



The data listing facility: Data is just a few clicks away



Luigi De Vittorio, OCS Consulting, Netherlands

Index



- ▶ Introduction.
- ▶ Structure and functioning.
- ▶ Input and Output.
- ▶ Execution.
- ▶ A look at the code: what happens in the background?
- ▶ Conclusion.



Index



- ▶ Introduction.
- ▶ Structure and functioning.
- ▶ Input and Output.
- ▶ Execution.
- ▶ A look at the code: what happens in the background?
- ▶ Conclusion.



Introduction



- ▶ Data listings (with tables and figures) are part of the documentation to generate, review and validate for a clinical trial.
- ▶ They present a picture of the data collected for a clinical trial to correct issues and assuring that final data has the highest quality possible.
- ▶ "Data collected during a clinical trial must have as few errors as possible to be able to support the findings or conclusions drawn from that trial."

Introduction



- ▶ The data listing facility is a tool that was implemented within the SAS® Drug Development (SDD 3.5) environment of a pharmaceutical company.
- ▶ It allows to interactively generate data listings for clinical trials on vaccines studies.
- ▶ The tool uses only SAS programs.



Index



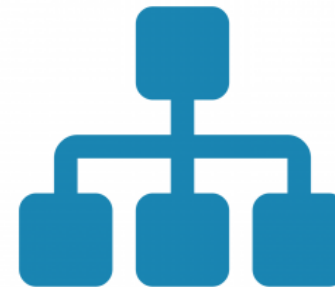
- ▶ Introduction.
- ▶ Structure and functioning.
- ▶ Input and Output.
- ▶ Execution.
- ▶ A look at the code: what happens in the background?
- ▶ Conclusion.



Structure and functioning



- ▶ Four components can be identified within the tool:
 - ▶ Program dl__userinterface.sas
 - ▶ Program dl__driver.sas
 - ▶ Listings programs
 - ▶ Macro programs



Structure and functioning



- ▶ dl__userinterface.sas
 - ▶ Main component of the tool.
 - ▶ Program that the user can execute in SDD with the “Job Results Viewer” functionality.
 - ▶ It generates and presents a user interface to the user to specify parameters for the execution.

- ▶ dl__driver.sas
 - ▶ Program included by dl__interface.sas after the parameters have been selected:
 - ▶ Includes all macros;
 - ▶ Sets up the libnames and filenames;
 - ▶ Executes the listing program(s) previously selected;
 - ▶ Generates the results of the execution;
 - ▶ Sends the results via email to the user.

Listings programs



- ▶ The core of the tool.
- ▶ Collection of SAS programs, each program is developed to generate a specific listing.
- ▶ Executed by the tool in series (multiple listings can be run during one execution).
- ▶ Two separate categories of listings: general listings (so called standard listings) and study-specific listings.



Standard VS Study Specific



- ▶ General listings:
 - ▶ Can be executed on every study;
 - ▶ Code must be as generalized as possible and must handle differences within studies (data structure);
 - ▶ Use of macro programs makes development easier;
 - ▶ No need to worry about re-use or adapt of code for other studies.

- ▶ Study-specific listings:
 - ▶ Intended to be executed for one study only;
 - ▶ Code is less generalized and can be easier to develop (e.g. hardcoding is allowed);
 - ▶ Code is study related, code re-using for another study can generate unexpected issues (should be performed carefully).

