	ACN			
AE	AEACNOTH	51	Other Action Taken	C	100			AE.AEACNOTH	AE.AEACNOTH
AE	AECNOTHN	52	Other Action Taken (N)	N	3	XACNOTH		AE.AECNOTHN	
AE	AEREL	53	Causality	C	2			AE.AEREL	D AE.AEREL,Length>meta data definition

Fig. 9 Matching Variables with further information

In Fig. 9 we see in A a variable AEBODSYCD that was found in Study 1001 but which was assigned a different codelist ("Codelist is different"). B shows AESITEO which only contains "Missings". Fig. 9 C shows a variable that has a different type ("Type is different than in definition"). The codelist is also different here. In Fig. 9 D the content of the variable is longer than in the metadata description. In addition, the variable label is checked against the label defined in the metadata structure of the IDB. This information is intended to support the programmer when creating the mapping programs. They give him or her helpful information about the variables and, in part, about their content.

In the following section we look at how the macro %create_meta_docu_xls makes mapping suggestions. The aim is to map the programmer-supported variables that were not previously found in the metadata structure.

Fuzzy Matching Techniques

The macro %create_meta_docu_xls can also make mapping suggestions if variables could not be automatically assigned to any variable in the IDB. In Fig. 10 we see an example of this in the area marked in red, where we see the variables BIRTH_D, BIRTH_M and BIRTH_Y. BIRTH_D is assigned to BRTH_D, BIRTH_M is assigned to BRTH_M, and BIRTH_Y is assigned to BRTH_Y. The target variables differ only slightly from the source variables, which could not be matched previously.

DOMAIN	VARIABLE	LABEL	TYP	LENG T	CODELI	BSP_TARGE T	SN_1000	SN_1001	SN_1002
DM	BIRTH_D						PATIENT.BIRTH_D , mapping suggestion: BRTH_D	Variable does not exist	Variable does not exist
DM	BIRTH_M						PATIENT.BIRTH_M , mapping suggestion: BRTH_M	Variable does not exist	Variable does not exist
DM	BIRTH_Y						PATIENT.BIRTH_Y , mapping suggestion: BRTH_Y	Variable does not exist	Variable does not exist
DM	BRTHDTC	Birth	C	19		SDTM		DM.BRTHDTC	DM.BRTHDTC
DM	BRTH_D	Day of Birth	N	2		ADS		DM.BRTH_D	
DM	BRTH_M	Month of Birth	N	2		ADS		DM.BRTH_M	
DM	BRTH_Y	Year of Birth	N	4		ADS		DM.BRTH_Y	

Fig. 10 Mapping suggestion

In the macro, the *INCLUDESUGGESTION* = Y parameter can be used to call the routine for mapping suggestions. In the case of variables that could not yet be assigned (marked in red in the Excel file), suggestions for the assignment are made using fuzzy matching techniques.
The macro uses various SAS functions such as COMPLEV (*Levenshstein distance*), SPEDIS (*spelling distance*) and SOUNDEX (*phonetic matching*) to calculate the distance measurements. The macro tries to assign all existing variables of a domain of the metadata structure that previously had no match to a variable that does not appear in the metadata. This is done via an n to n join. Then the SAS functions mentioned above are applied to the variable name and the label and these are compared with the variables of the metadata structure. The functions COMPLEV, SPEDIS and SOUNDEX return a corresponding value. This value can be viewed as a cost. In the macro, the results of these functions are summed up and a score is calculated. If the score is below a certain threshold value, a match is deemed to have occurred. The variable with the lowest score is then found for the match and entered in the Excel file as a "Mapping suggestion".

variable Variable Name	label1 Variable Label	SASNAME	suggest1	dist	dist2	dist3	sound1	sound2	sound_dist	type_cost	sound_cost	sum_costs	match	cost_min
BIRTH_D	Day of Birth	BRTH_D	BRTH_D	8	1	100	B633	B633	0	0	0	11	.	11
BIRTH_M	Month of Birth	BRTH_M	BRTH_M	8	1	100	B635	B635	0	0	0	11	.	11
BIRTH_Y	Year of Birth	BRTH_Y	BRTH_Y	8	1	100	B63	B63	0	0	0	11	.	11
RACEC	Race	RACE	RACE	12	1	10	R2	R22	1	0	2	19	.	19

