

Pooled Database Loader (PDL)

**Combining documentation and code
to create a pooled database**

US Connect PhUSE 2018



Agenda

The Introduction

The Components

- The interface
- The call

The Procedure

- Step 1: defining the list of studies and where the data is located (studylib.xlsx)
- Step 2: defining the list of datasets to be pooled (domainmapv1.xlsx)
- Step 3: defining the code list for the pooled database (pdb_codelist.xlsx)
- Step 4: defining the mapping rules for each study to be pooled (*DOMAIN*domain_var_mapv1.xlsx)
- Step 5: loading the study data into the pooled database

The Demonstration

The Conclusion

Questions

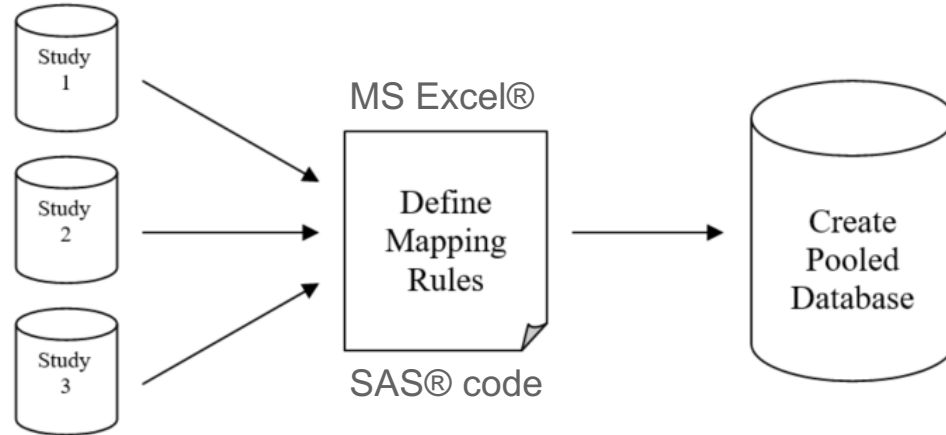
The Introduction

PB&J



The Introduction

PDL is an application where two or more studies within a drug project can be combined into one database



Components: the interface

PDL consists of 4 Excel Spreadsheets:

1. STUDYLIB.xlsx – contains the list of studies and the paths to where their data is located
2. DOMAINMAPv1.xlsx – contains the list of studies and the datasets that exist for each study
3. PDB_CODELIST.xlsx – contains the list of study and pooled level codes and decodes to be used for remapping of data as needed.
4. *DOMAINDOMAIN*_var_mapv1.xlsx – contains the list of variables and their attributes that will be included in the pooling of the data. It is here that work of mapping takes place (ex. ADSLdomain_var_mapv1.xlsx).

Components: the call

There are 8 parameters included in the call to PDL:

1. LOADFUNC - *defines a specific task to be performed*
2. PDBPATH - *path to folder where the pooled datasets are to be stored*
3. XLSPATH - *path to spreadsheets*
4. PGMPATH - *path to customized functions*
5. LOADMAC - *list of customized functions to be loaded for algorithms*
6. DDT - *populates the “template” section of the VARMAP spreadsheet*
7. DOMLST - *list of domains to be mapped and loaded into project database*
8. SASOPTS - *valid SAS options*

Components: the call

There 4 tasks that are performed by PDL through the **LOADFUNC** parameter:

1. DOMAINMAP - produces the domain_mapvn.xlsx spreadsheet
2. VARMAP - produces the *DOMAINdomain_var_mapvn.xlsx spreadsheet*
3. FMTLOAD - using the pdb_codelist.xlsx spreadsheet, the project format and recode definitions are loaded into a SAS dataset and stored in the project database.
4. DTALOAD - using the studylib.xlsx and *DOMAINdomain_var_mapvn.xlsx* spreadsheets the study level data is mapped into the project standard structure and loaded into the project database.

