Non-Executable SAS®-Code for the FDA

Background, Examples and Experiences from an Oncology Filing

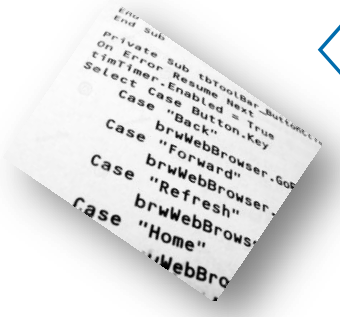
Christoph Ziegler
F. Hoffmann-La Roche Ltd | MDBP
Malzgasse 30 | Bldg. 670/317
CH-4070 Basel | Switzerland
+41 61 68 89200
christoph.ziegler@roche.com

PhUSE - Single Day Event – Frankfurt 15APR2015

Agenda

- Background
  - What is (Non-)Executable Code?
  - Why is (Non-)Executable Code needed?
  - Considerations for providing Non-Executable Code and Strategy
- Examples of Non-Executable Code
  - Annotated Output
  - Annotated Output with Code-Snippets
  - Metadata-File Approach
  - **Non-Executable Program**
- Advantages of Non-Executable Code
- Summary / Experiences
- Outlook / Q&A

Background I
What is (Non-)Executable Code?



- Executable Code (aka Stand-Alone Programs):
  - Program is prepared in a way that it can be run on eSUB deliverables (datasets / *.xpt) at the agency by just updating the path to the input datasets
  - All underlying macros included
  - Usually lots of lines of code (if company specific reporting system is used)
- Non-Executable Code (aka Readable Programs):
  - Program cannot be run / executed
  - Main purpose is that the agency can see derivations, procedures, calculations, options applied
  - Streamlined approach with less code compared to executable files
- Report Object Description (aka Output Description):
  - Annotated output strategy where parts of the output are described in words and/or code-snippets
  - Streamlined approach that allows the agency to better understand how numbers are derived /calculated

Background II
Why is (Non-)Executable Code needed?

- FDA might ask for code (executable or non-executable) at any time:
  - Whether offered upfront (e.g. in Pre-Meeting Package - PMP) or not
  - Whether sent upfront (as part of the eSUB package/ BLA) or not
  - During actual BLA review process (usually with short turnaround times)
  - → Regardless of sponsor strategy (offering / sending code upfront)
  - → Delivering code is not a standard deliverable

- Request for code due to various reasons:
  - Understand derivations applied
  - Understand analyses done
  - Understand numbers in report objects (outputs)
  - Clarification of programming / coding done at the agency
  - Agency might want to see SAS® procedure used and the options applied

→ No standard available / every company approaches it their own way

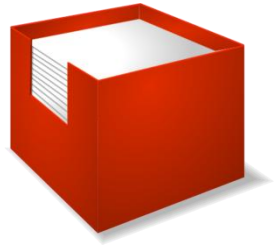
Background III

Considerations for providing Non-Executable Code and Strategy

- Every company has a different strategy
  - Offer executable code for ALL programs by default
  - Offer executable code for limited number of programs by default
  - Offer non-executable programs for limited number of (key-safety/efficacy) programs by default
  - Offer nothing by default
- Clarification on scope needed
  - Executable vs non-executable code vs report object description
  - Key-programs vs non-key (all) programs vs no programs

Background IV

Considerations for providing Non-Executable Code and Strategy



- Typical approach from the sponsor might look as follows:
  - Sponsor asks whether the agency agrees with proposed content of e.g. a (s-)BLA, i.e. table of contents, patient narratives, CSRs, patient data, dataset content and format **and statistical programs content**
- Typical answer from the agency might look as follows:
  - "Yes, but please include the SAS® programs **(do not have to be stand-alone)** used for generating tables / numbers xyz / numbers in the label etc etc"

  → Very vague, i.e. can be stand-alone or not

  → Various ways to provide non-executable code (or output descriptions)

  → Consistent approach needed in order to fulfill the agency's request

Background V

Considerations for providing Non-Executable Code and Strategy

- It is important to make use of asking questions and sending an example
  - *Does the agency need *.sas® files or is it OK to provide SAS® code in PDF format?*
  - *is it OK for the agency to exclude the pieces of code producing the layout?*
  - *if a program is only there to analyze / summarize data that is already derived in analysis datasets, does the agency need the code that produces the variables / endpoints that are analyzed?*
  - *in general, does the agency prefer non-executable code or executable code (hard to read / long / …)?*
- Request for non-executable file understandable (→ executable code = very hard to understand (lots of lines of code))
- The agency agreed to the proposed way of submitting non-executable code based on the example sent by the sponsor
- Different agency reviewers might have different preferences

Examples of Non-Executable Code I Annotated Output

Table 1 Proportion of Patients with XY Response at Time-point X (ITT Population)

	TRT1	TRT2	TRT2+TRT3	TRT2+TRT1
	N=1	N=1	N=1	N=1
XY Response at Time-point X				
Evaluable	2	2	2	2
Responders	3(4)	3(4)	3(4)	3(4)
Difference in Treatment effect	NA	5	5	5
95% C.I.	NA	[6,7]	[6,7]	[6,7]
P-value	NA	8	8	8

Last Observation Carried Forward rule was used for missing data and data were carried forward at each time point

XY Response is a 50% reduction of x y z, as well as in 2 out of 4 additional parameters of :
physician's global assessment of x y z, patient's global assessment of x y z, patient's assessment of pain, Health Assessment Questionnaire [HAQ]

At each time point where an XY response was calculated, x y z was the one with the largest percentage improvement over its baseline value.