Fig. 11 Fuzzy matching score example

Fig. 11 shows the results of the various fuzzy matching functions in a section as a data set. The summed up minimum is shown in the *cost_min* column. In this case, a suitable match was found, the variables to be examined are in *variable* and the proposed variables in *suggest1*. The variable *dist2* shows the Levenshtein distance with the respective values 1. In this case, only one letter in the search pattern has to be changed to attain the target word.
In Fig. 12 we see a simple example of the Levensthein distance. The pattern word “FORM” becomes the target word “FORK” by substituting a letter from M to K. In principle, this incurs a cost of 1, since a letter has to be replaced or changed. The Levenshtein distance indicates the amount of costs that arise in order to get from a search word to a target word, i.e. it indicates how many letters you have to change.

F	O	R	M	substitution
F	O	R	K	M to K

Fig. 12 Levenshtein distance – Example costs=1

The SPEDIS function, which specifies a type of probability, was also applied in the algorithm used. We also used the SOUNDEX function in the macro, which uses phonetic matching.

```
data suggestion&_std.&target.;
  set suggestion&_std.&target.;
  dist1=spedis(sasname, variable);
  dist2=complev(sasname, variable);
  sound_dist=complev(soundex(sasname), soundex(variable);
  dist3_label=complev(label,label&_std);
  dist4_label=spedis(label, label&_std);
  if (type='C' and type&_std.=2) or
     (type='N' and type&_std.=1)
  then type_cost=0; else type_cost=3;
  if soundex(sasname)=soundex(variable) then sound_cost=0;
  else sound_cost=2;
  sum_costs=sum(dist1,dist2,comp_,type_cost, sound_dist,
                dist3_label,dist4_label);
  if sasname=variable then match=1;
run;
```

Tab. 3 Fuzzy matching functions in a data step

Table 3 shows a small part of the fuzzy matching algorithm in a data step. Here we see the call of the SAS functions SPEDIS, COMPLEV and SOUNDEX on the variable names and labels. In addition, the type of the compared variable was used and given a score. All in all, everything is then summed up to form a score and the smallest value below a threshold value leads to a mapping proposal.

Adding Previous Mapping Documentation

In some cases it may be necessary to update existing mapping documentation.
Here you can call the macro %create_meta_docu_xls with an existing Excel file. The information already entered is transferred to the new document, and if there are new variables or new domains, these are inserted or recreated. The previous information is retained. New studies can also be easily added in the macro call. For the new study, a column is added for each domain in the resulting Excel document. The new study is also added to the info page with an additional column and the corresponding mapping information is displayed.

- Update of meta data of the IDB
- New study
- Changed mapping
- Update Excel-file documentation

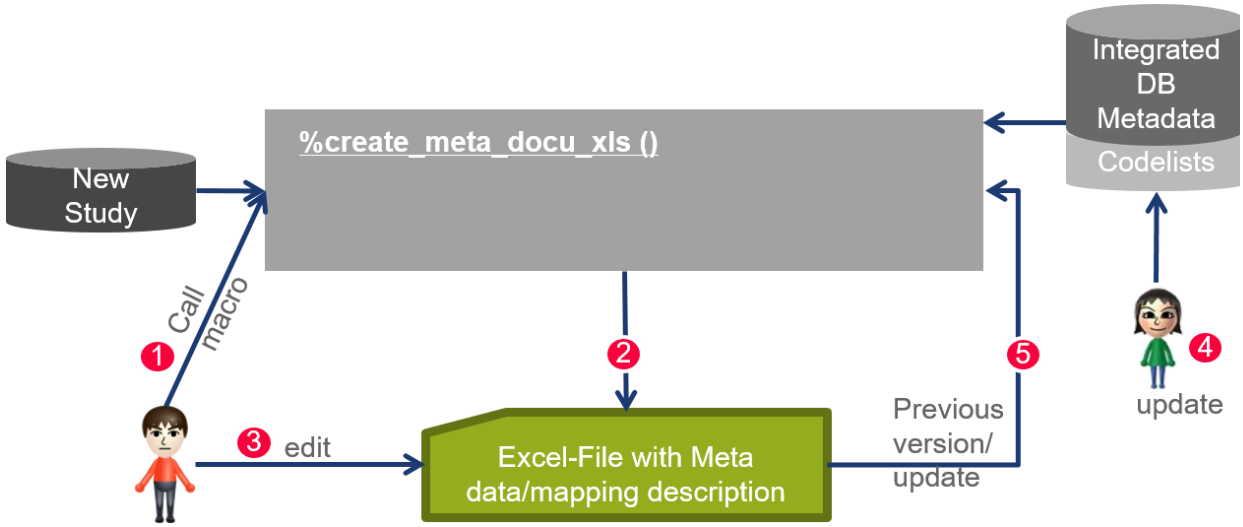
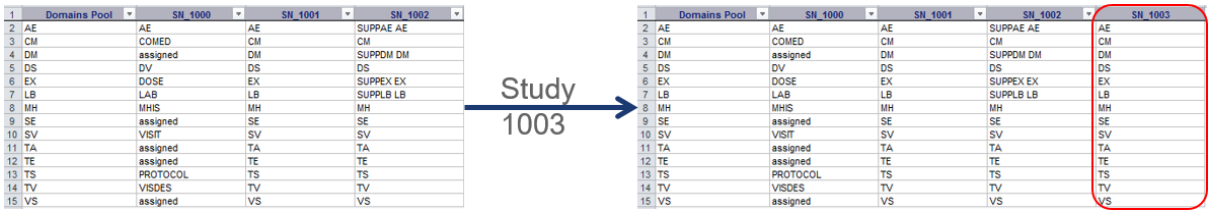


