



»» TRANSFORMING PROMISING IDEAS INTO COMMERCIAL REALITY

Automated generation of program documentation by means of tagged SAS comments and metadata for integrated analysis

Thomas Wollseifen - London PhUSE 13-15 Oct 2014



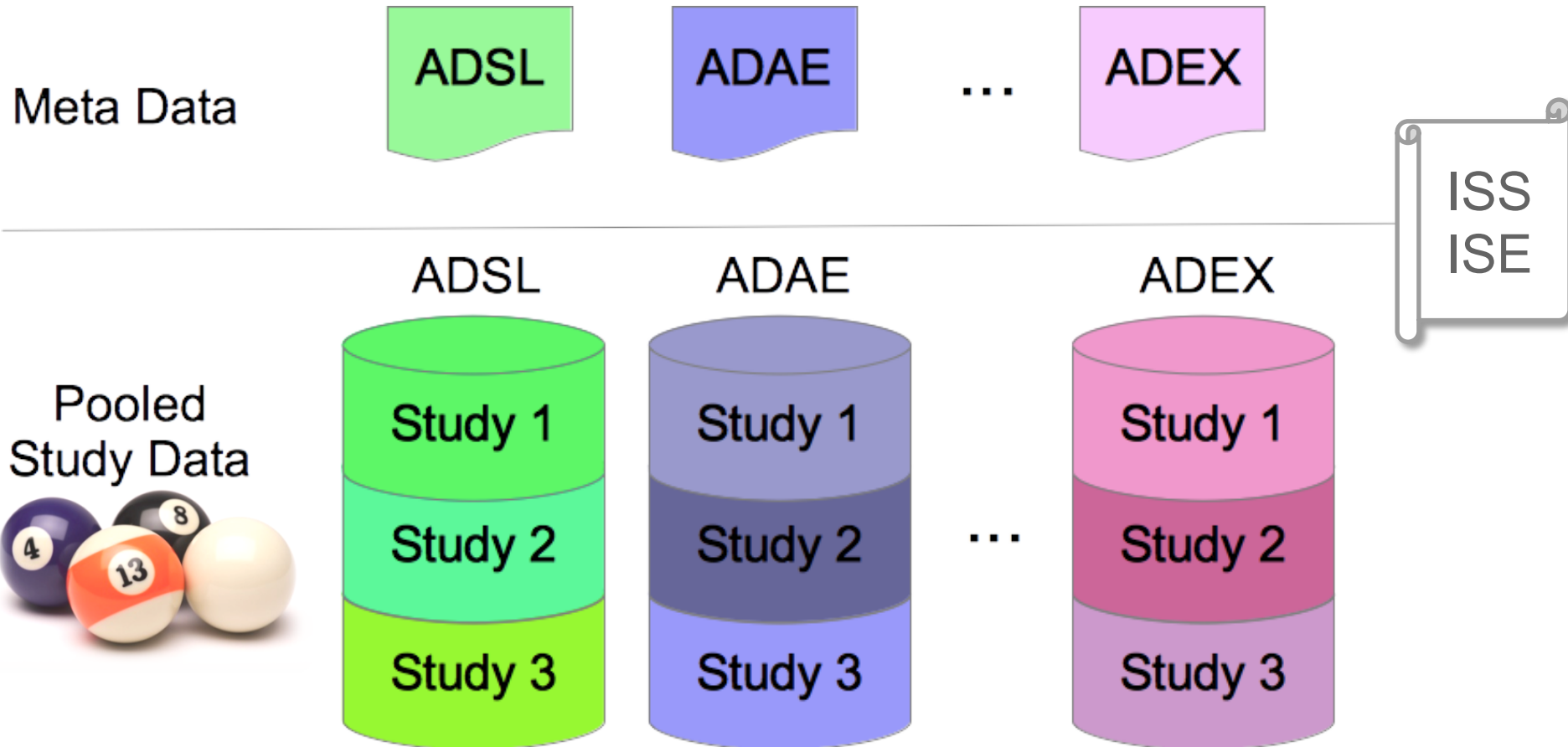


Contents

1. Introduction to integrated databases
2. Documentation of transfer programs
3. Automating documentation process
4. Summary