The Procedure

Step 1: the list of studies the location of data

studylib.xlsx

	A	B	C	D	E
1	studyid	study	index	studylib	load
2	Study1	First Study		X:\study1\ad\data	
3	Study2	Second Study	sd	X:\study2\sd\data	
4	Study2	Second Study	ad	X:\study2\ad\data	
5	Study3	Third Study		X:\study3\ad\data	
6	Study4	Fourth Study		X:\study4\ad\data	

Column A: study identifiers used in the pooling of the data

Column B: study description used for those studies that have an acronym in addition to the ID

Column C: index which allows for study level data to come from 2 or more different locations

Column D: path to where the study level data is stored

Column E: with an 'x' the user can dictate to PDL what studies are to be loaded into the pooled database

The Procedure

Step 2a: Domain mapping

The DOMAINMAPv1.xlsx is created in this step using the call to PDL.

```
%pdblload(loadfunc=domainmap ,  
.  
.)
```

domain_mapv1.xlsx

	A	B	C	D	E
1	PRJDOMAIN	STDYStudy1	STDYStudy2	STDYStudy3	STDYStudy4
2		ADAE	AE_Ssd	ADAE	ADAE
3		ADAUEC	CE_Ssd	ADBILAG	ADBILAG
4		ADCM	CM_Ssd	ADCM	ADCM
5		ADCO	ADAE_Sad	ADDV	ADCSSRS
6		ADCP	ADAUEC_Sad	ADEG	ADDV

When creating the spreadsheet, PDL attaches a version number preceded by the letter “v”. The number is incremented each time DOMAINMAP is run.

The Procedure

Step 2b: Domain mapping

- Map study level datasets to pooled datasets.
- Insert a new row for each pooled data set to be created.
- The dataset names are copied from the studies to the pooled level.

domain_mapv1.xlsx

	A	B	C	D	E
1	PRJDOMAIN	STDYStudy1	STDYStudy2	STDYStudy3	STDYStudy4
2	ADSL	ADSL	ADSL_Sad DM_Ssd	ADSL	ADSL
3	ADLB	ADLB	ADLB_Sad	ADLB	ADLBCEM ADLBHEMA ADLBOTH ADLBURIN
4		ADAE	AE_Ssd	ADAE	ADAE

The Procedure

Step 3a: Pooled database code list

- Prior to defining the mapping rules in step 4, the pdb_codelist spreadsheet is created.
- The formats defined in the “pooled” tab can be applied across all of the studies in the pool.

pdb_codelist.xlsx

	A	B	C
1	format name	value	label
2	sexnf		1 Male
3			2 Female
4			3 Missing
5			
6	\$sexs	Male	M
7		Female	F

The Procedure

Step 3b: Pooled database code list

- Unlike the “pooled” tab, the formats defined in the study tabs are only applied to the study specified by the tab.

pdb_codelist.xlsx

	A	B	C	D	E
1	recode fmt	old value	old label	new value	new label
2	trt01af		1 Dose 1		2 Low Dose
3			2 Placebo		1 Placebo
4			9 Screen Failure		0 Screen Failure
5					
6	\$racef	WHITE			5
7		OTHER/MIXED			6

- Columns C (old label) and E(new label) provide additional documentation for historical purposes.

The Procedure

Step 3c: Pooled database code list

- PDL stores the “pooled” formats as a dataset when using FMTLOAD.

```
%pdbload(loadfunc=fmtload ,  
          .  
          .)
```

- This function is run prior to loading the study level data and is used as defined in the mapping process.
- At the time of loading the study level data, the format dataset is converted to a format catalog.

The Procedure

Step 4a: variable mapping

- Creating the variable mapping spreadsheet using the VARMAP function.

```
%pdbload(loadfunc=varmap ,
          domlst=adsl,
          .
          )
```

ADSLdomain_var_mapv1.xlsx

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	PRJDOMAIN	PRJVAR	PRJTP	PRJLEN	PRJLBL	PRJFMT	PRJKEY	STDTAStudy1	STVARStudy1	STTYPStudy1	STLENStudy1	STLBLStudy1	STFMTStudy1
2	ADSL							ADSL	ACTARM	char	200	Description of Actual Arm	
3	ADSL							ADSL	ACTARMCD	char	20	Actual Arm Code	
4	ADSL							ADSL	AGE	num	8	Age	BEST12.
5	ADSL							ADSL	AGECAT	char	200	Age Category	
6	ADSL							ADSL	AGEU	char	6	Age Units	
7	ADSL							ADSL	AGE_CALC	num	8	Age calculated	BEST12.
8	ADSL							ADSL	AGGDRC	char	200	Gender, Age, Race, Weight(kg) combined	
9	ADSL							ADSL	AGGRC	char	200	Gender, Age, Race combined	
10	ADSL							ADSL	ARM	char	200	Description of Planned Arm	