Fig. 13 Adding previous Excel file in macro %create_meta_docu_xls

Fig. 13 shows various options when mapping documentation should be updated.
If, for example, a new study is to be mapped in the IDB, the previous macro call is simply extended. This scenario can be seen in Fig. 14.



	Domains Pool	SN_1000	SN_1001	SN_1002
2	AE	AE	AE	SUPPAE AE
3	CM	COMED	CM	CM
4	DM	assigned	DM	SUPPDM DM
5	DS	DV	DS	DS
6	EX	DOSE	EX	SUPPEX EX
7	LB	LAB	LB	SUPPLB LB
8	MH	MHIS	MH	MH
9	SE	assigned	SE	SE
10	SV	VISIT	SV	SV
11	TA	assigned	TA	TA
12	TE	assigned	TE	TE
13	TS	PROTOCOL	TS	TS
14	TV	VISDES	TV	TV
15	VS	assigned	VS	VS

	Domains Pool	SN_1000	SN_1001	SN_1002	SN_1003
2	AE	AE	AE	SUPPAE AE	AE
3	CM	COMED	CM	CM	CM
4	DM	assigned	DM	SUPPDM DM	DM
5	DS	DV	DS	DS	DS
6	EX	DOSE	EX	SUPPEX EX	EX
7	LB	LAB	LB	SUPPLB LB	LB
8	MH	MHIS	MH	MH	MH
9	SE	assigned	SE	SE	SE
10	SV	VISIT	SV	SV	SV
11	TA	assigned	TA	TA	TA
12	TE	assigned	TE	TE	TE
13	TS	PROTOCOL	TS	TS	TS
14	TV	VISDES	TV	TV	TV
15	VS	assigned	VS	VS	VS

Fig. 14 Adding new study “1003” to the mapping documentation

The corresponding macro call is shown in Tab. 5.

```
%create_meta_docu_xls(
     project       = Drugname
    ,libmeta       = meta
    ,studyid       = 1000#1001#1002#1003
    ,pathstudydata = &source1 #&source2 #&source3#&source4
    ,inbase        = patinfo#dm#dm#dm
    ,domains       = ae cm dm ds ex lb mh se sv vs ta te ts tv vs
    ,pathoutfile   = &outpath./docum
    ,outfilename   = drugname_mapping_documentation_&date._test.xlsx
    ,standard      = SDTM
    ,previousfile  = &pathto_excel_documentation/project_drugname_2021-06-10.xlsx
    ,mapto         = 1000:  COMED->CM#(PROTOCOL ENDPOINT DV)->DS#DOSE->EX#LAB->LB#MHS->MH#VISIT->SV# PROTOCOL->TS#VISDES->TV
                     /1001:
                     /1002: (SUPPAE AE)->AE#(SUPPDM DM)->DM#(SUPPEX EX)->EX#(SUPPLB LB)->LB
                     /1003:  <add mapping if necessary>
);
```

Tab. 5 Adding new study “1003” to macro call

In the macro call, only the STUDYID 1003 is added and another entry is made for the *MAPTO* parameter. If there is different source data for a certain domain in Study 1003, these would also have to be entered in *MAPTO*. Otherwise the algorithm uses a 1:1 mapping. The macro assumes that the source and target domains are always the same. This can be seen in Figure 8. Here the column with STUDYID = “1003“ is marked red and we see the same domains as in the target domain.