The 95% confidence interval was computed for the difference in proportions comparing the TRT1 group with each TRT2 group
[TRT2, TRT2+TRT3 and TRt2+TRT1]

P-value from Fisher's Exact Test, comparing the TRt1 group with each TRT2 group [TRT2, TRT2+TRT3 and TRT2+TRT1]

→ Basis = Output

→ Annotations (explanation on next slide) + Important Footnotes added

→ Not really non-executable code, more explanation how the numbers are derived

Examples of Non-Executable Code I Annotated Output

1	Total number of patients randomized to each treatment group (**programming**: all patients from LIB.DEMOGRAPHY dataset where popn=1)
2	Number of patients included in the summary (**programming**: all patients from LIB.DEMOGRAPHY dataset where popn=1 and who completed time-point X where timepointx=1)
3	The number of patients with an XY response at time-point x . (**programming**: where response_xy=1 and timepointx=T1 in the analysis dataset, for Response XY derivation see RAWDATA Spec in Appendix **) To achieve response xy requires at least a 50% improvement, compared to baseline, in both x and y counts, as well as in 2 out of 5 additional parameters of: physician's global assessment of disease, patient's global assessment of activity, patient's assessment of pain, Health Assessment Questionnaire (HAQ). At each time point where an xy response is calculated the xyz used will be that with largest percentage improvement over its baseline value.
4	(**3**/**2**) * 100
5	The difference in treatment effect is the difference in X Y response rates in TR1 group compared with each of the other TRT2 treatment arms (TRT2, TRT2+TRT3 and TRT2+TRT1). This is calculated by p_2-p_1 where p_1 is the proportion of patients with an X Y response at Time-point X for the TRT1 value p_2 is the proportion of patients with an X Y response at Time-point X for one of the TRT2 treatment arms. The difference in treatment effect will be calculated to 2 decimal places.
6,7	95% confidence interval for the difference in proportions[7] between TRT1 arm and each of the other TRT2 treatment is calculated as $p_2 - p_1 \pm 1.96 \times se(p_2 - p_1)$ Where $se(p_2 - p_1) = \sqrt{\frac{p_1(-p_1)}{n_1} + \frac{p_2(-p_2)}{n_2}}$ n_1=total number of evaluable patients in TRT1 group n_2=total number of evaluable patients in each of the TRT2 arms 95% Confidence Intervals will be calculated to 2 decimal places.
8	The p-value will be derived from the Fisher's Exact test, comparing the difference in X Y response rates between the TRT1 group with each of TRT2 group (TRT2, TRT2+TRT3 and TRT2+TRT1.). P-values will be calculated to 3 decimal places.

Examples of Non-Executable Code II
Annotated Output with Code-Snippets

et_time_pfs Summary of PFS (ITT)

Protocol(s): BOxxxxx (xxxxxx)
Analysis Population: ITT - Stage X Population
Snapshot Date: XXJUN2013   Cutoff Date: XXMAY2013

	A TRT1 (N=530)	B	TRT2 (N=523)	
Patients with event	209 (30.3 %)		304 (31.2 %)	C
Patients without event*	131 (39.7 %)		229 (48.8 %)	
Time to event (months)				
Median#	25.2		26.7	
95% CI for Median#	[19.2;17.0]		[11.2;23.0]	
25% and 75%-ile#	11.2;23.3		13.2;.	
Range##	0.0 to 34.6		0.0 to 36.2	
P-Value (Log-rank Test, stratified**)		<.0018 D		
Hazard Ratio (stratified**)		0.59		
95% CI		[0.51;0.56] E		
X year duration				
Patients remaining at risk	169		220	C
Event Free Rate	0.64		0.66	
95% CI for Rate	[0.58;0.70]		[0.59;0.90]	

* censored
\# Kaplan-Meier estimates
including censored observations
** Stratified by XXX stage at baseline

Program : /xxxxxx/boxxxxx/et_time_pfs.sas
Output : /xxxxxx/xxxxxxx/reports/et_time_pfs_O.out

Note: Numbers are fictive in this example

Examples of Non-Executable Code II
Annotated Output with Code-Snippets

- Actual program is provided
- References to red boxes and to important program code lines are made (excerpt)

Lines 52-78:
Macro parameters are set to initialize program.

NOTE: Treatment groups are set in %setupn.sas to be 'as randomized' [DEMO.RND]

A

Lines 94-99:
EVENT analysis dataset is processed so that Overall Survival Time (Days) [EVENT.EVOSTM] is converted from days to months:

```
data __event;
 set lib.event;
  __evostm = evostm/30.4;
run;
```

Lines 321-392

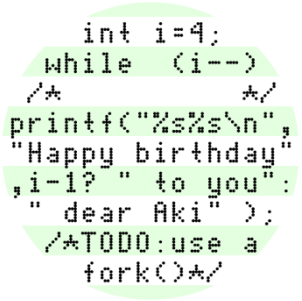
This section of code performs the actual analysis and is looped through twice, once for each treatment group. A breakdown of the analysis performed is below:

%lifetest

Macro that performs a PROC LIFETEST and extracts the crude rate information for the events, point estimators and CIs for a list of time points. Used to derive the first 2 and last blocks in the report - patients with/without event, Time to Event analysis (except the p-value) & 1 year duration. 2 key steps are performed in this macro:

C

```
data _analyse ;
    set _analysis1 end=eof ;
    where __evostm ne . and evdcen ne . ;
    _censor = 0 ;
    if evdcen = 0 then _censor = 1 ;
run ;

ODS OUTPUT ProductLimitEstimates=_estimates;
ODS OUTPUT Quartiles=_quart;
ODS OUTPUT CensoredSummary=_events;
proc lifetest data=_analyse   method=km   outs=_outsurv   alpha=0.05   alphaqt=0.05 ;
       time  __evostm * _censor ( 1 ) ;
       strata  _trtorder;
run ;
```



……**Etc Etc**……

Examples of Non-Executable Code III Metadata-File Approach

- As described in the CDISC Analysis Data Model v2.1
- To explain key analysis shown in report objects– a potential replacement for executable code / can be regarded as kind of non-executable code

→ However, it is not real non-executable code

Protocol: CDISCPILOT01 Page 1 of 1
Population: Efficacy

Table 14-3.01
Primary Endpoint Analysis: ADAS Cog (11) - Change from Baseline to Week 24 - LOCF

	Placebo (N=79)	Xanomeline Low Dose (N=81)	Xanomeline High Dose (N=74)
Baseline			
n	79	81	74
Mean (SD)	24.1 (12.19)	24.4 (12.92)	21.3 (11.74)
Median (Range)	21.0 (5;61)	21.0 (5;57)	18.0 (3;57)
Week 24			
n	79	81	74
Mean (SD)	26.7 (13.79)	26.4 (13.18)	22.8 (12.48)
Median (Range)	24.0 (5;62)	25.0 (6;62)	20.0 (3;62)
Change from Baseline			
n	79	81	74
Mean (SD)	2.5 (5.80)	2.0 (5.55)	1.5 (4.26)
Median (Range)	2.0 (-11;16)	2.0 (-11;17)	1.0 (-7;13
p-value(Dose Response) [1] [2]			0.245 (1)
p-value(Xan - Placebo) [1] [3]		0.569	0.233
Diff of LS Means (SE)		-0.5 (0.82)	-1.0 (0.84)
95% CI		(-2.1;1.1)	(-2.7;0.7)
p-value(Xan High - Xan Low) [1] [3]			0.520 (2)
Diff of LS Means (SE)			-0.5 (0.84)
95% CI			(-2.2;1.1)

[1] Based on Analysis of covariance (ANCOVA) model with treatment and site group as factors and baseline value as a covariate.
[2] Test for a non-zero coefficient for treatment (dose) as a continuous variable.
[3] Pairwise comparison with treatment as a categorical variable: p-values without adjustment for multiple comparisons.
Source: C:\cdisc_pilot\PROGRAMS\DRAFT\TFLs\rtf_eff1.sas 21:05 Monday, June 26, 2006

Examples of Non-Executable Code III
Metadata-File Approach

Metadata Field	*Definition of field*	Metadata
DISPLAY IDENTIFIER	*Unique identifier for the specific analysis display*	Table 14-3.01
DISPLAY NAME	*Title of display*	Primary Endpoint Analysis: ADAS Cog (11) - Change from Baseline to Week 24 - LOCF
RESULT IDENTIFIER	*Identifies the specific analysis result within a display*	Pairwise treatment comparisons
PARAM	*Analysis parameter*	ADAS-Cog (11) Total Score
PARAMCD	*Analysis parameter code*	ACTOT11
ANALYSIS VARIABLE	*Analysis variable being analyzed*	CHG
REASON	*Rationale for performing this analysis*	Primary efficacy analysis as pre-specified in protocol
DATASET	*Dataset(s) used in the analysis.*	ADQSADAS
SELECTION CRITERIA	*Specific and sufficient selection criteria for analysis subset and / or numerator*	ITTFL='Y' and AVISIT='Week 24' and PARAMCD='ACTOT11'
DOCUMENTATION	*Textual description of the analysis performed*	Linear model analysis of ADAS-Cog(11) total score change from baseline at Week 24 for pairwise treatment comparisons and adjusted means; missing values imputed using LOCF, Efficacy population. Used randomized treatment as class variable; site group as class variable; and baseline ADAS-Cog score in model.
PROGRAMMING STATEMENTS	*The analysis syntax used to perform the analysis*	PROC GLM; CLASS SITEGR1 TRTP; MODEL CHG = TRTP SITEGR1 BASE; ESTIMATE 'H VS L' TRTP 0 1 -1; ESTIMATE 'H VS P' TRTP -1 1 0; ESTIMATE 'L VS P' TRTP -1 0 1; LSMEANS TRTP / OM STDERR PDIFF CL; RUN;