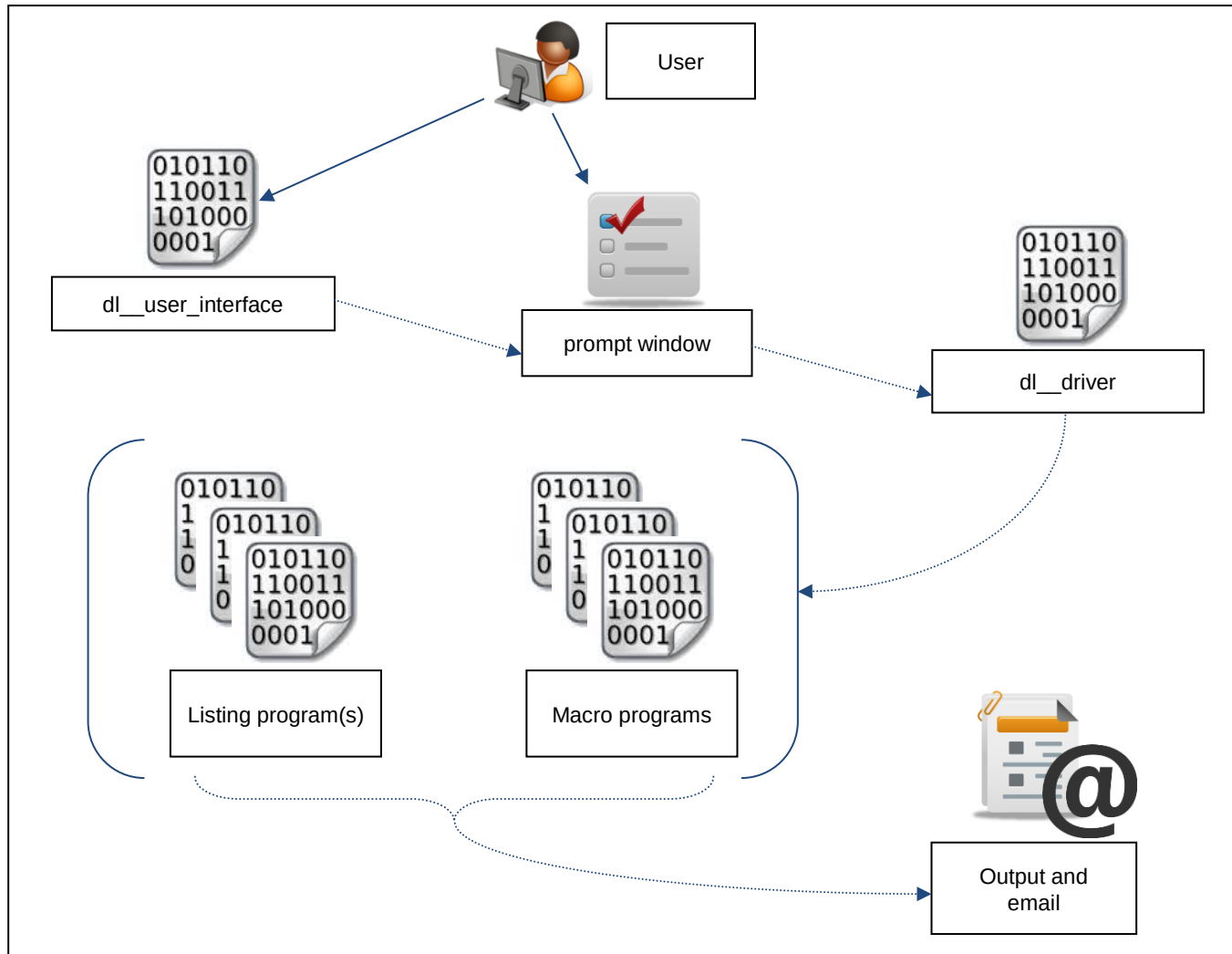
Macro Programs



- ▶ Collection of SAS user defined macros, each is stored within a dedicated program.
- ▶ Provide frequently used functionalities.
- ▶ Standardize the code writing and improve accessibility.
- ▶ Make development and resolution of issues easier and quicker.



Workflow



Index



- ▶ Introduction.
- ▶ Structure and functioning.
- ▶ Input and Output.
- ▶ Execution.
- ▶ A look at the code: what happens in the background?
- ▶ Conclusion.