The Procedure

There are 2 primary sections to the VARMAP spreadsheet

Template

Study

Template								Study						
	A	B	C	D	E	F	G	H	I	J	K	L		M
1	PRJDOMAIN	PRJVAR	PRJTYP	PRJLEN	PRJLBL	PRJFMT	PRJKEY	STDTAStudy1	STVARStudy1	STTYPStudy1	STLENStudy1	STLBLStudy1		STFMTStudy1
2	ADSL							ADSL	ACTARM	char	200	Description of Actual Arm		
3	ADSL							ADSL	ACTARMCD	char	20	Actual Arm Code		
4	ADSL							ADSL	AGE	num	8	Age		BEST12.
5	ADSL							ADSL	AGECAT	char	200	Age Category		
6	ADSL							ADSL	AGEU	char	6	Age Units		
7	ADSL							ADSL	AGE_CALC	num	8	Age calculated		BEST12.
8	ADSL							ADSL	AGGDRC	char	200	Gender, Age, Race, Weight(kg) combined		
9	ADSL							ADSL	AGGRC	char	200	Gender, Age, Race combined		
10	ADSL							ADSL	ARM	char	200	Description of Planned Arm		

The Procedure

Step 4b: variable mapping, populating the template

There are 3 options for populating the template:

Option 1 - Manual entry

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	PRJDOMAIN	PRJVAR	PRJTYP	PRJLEN	PRJLBL	PRJFMT	PRJKEY	STDTASTudy1	STVARStudy1	STTYPStudy1	STLENStudy1	STLBLStudy1	STFMTStudy1
2	ADSL	STUDYID	char	200	Study Identifier		1						
3	ADSL	USUBJID	char	200	Unique Subject Identifier		2						
4													
5	ADSL							ADSL	ACTARM	char	200	Description of Actual Arm	
6	ADSL							ADSL	ACTARMCD	char	20	Actual Arm Code	

The Procedure

Step 4b: variable mapping, populating the template

Option 2 - Metadata Collector

- The metadata collector is a UCB constructed spreadsheet that contains (in addition to some other information) a list of datasets as well as the variables that are contained within them.
- Most of the information contained in the metadata collector comes directly from the AdAM Implementation Guide.

ucb-metadata-collector-2017Q4.xlsx

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
	studyid/ pool an analysis/ project	level of standard	CURRENT STANDARD @UCB	ADaM type	ADaM	Parameter Identifier	Variable	Label	Key	Type	Length/ Display Format	Controlled Terms or Format	Origin	Role	Source/ Derivation/ Comment	Derivation Name	UCB core	flag
1																		
2	UCB	CDISC	Y	adsl	adsl	_all_	STUDYID	Study Identifier	1	text			SDTM	Study Identifier	Predecessor: DM.STUDYID		Req	1
3	UCB	CDISC	Y	adsl	adsl	_all_	USUBJID	Unique Subject Identifier	2	text			SDTM	Study Identifier	Predecessor: DM.USUBJID		Req	1
4	UCB	CDISC	Y	adsl	adsl	_all_	SUBJID	Subject Identifier for the Study		text			SDTM	Study Identifier	Predecessor: DM.SUBJID		Req	1
5	UCB	CDISC	Y	adsl	adsl	_all_	SITEID	Study Site Identifier		text			SDTM	Study Identifier	Predecessor: DM.SITEID		Req	1

version history | datasets | adam | tfl | ucb-ct | lists | column legend

```
%pdbload(loadfunc=varmap ,
           domlst=adsl,
           ddt=x:\pool\spreadsheet\ucb-metadata-collector-2017Q4.xlsx
           )
```

The Procedure

Step 4b: variable mapping, populating the template

Option 2 - Metadata Collector (continued)

ADSLdomain_var_mapv1.xlsx

	A	B	C	D	E	F	G
1	PRJDOMAIN	PRJVAR	PRJTYP	PRJLEN	PRJLBL	PRJFMT	PRJKEY
2	ADSL	STUDYID	char	200	Study Identifier		1
3	ADSL	USUBJID	char	200	Unique Subject Identifier		2
4	ADSL	SUBJID	char	200	Subject Identifier for the Study		
5	ADSL	SITEID	char	200	Study Site Identifier		
6	ADSL	REGIONy	char	200	Geographic Region y		
7	ADSL	REGIONyN	num	8	Geographic Region y (N)		
8	ADSL	ACOUNTRY	char	200	Analysis Country		
9	ADSL	AGE	num	8	Age		
10	ADSL	AGEU	char	200	Age Units		

PRJLEN is populated by PDL. If a variable is character then the length is defined as 200. If the variable is numeric then the length is defined as 8.