Integrated database of clinical studies







Documentation of transfer programs

Transfer program for domain

```
113 data &domain.;
114   set &domain.;
115   if base=. and ablf1='Y' then base=aval;
116   if chg =. and base^=. and aval^=. then chg=round(aval-base,0.0001);
117   if pchg=. and base^=. and aval^=. then pchg=round((aval-base)/base*100,0.01);
118   if basec=' ' and base^=. then basec=put(base,best.-1);
119   if chgc=' ' and chg^=. then chgc=put(chg,best.-1);
120   if pchg ne . then pchg = round(pchg, 0.01);
121   egady=egdy;
122   if egtested in ('QTCB','QTCF') then do;
123     if .<aval<=450 then avalct1n=1;
124     else if 450<aval<=480 then avalct1n=2;
125     else if 480<aval<=500 then avalct1n=4;
126     else if aval> 500 then avalct1n=6;
127     if not missing(avalct1n) then avalcat1=put(avalct1n,egavc.);
128
129     if .<chg<=30 then chgcat1n=1;
130     else if 30<chg<=60 then chgcat1n=2;
131     else if chg> 60 then chgcat1n=4;
132     if not missing(chgcat1n) then chgcat1=put(chgcat1n,egchgc.);
133   end;
134
135   param=put(paramcd,Segpar.);
136
137   if paramcd='PRMEAN' then do;
138     paramcd='4';
139     param=put(paramcd, zegparam.);
140   end;
141   else if paramcd='QRS' then do;
142     paramcd='5';
143     param=put(paramcd, zegparam.);
144   end;
145   else if paramcd='QTCB' then do;
146     paramcd='1';
147     param=put(paramcd, zegparam.);
148   end;
149   else if paramcd='QTCF' then do;
150     paramcd='2';
151     param=put(paramcd, zegparam.);
152   end;
153   else if paramcd='QTMEN' then do;
154     paramcd='6';
155     param=put(paramcd, zegparam.);
```

**Insert
Derivations
Manually/
automatically**

Documentation

Specification

Metadata
Pool

Metadata
Study

ADSL Subject-Level Analysis Dataset							
Variable name	LABEL	type AND length	format	Format decodes	Origin	Role	Comment
STUDYID	Study Identifier	Char 10	\$10.				1000: ADSL STUDYID
ADSNM	Dataset Name	Char 8	\$ADSNM.	ADSL=ADSL			1000: ADSL ADSNAME
USUBJID	Unique Subject Identifier	Char 20	\$20.				1000: ADSL USUBJID
AGE	Age	Num 8	3.				1000: ADSL AGE
AGEU	Age Units	Char 6	\$AGEU.	YEARS=YEARS			1000: ADSL AGEU
BRTH_D	Birth Day	Num 8	2.				1000: BRTH_D is derived from BIRTHDTH
BRTH_M	Birth Month	Num 8	2.				1000: BRTH_M is derived from BIRTHDTH
BRTH_Y	Birth Year	Num 8	2.				1000: ADSL BRTH_Y is derived from BIRTHDTH
<variable>	<label>	<type>	<format>	<decodes>			<studyid> <domain> <variable> [comment/derivation texts]

/* Comments ? */



Documentation of transfer programs

ADSL Subject-Level Analysis Dataset

Variable name	LABEL	type AND length	format	Format decodes	Origin	Role	Comment
STUDYID	Study Identifier	Char 10	\$10.				1000: ADSL.STUDYID
ADSNAME	Dataset Name	Char 8	\$ADSNM.	ADSL=ADSL			1000: ADSL.ADSNAME
USUBJID	Unique Subject Identifier	Char 20	\$20.				1000: ADSL.USUBJID
AGE	Age	Num 8	3.				1000: ADSL.AGE
AGEU	Age Units	Char 6	\$AGEU.	YEARS=YEARS			1000: ADSL.AGEU
BRTH_D	Birth Day	Num 8	2.				1000: BRTH_D is derived from BIRTHDTH
BRTH_M	Birth Month	Num 8	2.				1000: BRTH_M is derived from BIRTHDTH
BRTH_Y	Birth Year	Num 8	2.				1000: ADSL.BRTH_Y is derived from BIRTHDTH
<variable>	<label>	<type>	<format>	<decodes>			<studyid> <domain>.<variable> [comment/derivation texts]

Meta data

ADSL



```
%let domain=a
```

```
/* read data */
```

```
data &domain;
```

```
set source.&domain;
```

```
/** BRTH_Y is derived from BIRTHDT */  
if not(missing(birthdt)) then brth_y=year(birthdt);  
else brth_y=.;
```

```
*** BRTH_D is derived from BIRTHDT ***;  
if not(missing(birthdt)) then brth_d=day(birthdt);  
else brth_d=.;
```