Input and output

- ▶ Input
 - ▶ Standard CDISC CDASH data
 - ▶ Standard CDISC SDTM data
 - ▶ Non-standard clinical data
 - ▶ Raw data converted into SDD with appropriate tools
 - ▶ Excel spreadsheets read in by the listing using SDD functionality
 - ▶ Oracle tables

- ▶ Output
 - ▶ All listings are generated as .xml file saved with a .xls extension;
 - ▶ Stored in an archive .zip file to reduce the file size by up to 99%.

Output example



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	
1	Center	Subject	Date of Visit 1	Gender	Date of Birth	Numeric Age of Subject	Age Units	Group	Ethnicity	Race	Other Race	Height	Height Unit	Height in CM	Weight	Weight Unit	Weight in KG	Previous Vaccination	D
41	101	101019	2013-12-05	M	1961-05-28	52	YEARS	1	NOT HISPANIC OR LATINO	WHITE		69	IN	175	146	LB	66.2		Y
42	101	101020	2013-12-05						NOT HISPANIC OR LATINO	BLACK OR AFRICAN AMERICAN		73	IN	185	163	LB	74		Y
43	101	101020	2013-12-05	M	1968-09-23	45	YEARS	1	NOT HISPANIC OR LATINO	BLACK OR AFRICAN AMERICAN		73	IN	185	163	LB	74		Y
44	101	101021	2013-12-06						HISPANIC OR LATINO	WHITE		59	IN	150	155	LB	70.3		Y
45	101	101021	2013-12-06	F	1990-01-15	23	YEARS	1	HISPANIC OR LATINO	WHITE		59	IN	150	155	LB	70.3		Y
46	101	101022	2013-12-06						NOT HISPANIC OR LATINO	BLACK OR AFRICAN AMERICAN		67	IN	170	256	LB	116.2		Y
47	101	101022	2013-12-06	F	1982-01-04	31	YEARS	1	NOT HISPANIC OR LATINO	BLACK OR AFRICAN AMERICAN		67	IN	170	256	LB	116.2		Y
48	101	101023	2013-12-09						NOT HISPANIC OR LATINO	WHITE		67	IN	170	253	LB	114.8		Y
49	101	101023	2013-12-09	F	1972-06-10	41	YEARS	1	NOT HISPANIC OR LATINO	WHITE		67	IN	170	253	LB	114.8		Y
50	101	101024	2013-12-09						HISPANIC OR LATINO	WHITE		65	IN	165	173	LB	78.5		Y
51	101	101024	2013-12-09	F	1969-01-19	44	YEARS	1	HISPANIC OR LATINO	WHITE		65	IN	165	173	LB	78.5		Y
52	101	101025	2013-12-12						NOT HISPANIC OR LATINO	WHITE		71	IN	180	210	LB	95.3		Y
53	101	101025	2013-12-12	M	1966-06-05	47	YEARS	1	NOT HISPANIC OR LATINO	WHITE		71	IN	180	210	LB	95.3		Y
54	102	1021001	2013-12-05						NOT HISPANIC OR LATINO	WHITE		171	cm	171	58.8	kg	58.8		Y
55	102	1021001	2013-12-05	F	1971-05-16	42	YEARS	1	NOT HISPANIC OR LATINO	WHITE		171	cm	171	58.8	kg	58.8		Y
56	102	1021002	2013-12-10						NOT HISPANIC OR LATINO	WHITE		181	cm	181	73.3	kg	73.3		Y