The Procedure

Step 4b: variable mapping, populating the template

Option 3 – Existing dataset

```
%pdbload(loadfunc=varmap ,  
          domlst=adsl,  
          ddt= library=x:\study1\data\adam * file= adsl  
          )
```

	A	B	C	D	E	F	G
1	PRJDOMAIN	PRJVAR	PRJTYP	PRJLEN	PRJLBL	PRJFMT	PRJKEY
2	ADSL	STUDYID	char	10	Study Identifier		
3	ADSL	USUBJID	char	20	Unique Subject Identifier		
4	ADSL	SUBJID	char	10	Subject Identifier for the Study		
5	ADSL	SITEID	char	5	Study Site Identifier		
6	ADSL	AGECAT	char	200	Age Category		
7	ADSL	COUNTRY	char	3	Country	\$20.	
8	ADSL	AGE	num	8	Age	BEST12.	

The Procedure

Step 4c: variable mapping, populating the study section

The study level mapping section contains 6 columns with each column header containing a 5 letter prefix followed by the study ID.

ADSLdomain_var_mapv1.xlsx

	B	G	H	I	J	K	L	M
1	PRJVAR	PRJKEY	STDTAStudy1	STVARStudy1	STTYPStudy1	STLENStudy1	STLBLStudy1	STFMTStudy1

STDTA – dataset name

STVAR – variable name

STTYP – variable type (char,num)

STLEN – variable length

STLBL – variable label

STFMT – variable format

The Procedure

Step 4c: variable mapping, populating the study section

Defining the study level mapping rules.

- If a study level variable is of the same name as the template variable, PDL will automatically populate the template level rows under that study.

ADSLdomain_var_mapv1.xlsx

	B	G	H	I	J	K	L	M
1	PRJVAR	PRJKEY	STDTAStudy1	STVARStudy1	STTYPStudy1	STLENStudy1	STLBLStudy1	STFMTStudy1
2	STUDYID	1	ADSL	STUDYID	char	10	Study Identifier	
3	USUBJID	2	ADSL	USUBJID	char	20	Unique Subject Identifier	
4	SUBJID		ADSL	SUBJID	char	10	Subject Identifier for the Study	
5	SITEID		ADSL	SITEID	char	5	Study Site Identifier	
6	REGIONy							

- The automatic populating of the study level information occurs when a study is loaded into the variable mapping spreadsheet for the first time.

The Procedure

Step 4c: variable mapping, populating the study section

Copying and pasting of variables from study level variable list.

- If a study level variable contains the information needed for a pooled variable and the variable names are not the same then copying and pasting is done.

ADSLdomain_var_mapv1.xlsx

	B	E	H	I	J	K	L	M
1	PRJVAR	PRJLBL	STDASTudy1	STVARStudy1	STTYPStudy1	STLENStudy1	STLBLStudy1	STFMTStudy1
12	AGE	Age	ADSL	AGE	num	8	Age	BEST12.
13	AGEU	Age Units	ADSL	AGEU	char	6	Age Units	
14	AGEGR1	Pooled Age Group 1	ADSL	AGECAT	char	200	Age Category	

The Procedure

Step 4c: variable mapping, populating the study section

If computing of values are required, this can be done in 2 ways:

1. **Variable level** – derivations are computed in the last dataset for each study just prior to loading. PDL provides 2 functions (macro programs) for this.

RECODE computes new values through the use of the formats.

ADSLdomain_var_mapv1.xlsx

	A	B	H	I	J	K	L	M
1	PRJDOMAIN	PRJVAR	STDTAStudy1	STVARStudy1	STTYPStudy1	STLENStudy1	STLBLStudy1	STFMTStudy1
8	ADSL	AGE	ADSL	AGE	num	8 Age		BEST12.
9	ADSL	AGEU	ADSL	AGEU	char	6 Age Units		
10	ADSL	AGEGR1	ADSL					
11	ADSL	AGEGR1N	ADSL	*recode(agegr1n,age,agegr1nf.)	num			

The purpose of “*” at the beginning of RECODE is to prevent SAS from executing this macro until the appropriate time.

The Procedure

Step 4c: variable mapping, populating the study section

DERIVE computes values for new variables. PDL simply takes whatever condition is defined and sets it equal to the variable that is being created.