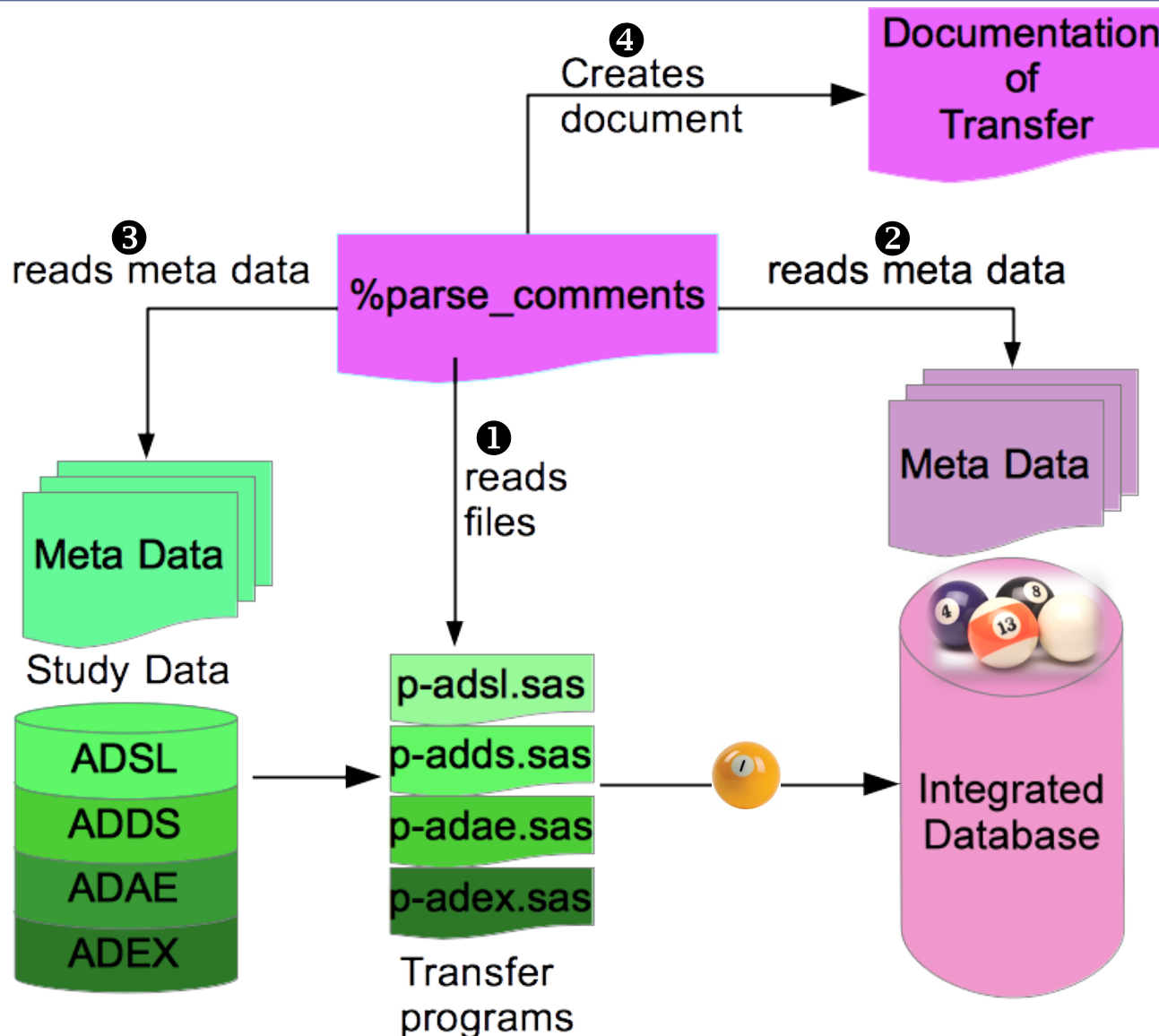
```
/** BRTH_M is derived from BIRTHDT ***/  
if not(missing(birthdt)) then brth_m=month(birthdt);
```