Index



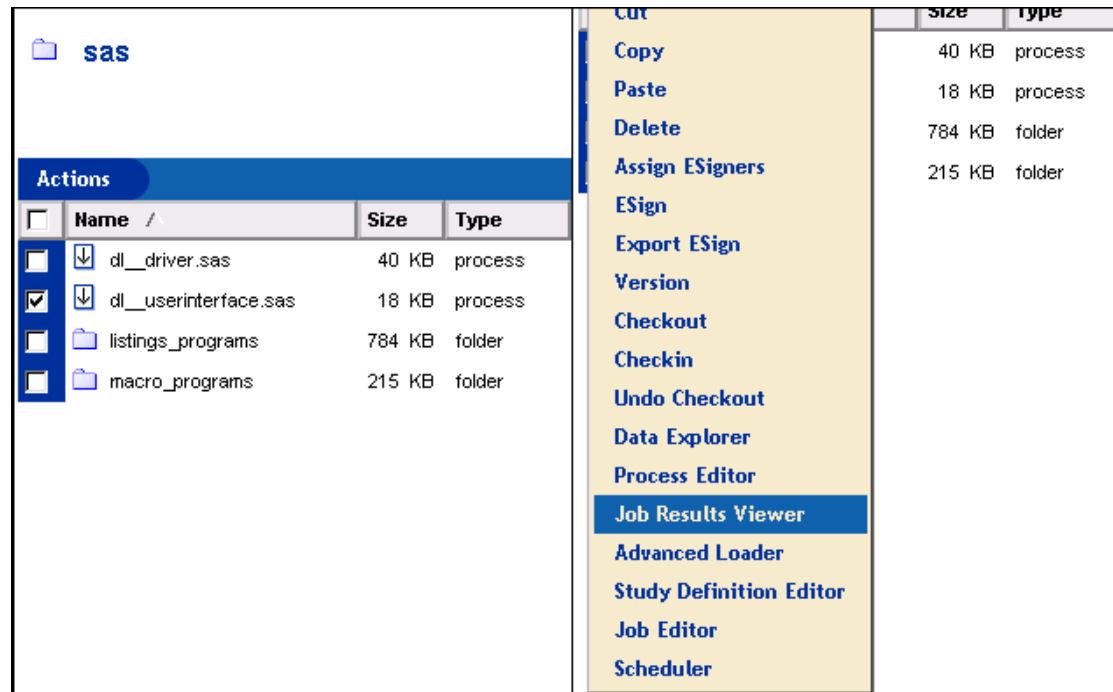
- ▶ Introduction.
- ▶ Structure and functioning.
- ▶ Input and Output.
- ▶ Execution.
- ▶ A look at the code: what happens in the background?
- ▶ Conclusion.



Execution



- ▶ User executes the main program dl__userinterface.sas in SDD using the “Jobs Result Viewer” functionality.

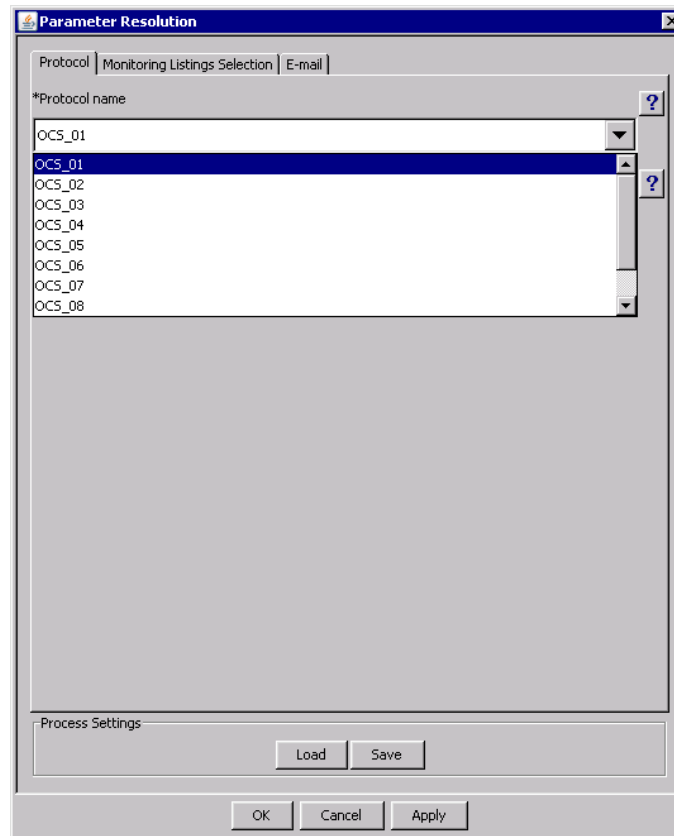


- ▶ SDD builds a prompt window presented to the user to select parameters required to complete the execution