Analysis Results Metadata Field	Description
DISPLAY IDENTIFIER	A unique identifier for the specific analysis display (such as a table or figure number)
DISPLAY NAME	Title of display, including additional information if needed to describe and identify the display (e.g., analysis population)
RESULT IDENTIFIER	Identifies the specific analysis result within a display. For example, if there are multiple p-values on a display and the analysis results metadata specifically refers to one of them, this field identifies the p-value of interest. When combined with the display identifier provides a unique identification of a specific analysis result.
PARAM	The analysis parameter in the BDS analysis dataset that is the focus of the analysis result. Does not apply if the result is not based on a BDS analysis dataset.
PARAMCD	Corresponds to PARAM in the BDS analysis dataset. Does not apply if the result is not based on a BDS analysis dataset.
ANALYSIS VARIABLE	The analysis variable being analyzed
REASON	The rationale for performing this analysis. It indicates when the analysis was planned (e.g., "Pre-specified in Protocol," "Pre-specified in SAP," "Data Driven," "Requested by Regulatory Agency") and the purpose of the analysis within the body of evidence (e.g., "Primary Efficacy," "Key Secondary Efficacy," "Safety"). The terminology used is sponsor defined. An example of a reason is "Primary Efficacy Analysis as Pre-specified in Protocol."
DATASET	The name of the dataset used to generate the analysis result. In most cases, this is a single dataset. However, if multiple datasets are used, they are all listed here.
SELECTION CRITERIA	Specific and sufficient selection criteria for analysis subset and / or numerator – a complete list of the variables and their values used to identify the records selected for the analysis. Though the syntax is not ADaM-specified, the expectation is that the information could easily be included in a WHERE clause or something equivalent to ensure selecting the exact set of records appropriate for an analysis. This information is required if the analysis does not include every record in the analysis dataset.
DOCUMENTATION	Textual description of the analysis performed. This information could be a text description, pseudo code, or a link to another document such as the protocol or statistical analysis plan, or a link to an analysis generation program (i.e., a statistical software program used to generate the analysis result). The contents of the documentation metadata element contains depends on the level of detail required to describe the analysis itself, whether or not the sponsor is providing a corresponding analysis generation program, and sponsor-specific requirements and standards. This documentation metadata element will remain free form, meaning it will not become subject to a rigid structure or controlled terminology.
PROGRAMMING STATEMENTS	The software programming code used to perform the specific analysis. This includes, for example, the model statement (using the specific variable names) and all technical specifications needed for reproducing the analysis (e.g., covariance structure). The name and version of the applicable software package should be specified either as part of this metadata element or in another document, such as a Reviewer's Guide (see Appendix B for more information about a Reviewer's Guide).

Examples of Non-Executable Code IV
Non-Executable Program

- Example sent to the FDA in advance in order to confirm that this is what the agency expects (example on next slides)
- As multiple non-executable programs were provided, rules for the files were set up (see below)
- TOC for these programs to be part of the eSUB TOC

  - each program should be one word document (will be converted to PDF later) – if subgroup analyses are done / same output for different efficacy endpoint etc. then its not repeated in separate files (selection rules described)
  - all code that isn't relevant for the FDA should be removed
  - all comments should be revised / looked at and additional comments should be inserted if code is unclear
  - all SAS® keyword should be in upper case
  - all VAD variables should have the dataset name attached. eg, if the code says "where pop_saf=1" it should read "where demo.pop_saf=1" → increases readability
  - all program comments should be done this way:

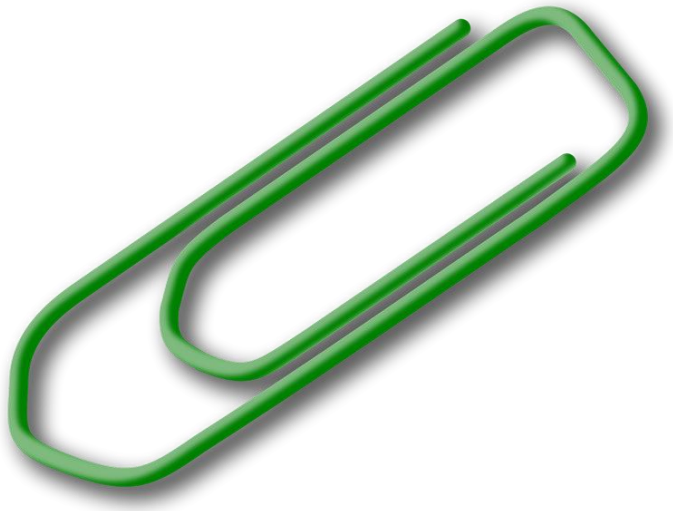
```
/*
||  xxxxx
*/
```

  - the program header should always be the same and consistent across files
  - font: courier new 7 (still readable but not too small)
  - the filename should be equal to the program name
  - efficacy programs should include the data selection from the analysis dataset(s) and the SAS® stats procedure with the options that were used