Fig. 15 shows a scenario where an existing Excel documentation is used as an input file in the macro call. After the programmer has edited the Excel documentation manually and has made missing entries, he or she calls the macro. The *PREVIOUSFILE* parameter is filled with the file and the macro is run.

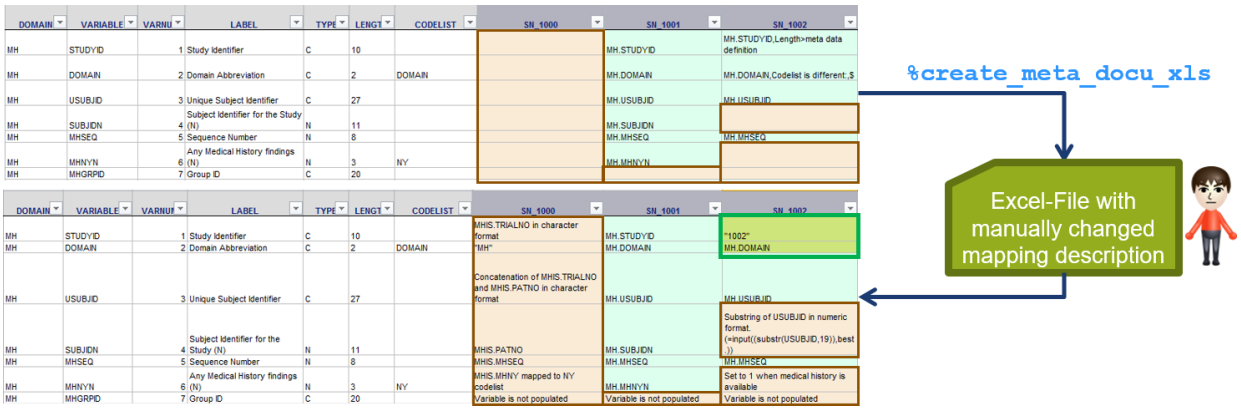


Fig. 15 Add manually changed Excel-documentation with macro call

The macro uses the information from *PREVIOUSFILE* and reads it in and performs a join with the data created from the macro run. In the beige fields we see the transferred entries, the light green fields are also retained because the variables have already been found here. If a "green" entry has been changed manually, this is marked with a darker green colour. In this case the STUDYID and DOMAIN variables were changed by the programmer and the different colour indicates the change.

Summary

We have seen that with the help of %create_meta_docu_xls, the documentation of mapping descriptions for integrated databases (IDBs) can be easily created. The macro has the following features and advantages:

Features of %create_meta_docu_xls

- Processes any source, SDTM or ADaM
- Multiple studies
- Different source domains to one target domain (source1…source N) → target
- Adds information from SUPP-domains
- Adds information about data / variables
- Adds mapping suggestions, variables found
- Processes a previously created Excel document

- Handles updated source / target metadata
- Adds colours to support users
- Could be used as mapping specification (draft) / define.xml / documentation

We would like to emphasize the ease of use of the macro and its flexibility. Changes in the metadata are simply inserted automatically when %create_meta_docu_xls is run. Previous versions of documentation can also be reused; the macro inserts the information about this into the new version. New studies can also be easily added to the documentation using the macro call. Finally, the Excel document can also be used to create a define.xml.
For the future, we could add standard entries that appear regularly in the documentation and the mapping programs to the Excel document via another source. This would make the work of the programmer even easier.
The programmer is supported in the documentation of the mapping and does not have to worry about manual changes in the metadata. This is done simply with the macro %create_meta_docu_xls.

ACKNOWLEDGMENTS
I would like to thank **Rowland Hale** (Senior Principal Statistical Programmer, Syneos Health) for his great support and review of this manuscript.

CONTACT INFORMATION
Your comments and questions are valued and encouraged. Contact the author at:

Thomas Wollseifen
Syneos Health Germany GmbH
Stefan-George-Ring 6
81929 München, Germany
Email: Thomas.wollseifen@syneoshealth.com
Web: https://www.syneoshealth.com/