Insert
Derivations

Manually/
automatically

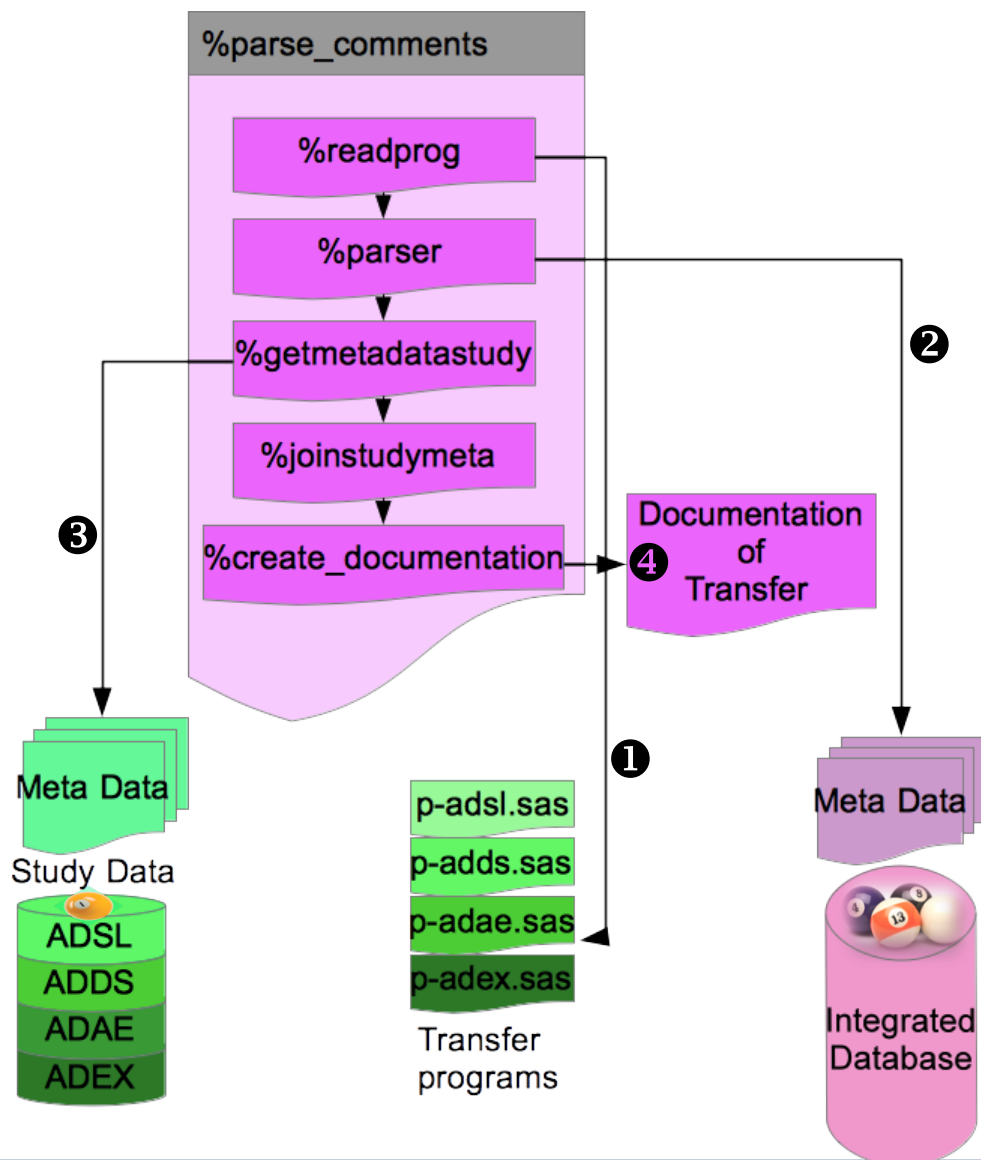


Automated generation of program documentation





Macro %parse_comments





Details of the %parser_comments

```
%parse_comments(studyid=, domain=, meta=);
```

```
%do st=1 %to &studynum;
```

```
  %varcount(list=&domain);
```

```
  %do d=1 %to &var_num;
```