Examples of Non-Executable Code IV
Non-Executable Program

```
/*****************************************************************************************
|| Output Title                End of Treatment Response (ITT)
|| Output Name                 et_resp_rsetr_O
|| Input Analysis Data Sets    RESPONSE, DEMOEXT
|| Analysis Population         ITT  - patients with RESPONSE.POP_ITT = 1
|| Displayed Treatment Arm 1   TRTX - patients with RESPONSE.RNDGRD_I = 'TRTX'
|| Displayed Treatment Arm 2   TRTY - patients with RESPONSE.RNDGRD_I = 'TRTY'
||
|*****************************************************************************************
||  Sub-setting of patient data from analysis dataset RESPONSE
||      - select patients included in ITT population (RESPONSE.POP_ITT)
||      - use randomized treatment for analysis (RESPONSE.RNDGRD_I)
||      - assign treatment order (WORK.RXORD)
||      - select end of treatment response (RESPONSE.RSETR)
||
||  Note: RESPONSE data set has more than one observation per patient,
||        but end of treatment response is unique per patient
*/
PROC SORT DATA = response OUT = _response(KEEP = proto crtn pt rsetr rndgrd_i pop_itt) nodupkey;
   BY proto crtn pt;
RUN;

DATA _response;
   ATTRIB rxord  LENGTH = 3 FORMAT = 3. LABEL = 'Randomized Treatment Order';
   SET _response;
   IF pop_itt EQ 1 and UPCASE(rndgrd_i) IN ('TRTX' 'TRTY');
   IF UPCASE(rndgrd_i) EQ 'RCLB' THEN rxord = 1;
   ELSE                               rxord = 2;
RUN;

/*
|| End of treatment response reached at time of analysis
*/
PROC SORT DATA = demoext (KEEP = proto crtn pt etricl) OUT = _demoext;
   BY proto crtn pt;
RUN;

DATA _response;
   MERGE _response (IN = inr)
         _demoext;
   BY proto crtn pt;
   IF inr;
RUN;
```

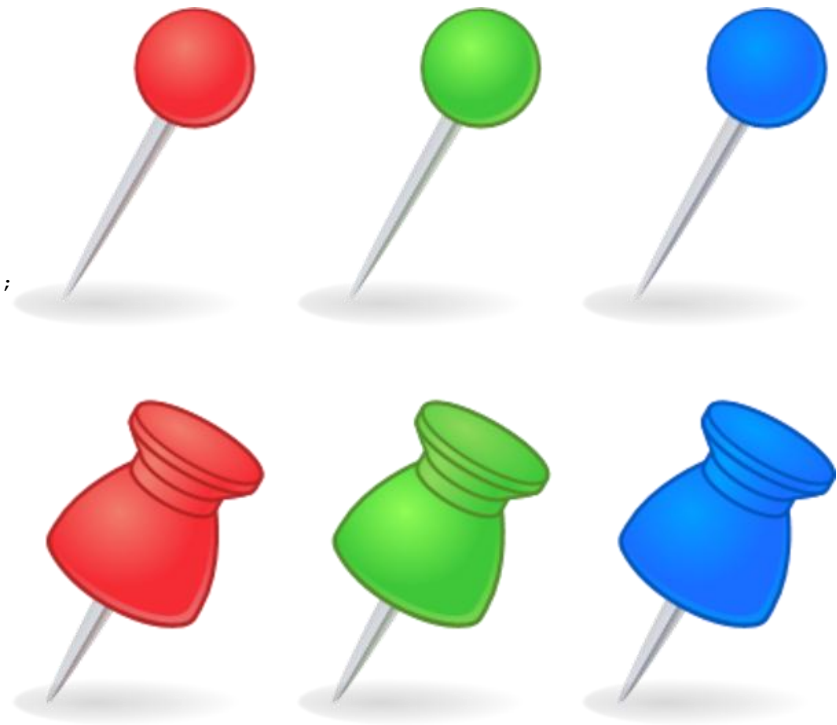
Examples of Non-Executable Code IV
Non-Executable Program



```
DATA _response;
   SET _response;
   BY proto crtn pt;

   /*
   ||  Map all PD values to PD
   */
   LENGTH etr $10;
   SELECT (rsetr);
      WHEN ("PD", "PD+", "PD**", "PDSF") etr = "PD";
      WHEN ("UNK", " ")                  etr = "MISS";
      OTHERWISE                          etr = rsetr;
   END;
   /*
   ||  Patients who did not reach end of treatment at time of analysis in separate category
   */
   IF etr EQ "MISS" AND etricl EQ "NO" THEN etr = "NOETR";
   /*
   ||  define responders and non-responders
   */
   IF UPCASE(etr) IN ('CR' 'CRI' 'PR' 'NPR') THEN response = 1;
   IF UPCASE(etr) IN ('SD' 'PD' 'MISS')      THEN response = 0;
   IF UPCASE(etr) IN ('NOETR')               THEN response = .;
RUN;

/*********************************************************************************************
||  Derive statistics
||*********************************************************************************************
||  Count totals per treatment group
*/
PROC FREQ DATA = _response;
   TABLES rxord / MISSING NOPRINT OUT = _freqtot;
RUN;
```