Prompt window: Protocol

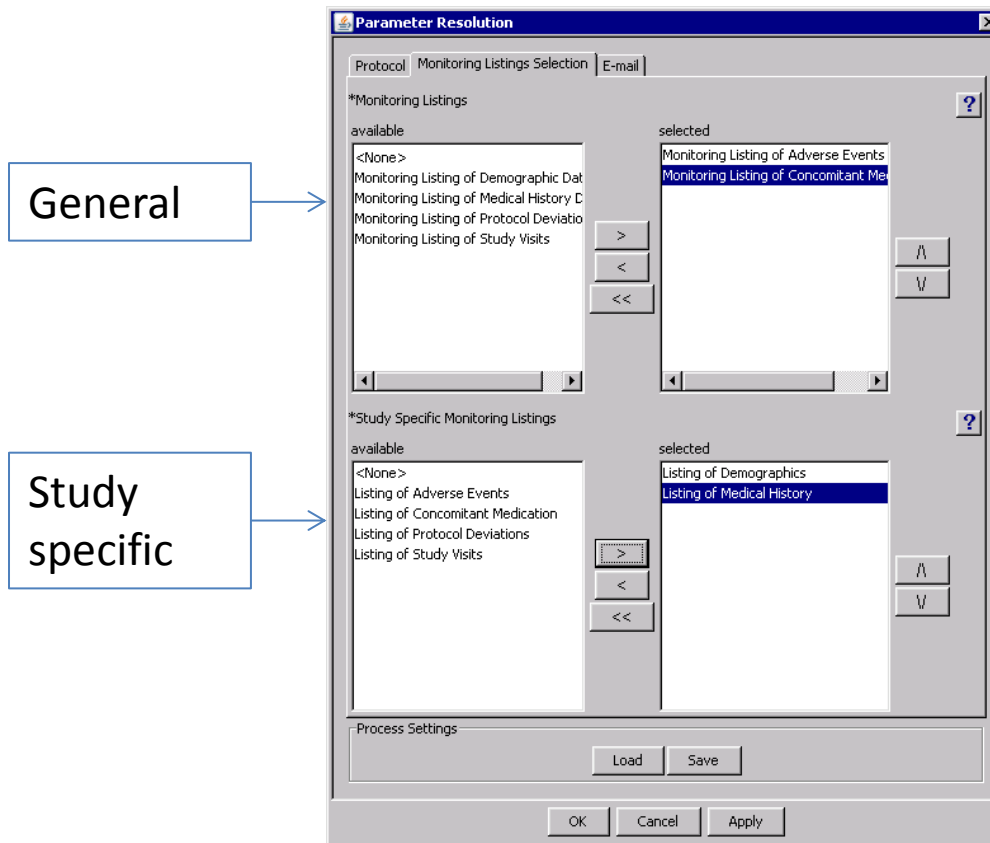
- ▶ Protocol
- ▶ To select the study protocol.





Prompt window: Listings

- ▶ Monitoring listings selection
- ▶ To select the monitoring listing(s) that he wants to generate.