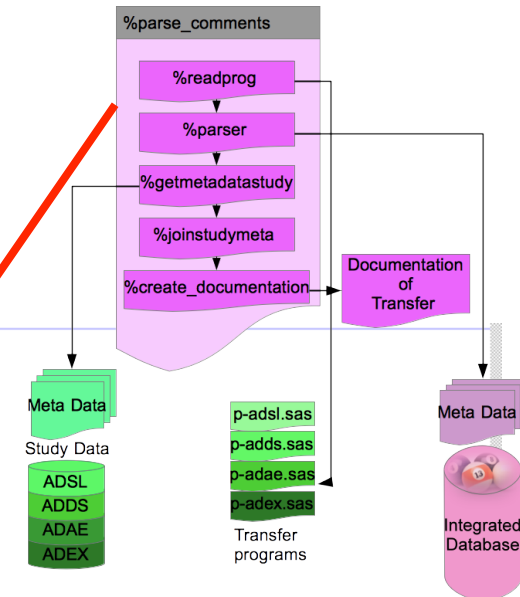
```
    %let _domain= %left(%qscan(&domain.,&d,"#"));
```

```
    %put domain=&_domain.;
```

```
  %readprog(study=&studyidm., domain=&_domain, pathr=&_pathtransfer, outdat=_&studyidm);
  %parser(study=&studyidm, metalib=meta, domain=&_domain);
  %getmetadastudy(study=&studyidm, libstudy=in, domain=&_domain);
  %joinstudymeta(study=&studyidm, metalib=meta, domain=&_domain);
  %create_documentation(study=&studyidm, metalib=&meta, domain=&_domain);
```

```
%end;
```

```
%end;
```



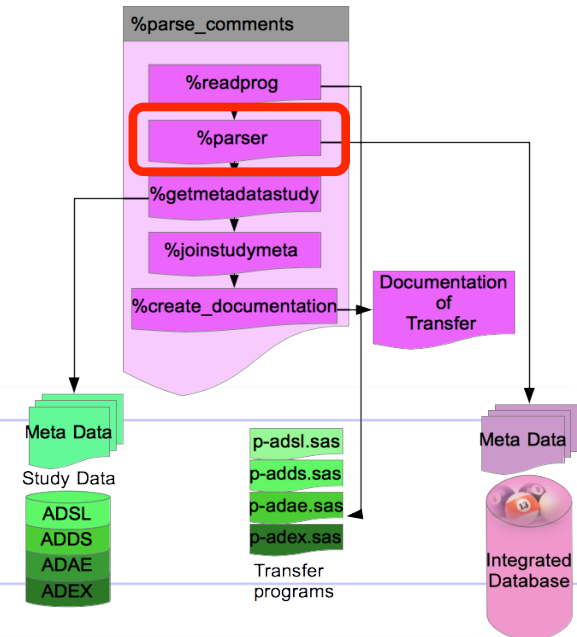


Details of the %parser

```
%parser(study=, metalib=, domain=);
```

```
_c1=prxparse("/\w|<end>/");  
_c2=prxparse("/\*\\//");
```

```
%do i=1 %to &_varcountdomain.;  
  pattern=prxparse("/&&_vardomain&i/");  
  lengthcomment=length(comment);  
  if prxmatch(pattern,comment) >0 then do;  
    if missing(foundmatch1) then do;  
      foundmatch1=prxmatch(pattern,comment);  
      foundmatch1c="&&_vardomain&i: " || compbl(left(substr(comment,foundm  
      sasname1="&&_vardomain&i");  
    end;  
  end;  
%end;
```





Documentation files generated by %parse_comments

```
%let domain=adsl;  
*****  
* read data and prepare derivations */  
*****  
  
data adslmeta;  
set source.<domain>;  
  
*** BIRTH_Y is derived from BIRTHDT **/  
if not(missing(birthdt)) then birth_y=year(birthdt);  
else birth_y=-;  
  
*** BIRTH_D is derived from BIRTHDT **/  
if not(missing(birthdt)) then birth_d=day(birthdt);  
else birth_d=-;  
  
*** BIRTH_M is derived from BIRTHDT **/  
if not(missing(birthdt)) then birth_m=month(birthdt);  
else birth_m=-;  
  
***BASEBL is available in study data,  
rounded to 1 decimal place  
-end-  
***/  
if not(missing(basebl)) then basebl=round(basebl,0.1);
```

%parse_comments

%readprog

%parser

%getmetadastudy

%joinstudymeta

%create_documentation

%parse_comments(studyid =1000#1001#1002,
domain =adsl#adex,
meta =meta);

Variable name	Label AND length	format	Format decodes	Origin	Role	Comment
<variable>	<label>	<type>	<format>	<decodes>		<studyid 1:> <domain>.<variable> [comment/derivation texts] <studyid 2:> <domain>.<variable> [comment/derivation texts] <studyid 3:> <domain>.<variable> [comment/derivation texts] ---



Syntax for commenting derivations needed by %parser

/* VARIABLE1:<derivation text> */

<program code>

**/* VARIABLE2:<derivation text line1
derivation text line2>**

...