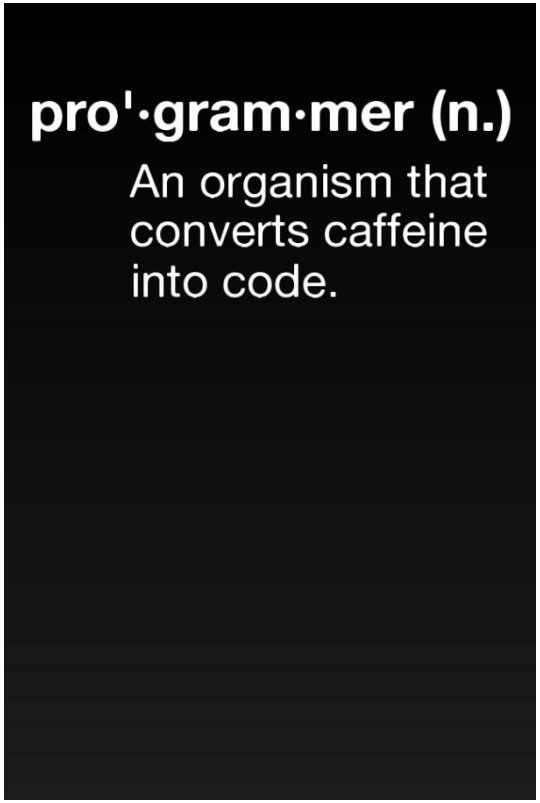
Examples of Non-Executable Code IV
Non-Executable Program

```
/*
||  Treatment comparison: Calculate Hauck-Anderson CI and perform Chi-Square test
||                        Reference group: RClb (rxord = 1)
*/
PROC SORT DATA = _ana1;
   BY mergevar;
RUN;

DATA _ana3;
   MERGE _ana1 (WHERE  = (rxord EQ 1)
                RENAME = (_acount = totref _ucount1 = respref)
                KEEP   = mergevar rxord _acount _ucount1)
         _ana1 (WHERE  = (rxord EQ 2)
                RENAME = (_acount = tot _ucount1 = resp)
                KEEP   = mergevar rxord _acount _ucount1);
   BY mergevar;
   /*
   ||  Response rates
   */
   _rate1 = resp/tot ;
   _rate2 = respref/totref ;
   /*
   ||  Differences, standard error, p-value, confidence limits
   */
   _unom  = (_rate1-_rate2) ;
   _se    = SQRT(((_rate1*(1-_rate1))/(tot-1))+(_rate2*(1-_rate2)/(totref-1))) ;
   _p     = PROBIT(1-((100*0.05)/200)) ;

   _lower   = 100 * ( _unom - _p * _se - (1/(2* MIN(totref,tot)))) ;
   _upper   = 100 * ( _unom + _p * _se + (1/(2* MIN(totref,tot)))) ;
RUN;
```



Examples of Non-Executable Code IV
Non-Executable Program

```
/*************************************************************************************************************************************
|| Output Title:              Number of Treatment Cycles Received (SAP)
|| Output Name:               st_ncyc2_V
|| Input Analysis Dataset:    DEMOEXT
|| Analysis Population:       SAP  - patients with DEMOEXT.POP_SAF = 1
|| Displayed Treatment Arm 1: RClb - patients with DEMOEXT.TRT1D   = 'TRTX'
|| Displayed Treatment Arm 2: GClb - patients with DEMOEXT.TRT1D   = 'TRTY'
||
|| *************************************************************************************************************************************
|| DEMOEXT.TOTCYR  -- source of total numbers of cycles received by patient
|| DEMOEXT.CUMDOSC -- source of cumulative doses of TRT1 by patient
|| DEMOEXT.CUMDOSR -- source of cumulative doses of TRT2 by patient
|| DEMOEXT.CUMDOSG -- source of cumulative doses of TRT3 by patient
||
|| Create dataset WORK._TO_ANALYZE - an input dataset for the output creation.
|| Create the variable WORK._TO_ANALYZE._CRTNPT - patient number combined with the clinical research center number.
*/
DATA _to_analyze;
  SET demoext;
  ATTRIB _crtnpt LENGTH=$10.;
  _crtnpt = PUT(crtn,6.)||'/'||PUT(pt,4.);
RUN;

/*
|| Calculate statistics for each treatment arm and display per row on the output.
|| Percentages are calculated based on the number of patients in the corresponding treatment arm.
|| The body of the summary report has 3 columns:
|| -- the first column contains labels/description of the calculated statistics,
|| -- the second column contains statistics calculated for RClb treatment arm,
|| -- the third column contains statistics calculated for GClb treatment arm.
||
|| For each row of the body of the summary report a macro call of %CREATE_LINE is done with the following parameters:
|| data_in  = WORK._TO_ANALYZE                  -> input dataset used for calculation of statistics;
||                                                 default value is data_in = WORK._TO_ANALYZE
|| lbl      = <value of first column of report> -> label/description of the statistics;
|| var      = <DISTINCT> <variable>             -> variable, which is used in the calculation of statistics;
||                                                 default value is var = DISTINCT _crtnpt;
|| stat     = <statistics option>               -> type of the calculated statistics; F is an absolute frequency;
|| where    = <part of SAS® "where" statement>   -> condition for selecting the data on which statistics are based on;
||                                                 default value is where = " ";
|| percent  = Y                                 -> flag indicating whether relative frequency is calculated based on the
||                                                 number of events in the corresponding treatment arm; calculated only
||                                                 if macro variable "stat" is set to F (absolute frequency);
||                                                 default value is percent = N (not calculated);
|| where_ln = <part of SAS® "where" statement>   -> condition for selecting the data on which the total number of events is
||                                                 based for calculation of relative frequency;
||                                                 default value is where_ln = " ";
|| var_ln   = <DISTINCT> <variable>             -> variable which defines the total number of events for calculation of
||                                                 relative frequency; should be set to the same value as parameter VAR;
||                                                 default value is var_ln = DISTINCT _crtnpt;
|| data_out = <resulting dataset>               -> dataset with one observation which contains the statistics
||                                                 and relative frequency (if calculated) for each treatment arm.
|| During each macro call of %CREATE_LINE a temporary dataset WORK.LINEi is created,
|| where i corresponds to the row number i of the body of the summary report.
*/
```