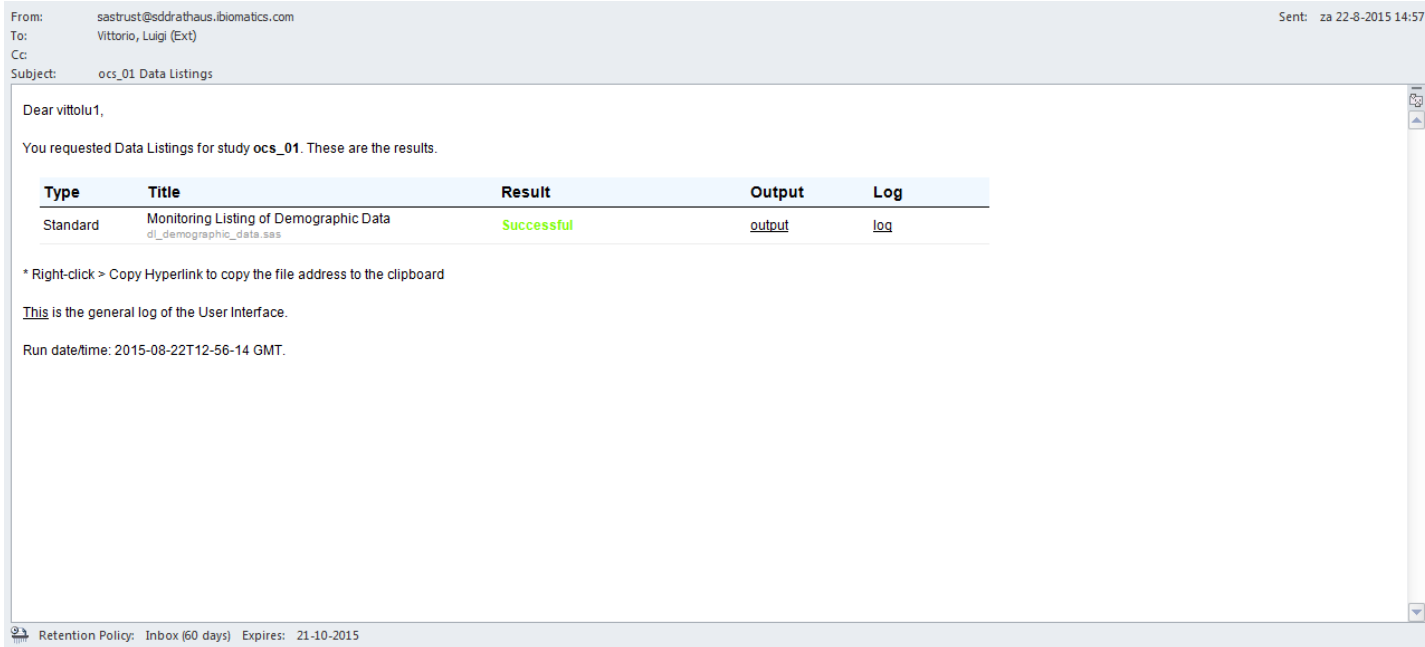


Prompt window: E-mail

- ▶ E-mail
- ▶ To handle the sending of the results via email.

The screenshot shows a Windows-style dialog box titled "Parameter Resolution". It has three tabs: "Protocol", "Monitoring Listings Selection", and "E-mail". The "E-mail" tab is active. Inside the dialog, there is a section labeled "*Send to e-mail" with a dropdown menu currently set to "Yes". To the right of this dropdown is a question mark icon. Below this is a section labeled "Other e-mail addresses" with a text input field containing "example@email.com". To the right of this text field is another question mark icon. At the bottom of the dialog, there is a "Process Settings" section with "Load" and "Save" buttons. At the very bottom are "OK", "Cancel", and "Apply" buttons.

Output e-mail



Possible scenario	Message in the email
The tool could not find the program in the appropriate location. There is no log file and output file available for the user.	Program not found
Program is executed. Errors occur during execution and prevent SAS from generating an output file. Only the log is available for the user.	No output created
Program is executed. Errors occur during execution, but do not prevent SAS from generating an output file. Both log and output file are available for the user, however output is not reliable, since errors occurred during execution.	Output with errors
Program is executed. No errors occur during execution. Both log and output file are available for the user.	Successful



Sections



- ▶ Introduction.
- ▶ Structure and functioning.
- ▶ Input and Output.
- ▶ Execution.
- ▶ A look at the code: what happens in the background?
- ▶ Conclusion.



dl__userinterface.sas



- ▶ dl__userinterface.sas makes use of several parameters set up in SDD parameters section.