ADSLdomain var mapv1.xlsx

	A	B	H	I	J	K	L	M
1	PRJDOMAIN	PRJVAR	STDTAStudy1	STVARStudy1	STTYPStudy1	STLENStudy1	STLBLStudy1	STFMTStudy1
2	ADSL	STUDYID	ADSL	STUDYID	char	10	Study Identifier	
3	ADSL	USUBJID	ADSL	USUBJID	char	20	Unique Subject Identifier	
4	ADSL	SUBJID	ADSL	SUBJID	char	10	Subject Identifier for the Study	
5	ADSL	SITEID	ADSL	SITEID	char	5	Study Site Identifier	
6	ADSL	TEST	ADSL	*derive(test,'This is a cool tool.')	char			
7	ADSL	ACOUNTRY						
8	ADSL	AGE	ADSL	AGE	num	8	Age	BEST12.
9	ADSL	AGEU	ADSL	AGEU	char	6	Age Units	

Like RECODE, the ‘*’ at the beginning of DERIVE is to prevent SAS® from executing this macro until the appropriate time.

The Procedure

Step 4c: variable mapping, populating the study section

2. **Data step level** – derivations are computed prior to the last data step of creating the pooled dataset for each study.
 - At the data step level one or more data steps can be written to perform the necessary computations for one or more variables to be included in the pool.
 - SAS® code can be written within a cell in a spreadsheet or as a separate program that can be called by PDL.
 - The data step programs are defined in the VARMAP spreadsheet using FUNCTION which is added in column A after the last pool variable in that column.
 - Whether the data step is defined in the spreadsheet or as an external program, the FUNCTION is defined in the STDTA column for each study.

The Procedure

Step 4c: variable mapping, populating the study section

Defining a data step in the spreadsheet

ADSLdomain_var_mapv1.xlsx

	A	B	H	I	J	K	L	M
1	PRJDOMAIN	PRJVAR	STDTAStudy1	STVARStudy1	STTYPStudy1	STLENStudy1	STLBLStudy1	STFMTStudy1
6	ADSL	AGE	ADSL	AGE	num	8	Age	BEST12.
7	ADSL	AGEU	ADSL	AGEU	char	6	Age Units	
8	FUNCTION		data fun; set adsl; test='This is a cool tool'; run;					

The Procedure

Step 4c: variable mapping, populating the study section

Defining a data step as a macro program.

ADSLdomain_var_mapv1.xlsx

	A	B	H	I	J	K	L	M
1	PRJDOMAIN	PRJVAR	STDASTudy1	STVARStudy1	STTYPStudy1	STLENStudy1	STLBLStudy1	STFMTStudy1
5	ADSL	SITEID	ADSL	SITEID	char	5	Study Site Identifier	
6	ADSL	TEST	ADSL	*derive(test,'This is a cool tool.')	char			
7	ADSL	ACOUNTRY						
8	ADSL	AGE	ADSL	AGE	num	8	Age	BEST12.
9	ADSL	AGEU	ADSL	AGEU	char	6	Age Units	
10	FUNCTION		%fun_tool					

The Procedure

Step 4c: variable mapping, populating the pooled section

Defining a data step as a macro program at the pooled level.

ADSLdomain_var_mapv1.xlsx

	A	B	C	D	E	F	G
1	PRJDOMAIN	PRJVAR	PRJTYP	PRJLEN	PRJLBL	PRJFMT	PRJKEY
71	ADSL	PROTDATE	char		Protocol date subject enrolled under		
72	ADSL	PROTVERS	char		Protocol version		
73	FUNCTION	data adsl; set adsl; newvar='This tool is cool'; run;					

The Procedure

Step 5: data loading

- This is the final step in the process where PDL executes the information contained in the VARMAP spreadsheet and builds the pooled datasets.
- The PDL loadfunc task is DTALOAD

```
%pdbload(loadfunc=dtaload ,  
          domlst=adsl,  
          pgmpath=path to where the custom pooling macros are stored,  
          loadmac=list of programs to be included in the pooling process  
          )
```

Demonstration

The Conclusion

- The Pooled Database Loader (PDL) is an alternative to the commonly used method of pooling data which is done by simply writing SAS® code.
- PDL combines the documentation found in 4 spreadsheets with SAS® code to pool any kind of data of any structure.
- As you move from one spreadsheet to the next you can obtain more detailed information from the highest level where you can see what studies were included in the pool and where that data comes from (studylib.xlsx), to what study level datasets were used for each pooled dataset (domainmapv1.xlsx), to the details of what variables were included and how they were mapped (*DOMAINdomain_var_mapv1.xlsx*).
- With the mapping completed and documented, the call to PDL takes over and loads the data and finishes the job of pooling.

Questions?

Thanks!