Examples of Non-Executable Code IV
Non-Executable Program

```
%CREATE_LINE( lbl="Total number of cycles received"                                                    ,data_out=line1  );
%CREATE_LINE( lbl="  1"       ,var=_crtnpt  ,stat=F        ,where="totcyr=1"  ,fmtnum=6.  ,percent=Y  ,data_out=line2  );
%CREATE_LINE( lbl="  2"       ,var=_crtnpt  ,stat=F        ,where="totcyr=2"  ,fmtnum=6.  ,percent=Y  ,data_out=line3  );
%CREATE_LINE( lbl="  3"       ,var=_crtnpt  ,stat=F        ,where="totcyr=3"  ,fmtnum=6.  ,percent=Y  ,data_out=line4  );
%CREATE_LINE( lbl="  4"       ,var=_crtnpt  ,stat=F        ,where="totcyr=4"  ,fmtnum=6.  ,percent=Y  ,data_out=line5  );
%CREATE_LINE( lbl="  5"       ,var=_crtnpt  ,stat=F        ,where="totcyr=5"  ,fmtnum=6.  ,percent=Y  ,data_out=line6  );
%CREATE_LINE( lbl="  6"       ,var=_crtnpt  ,stat=F        ,where="totcyr=6"  ,fmtnum=6.  ,percent=Y  ,data_out=line7  );
%CREATE_LINE( lbl=" "                                                                                 ,data_out=line8  );
%CREATE_LINE( lbl="  Mean"    ,var=totcyr   ,stat=MEAN                        ,fmtnum=8.1            ,data_out=line9  );
%CREATE_LINE( lbl="  SEM"     ,var=totcyr   ,stat=STDMEAN                     ,fmtnum=9.2            ,data_out=line10 );
%CREATE_LINE( lbl="  Std"     ,var=totcyr   ,stat=STD                         ,fmtnum=9.2            ,data_out=line11 );
%CREATE_LINE( lbl="  Median"  ,var=totcyr   ,stat=MEDIAN                      ,fmtnum=8.1            ,data_out=line12 );
%CREATE_LINE( lbl="  Min"     ,var=totcyr   ,stat=MIN                         ,fmtnum=8.             ,data_out=line13 );
%CREATE_LINE( lbl="  Max"     ,var=totcyr   ,stat=MAX                         ,fmtnum=8.             ,data_out=line14 );
%CREATE_LINE( lbl="  n"       ,var=totcyr   ,stat=N                           ,fmtnum=6.             ,data_out=line15 );
%CREATE_LINE( lbl=" "                                                                                 ,data_out=line16 );
%CREATE_LINE( lbl="Total cumulative dose Chlorambucil (mg)"                                           ,data_out=line17 );
%CREATE_LINE( lbl="  Mean"    ,var=cumdosc  ,stat=MEAN                        ,fmtnum=8.1            ,data_out=line18 );
%CREATE_LINE( lbl="  Std"     ,var=cumdosc  ,stat=STD                         ,fmtnum=9.2            ,data_out=line19 );
%CREATE_LINE( lbl="  SEM"     ,var=cumdosc  ,stat=STDMEAN                     ,fmtnum=9.2            ,data_out=line20 );
%CREATE_LINE( lbl="  Median"  ,var=cumdosc  ,stat=MEDIAN                      ,fmtnum=8.1            ,data_out=line21 );
%CREATE_LINE( lbl="  Min"     ,var=cumdosc  ,stat=MIN                         ,fmtnum=8.2            ,data_out=line22 );
%CREATE_LINE( lbl="  Max"     ,var=cumdosc  ,stat=MAX                         ,fmtnum=8.2            ,data_out=line23 );
%CREATE_LINE( lbl="  n"       ,var=cumdosc  ,stat=N                           ,fmtnum=6.             ,data_out=line24 );
%CREATE_LINE( lbl=" "                                                                                 ,data_out=line25 );
%CREATE_LINE( lbl="Total cumulative dose GA101 (mg)"                                                  ,data_out=line26 );
%CREATE_LINE( lbl="  Mean"    ,var=cumdosg  ,stat=MEAN                        ,fmtnum=8.1            ,data_out=line27 );
%CREATE_LINE( lbl="  Std"     ,var=cumdosg  ,stat=STD                         ,fmtnum=9.2            ,data_out=line28 );
%CREATE_LINE( lbl="  SEM"     ,var=cumdosg  ,stat=STDMEAN                     ,fmtnum=9.2            ,data_out=line29 );
%CREATE_LINE( lbl="  Median"  ,var=cumdosg  ,stat=MEDIAN                      ,fmtnum=8.1            ,data_out=line30 );
%CREATE_LINE( lbl="  Min"     ,var=cumdosg  ,stat=MIN                         ,fmtnum=8.2            ,data_out=line31 );
%CREATE_LINE( lbl="  Max"     ,var=cumdosg  ,stat=MAX                         ,fmtnum=8.2            ,data_out=line32 );
%CREATE_LINE( lbl="  n"       ,var=cumdosg  ,stat=N                           ,fmtnum=6.             ,data_out=line33 );
%CREATE_LINE( lbl=" "                                                                                 ,data_out=line34 );
%CREATE_LINE( lbl="Total cumulative dose Rituximab (mg)"                                              ,data_out=line35 );
%CREATE_LINE( lbl="  Mean"    ,var=cumdosr  ,stat=MEAN                        ,fmtnum=8.1            ,data_out=line36 );
%CREATE_LINE( lbl="  Std"     ,var=cumdosr  ,stat=STD                         ,fmtnum=9.2            ,data_out=line37 );
%CREATE_LINE( lbl="  SEM"     ,var=cumdosr  ,stat=STDMEAN                     ,fmtnum=9.2            ,data_out=line38 );
%CREATE_LINE( lbl="  Median"  ,var=cumdosr  ,stat=MEDIAN                      ,fmtnum=8.1            ,data_out=line39 );
%CREATE_LINE( lbl="  Min"     ,var=cumdosr  ,stat=MIN                         ,fmtnum=8.2            ,data_out=line40 );
%CREATE_LINE( lbl="  Max"     ,var=cumdosr  ,stat=MAX                         ,fmtnum=8.2            ,data_out=line41 );
%CREATE_LINE( lbl="  n"       ,var=cumdosr  ,stat=N                           ,fmtnum=6.             ,data_out=line42 );

/*
|| The datasets WORK.LINEi are now combined together in a data step and the dataset WORK._FINAL is created:
*/
DATA _final;
  SET %DO i=1 %TO 42;
        line&i
      %END;;;;;
RUN;

/*
|| The final summary report is produced from the observations of the dataset WORK._FINAL where each observation
|| of the WORK._FINAL corresponds to one row of the body of the final summary report.
*/
```