<end>

***/**

<program code>

```
%let domain=adsl;
```

```
/* read data and prepare derivations */
```

```
data &domain;
```

```
set source.&domain;
```

```
/** BRTH_Y is derived from BIRTHDT */
```

```
if not(missing(birthdt)) then brth_y=year(birthdt);  
else brth_y=.;
```

```
*** BRTH_D is derived from BIRTHDT ***;
```

```
if not(missing(birthdt)) then brth_d=day(birthdt);  
else brth_d=.;
```

```
/** BRTH_M is derived from BIRTHDT */
```

```
if not(missing(birthdt)) then brth_m=month(birthdt);  
else brth_m=.;
```

```
/** BASEBMI is available in study data,  
rounded to 1 decimal place  
<end>
```

```
***/
```

```
if not(missing(basebmi)) then basebmi=round(basebmi,0.1);
```



Comments in SAS programs

Single-line

```
/*    Comment version 1 */  
*    Comment version 2 ;  
%*    Comment version 3 ;
```

Multi-line

```
/*  
    Common multi-line comment style  
    Comment 1  
    Comment 2  
*/
```

Nested

```
/* Comment /* Text */ ..... */
```



Example

```
%parse_comments(studyid=1000#1001#1002, domain=ALL, meta=meta);
```

Study 1000 p-adsl.sas

```
***BASEBMI is available in study data,  
    rounded to 1 decimal place  
    comment 1  
    comment 2  
    comment 3  
  
<end>  
    no comment for variable  
***;  
if not(missing(basebmi)) then
```

Study 1001 p-adsl.sas

```
***BASEBMI is rounded  
    to 1 decimal place  
    comment 1  
  
<end>  
***;  
if not(missing(basebmi)) then basebmi=round(basebmi,0.1)
```

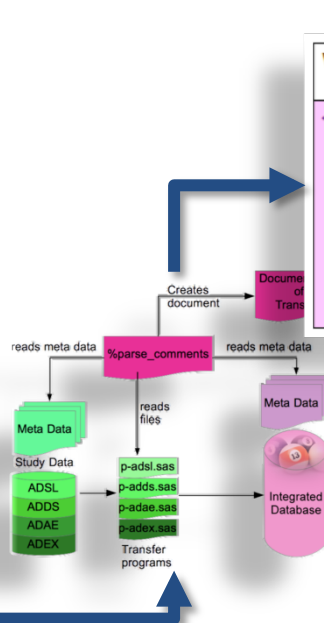
Variable name	Label	Type and length	Format	Decodes	Origin	Role	Comment
BASEBMI	Baseline Body Mass Index (kg/m2)	Num 8	5.1		derived		1000: BASEBMI/BASEBMI is available in study data, rounded to 1 decimal place comment 1 comment 2 comment 3 1001: BASEBMI/BASEBMI is rounded to 1 decimal place comment 1 1002: BASEBMI/BASEBMI is recalculated as : bodyweight / height (m) * height (m)



Summary

- Documentation makes programs more readable for reviewers/other programmers
- **%parse_comments** automates process of documentation
- Combines program source with documentation files

```
%let domain=adsl;  
  
/* read data and prepare derivations */  
data &domain;  
  set source.&domain;  
  
  /** BRTH_Y is derived from BIRTHDT */  
  if not(missing(birthdt)) then brth_y=year(birthdt);  
  else brth_y=.;  
  
  *** BRTH_D is derived from BIRTHDT ***  
  if not(missing(birthdt)) then brth_d=day(birthdt);  
  else brth_d=.;  
  
  /** BRTH_M is derived from BIRTHDT */  
  if not(missing(birthdt)) then brth_m=month(birthdt);  
  else brth_m=.;  
  
  /**BASEBMI is available in study data,  
  rounded to 1 decimal place  
  <end>  
  */  
  if not(missing(basebmi)) then basebmi=round(basebmi,0.1);
```



Variable name	LABEL	type AND length	format	Format decodes	Origin	Role	Comment
<variable>	<label>	<type>	<format>	<decodes>			<studyid 1:> <domain>.<variable> [comment/derivation texts] <studyid 2:> <domain>.<variable> [comment/derivation texts] <studyid 3:> <domain>.<variable> [comment/derivation texts] ...





Comments in program files





Contact information

Thomas Wollseifen

Principal Statistical Programmer

inVentiv Health Germany GmbH

Grosse Hub 10d, 65344 Eltville, Germany

T +49 (0) 6123 7043710

thomas.wollseifen@inventivhealth.com



<http://www.inVentivHealthclinical.com>