* Process Editor - dl__userinterface.sas - <version not available> (/user_folders/vittolu1/demo/swlib/cdm/monit/dev/sas) from SAS Drug Development Domain

```

35
36 /* SDD Macro Parameters */
37 data work.sddparms;
38   set &sddparms
39     (where=(id = "<process>" and valtype = "path"));
40 run;
41

```

#	Var Name	Type	Label	Enabled	Required	Default	Tabname
1	*LOG*	SAS log	SAS log	☐	☐	SAS Log File	System Files
2	*LST*	SAS output	SAS output	☐	☐	SAS Output File	System Files
3	SDDPARMS	Process parameter val...	Process parameter values	☐	☐	SAS v9 Data Set (PC, Unix)	System Files
4	*PGM*	SAS program	SAS program	☐	☐	SAS Program File	System Files
5	PROT_LIST	Input data table	Protocol Selection	☐	☐		Protocol
6	PROT_NAME	Single variable selection	Protocol Name	☐	☐	<STUDYID>	Protocol
7	PROT_VAL	Column value list	Protocol name	☐	☒		Protocol
8	DL_LIST	Input data table	List of Monitoring Listings	☐	☐		Monitoring Listings Selection
9	TITLE_VAR	Single variable selection	Monitoring Listing Title	☐	☐	<TITLE>	Monitoring Listings Selection
10	DL_TITLE	Column value list	Monitoring Listings	☐	☒		Monitoring Listings Selection
11	SDL_LIST	Input data table	List of Study Specific Monitoring Listings	☐	☐		Monitoring Listings Selection
12	STITLE_VAR	Single variable selection	Monitoring Listing Title	☐	☐	<TITLE>	Monitoring Listings Selection
13	SDL_TITLE	Column value list	Study Specific Monitoring Listings	☐	☒		Monitoring Listings Selection
14	DRVLOC	Folder	Folder	☐	☐		Parameters
15	RPSTDAT	Text field	Text field	☐	☐		Parameters
16	RPENDAT	Text field	Text field	☐	☐		Parameters
17	USER_EMAIL	Selection list	Send to e-mail	☒	☒	<1>	E-mail
18	OTHER_EMAIL	Text field	Other e-mail addresses	☒	☐		E-mail
19	MACROPATH	Folder	Folder	☐	☐		Parameters
20	TESTCDP	Boolean field	Use TESTCDP environment?	☒	☐	N	Protocol
21	LISTTYPE	Text field	Text field	☒	☒		Parameters

dl_driver.sas



- ▶ Example 1: Obtain a list of all SAS macro programs and include each program file.

```
%sasdrugdev_login(  
  url          = https://&servername./sddremote,  
  sdduserid    = &_sddusr_,  
  sddpassword  = &_sddpw_  
);  
  
%sasdrugdev_getobjects(  
  sddpath       = &programpath./macro,  
  recursivelevel= 1,          /* No subfolders.          */  
  typedef       = process, /* Only program files. */  
  dsname        = work.list  
);  
  
%sasdrugdev_logout();  
  
data _null_;  
  set work.list (where = (iscontainer = 0));  
  call execute('%include macro('||strip(scan(name, 1, '.'))||');');  
run;
```

dl_driver.sas



- ▶ Example 2: Execute the data listings by including the programs. Each listing program contains a macro and a call to the macro, so including the program automatically executes it.

```
%global progame;
%let progame=&pgmname;
%global listtitle;
%let listtitle=&title;
%global filename zipfile;

%let filename=&outloc/&progame._&sddusr._&datetimestamp..xls;
%let filepath=&outpath/&progame._&sddusr._&datetimestamp..xls;
%let logfile =&outloc/&progame._&sddusr._&datetimestamp..log;
%let zipfile=&outloc/&progame._&sddusr._&datetimestamp..zip;

/* Redirect the log file. */
filename _logfile sasxbamw "&logfile" user="&sddusr_"
pass="&sddpw_";
%put Redirecting log to &logfile;
proc printto log=_logfile new;
run;

%put Redirected from &genlog;
%include &type._list(&progame..sas);
```