Advantages of Non-Executable Code

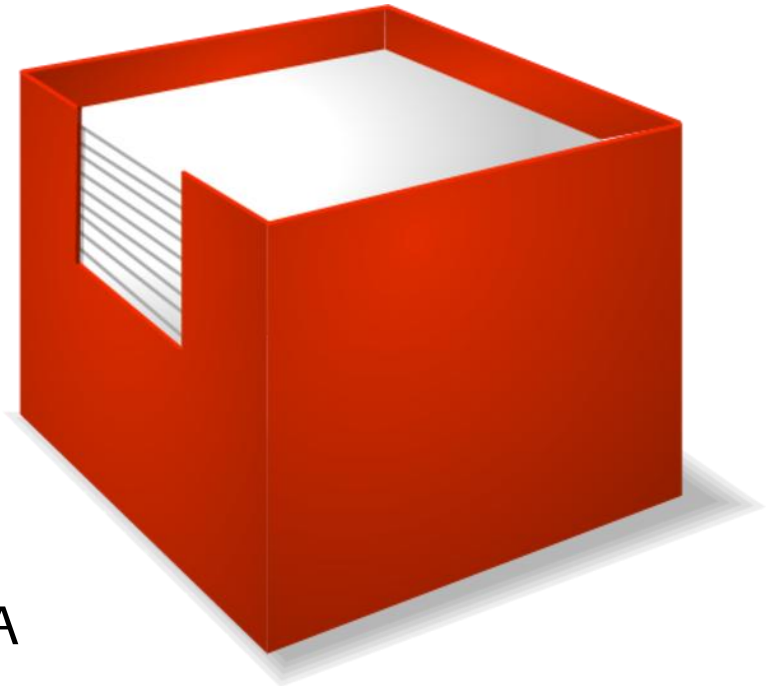
- Non-executable code easier to understand
- Less lines of code (executable code includes sponsor specific reporting macros)
- Sometimes thousands lines of code in stand-alone programs
- Non-executable files restricted to the important pieces of code
- Commenting on readable code can be done
- No added value for stand-alone programs because the result will be as on the output produced
- Often the QC program is used in order to provide executable code

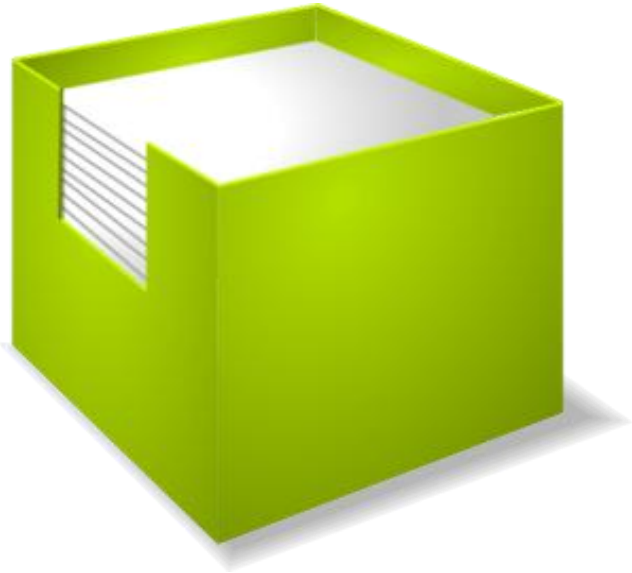
Summary / Experiences

- The Agency accepted the provided files
- No questions / comments received on these files
- Files were carefully checked and reviewed / validated by 2nd line code review
- Described in Reviewer's Guide for the FDA
- Further non-executable code provided in FDA questions after submission (reproduction of numbers)
- Very structured due to "fake" programming system "language"
- All numbers in the drug label are derived in provided non-executable-code
- Executable files at that time still delivered on top of non-executable files (might not be needed anymore?!)
- In non-executable-files all sponsor system references were removed (libnames etc)
- The sponsors needs to be ready for questions related to non-executable code

Outlook

- What is your experience?
- Will it be possible to come up with a standardized way?





- Q&A

Doing now what patients need next