Data listings programs



► Example 1: Listing of Duplicated subject numbers

Page 1/3

```
%macro duplicate_subjects;
%let progame=dl_duplicate_subjects;
  %let listtitle=Listing of Duplicate Subjects;
  %let hasdata = %sysfunc(exist(prot.dm));
  %if (&hasdata = 1) %then %do;
    proc sort  data = prot.dm
              out = work.dm;
      by subjectkey studyevent studyeventrepeatkey form formrepeatkey;
      where form = "DM";
    run;
    proc sql;
      create table work.dm_all as
      select *
      from work.dm
      group by dmsubjcd, sex, brthdat
      having count(*) > 1;
    quit;
    %dl_merge_sites(
      dsin=work.dm_all,
      dsout=work.dm_all
    );
  %end;
%end;
```

Data listings programs



Page 2/3

```
data _null_;
  if 0 then set dm_all nobs=nobs;
  call symputx("numobs",nobs);
  stop;
run;
%if (&numobs = 0) %then %let hasdata = 0;
%end;
%dl_setup;
%dl_standard_titles;
%dl_cover_page;
%if (&hasdata = 1) %then %do;
  %dl_merge_form_status(dsin=work.dm_all,
                        dsout=work.dm_all);
  %dl_determine_colstr(dsin=work.dm_all,
                      vars=pfmnemonic subjid dmsubjcd sex brthdat
                          svcomplete svc_date frozen locked
                          first_entry_datetime last_entry_datetime
                          upload_datetime);
  %dl_start_page(sheetname=&listtitle,
                 frozen_rowheaders=2,
                 frozen_headers   =1);
```



Data listings programs

Page 3/3

```
proc sort data=work.dm_all;
  by dmsubjcd subjid;
run;
proc report data=work.dm_all headline headskip pspace=2 split='@'
           nowd missing
           style(column)={font_size=7pt}
           style(header)={font_size=7pt font_weight=bold};
  column &colstr;
  define pfmnemonic / display %dl_col(35, string) 'Center';
  define subjid      / display %dl_col(35, string) 'Subject';
  define dmsubjcd    / display %dl_col(35)         'Subject Code';
  define sex         / display %dl_col(35)         'Gender';
  define brthdat     / display %dl_col(80)         'Birth date';
  &define_svlocfr;
  &define_timing;
run;
%end;
%if (&hasdata = 0) %then %do;
  %dl_no_data_page(listtitle=&listtitle);
%end;
%dl_end;
%mend duplicate_subjects;
```



Macro programs

Example 1: Macro %dl_has_var

```
%macro dl_has_var(ds, var);  
  %put ### Macro DL_HAS_VAR - start;  
  %put ### ;  
  %put ### Parameters::  
  %put ###    - 1      : &ds;  
  %put ###    - 2      : &var;  
  %let varnum=0;  
  %let dsid=%sysfunc(open(&ds));  
  %if &dsid > 0 %then %do;  
    %let varnum=%sysfunc(varnum(&dsid, &var));  
    %let dsid=%sysfunc(close(&dsid));  
  %end;  
  %put &varnum;  
  %put ### Macro DL_HAS_VAR - end;  
%mend dl_has_var;
```

Index



- ▶ Introduction.
- ▶ Structure and functioning.
- ▶ Input and Output.
- ▶ Execution.
- ▶ A look at the code: what happens in the background?
- ▶ Conclusion.





Pros and Cons

- ▶ Powerful instrument to generate monitoring listings for clinical studies.
- ▶ Pros:
 - ▶ Flexibility
 - ▶ Interactive
 - ▶ Multiple listings on demand
 - ▶ Code development easy (standard macros)
 - ▶ Output via email
- ▶ Cons:
 - ▶ Java engine
 - ▶ Execution time



Any questions?



Thank you

