Using SAS® Macros to Simplify Preparation of SDTM Data, Annotated CRFs and Define.xml

PhUse 2009, Basel

Niels Both
Principal Consultant
S-Cubed
Presentation - Overview

- **eCTD**
  - Overall Architecture
  - Define.xml
  - Annotated CRF

- **SDTM Generation**
  - SDTM Step 1: Mapping
  - SDTM Step 2: Generic Macros
  - Conclusion
Background - eCTD

Guidance for Industry
Providing Regulatory Submissions in Electronic Format — Human Pharmaceutical Product Applications and Related Submissions Using the eCTD Specifications

http://www.fda.gov/cder/regulatory/ersr/ectd.htm

The study data specification
eCTD Study Data

Data Tabulation Datasets - SDTM

Annotated Case Report Form

Case Report Tabulation, data definition specification Define.xml

Analysis Datasets - ADaM

Subject Profiles

Data Listings
eCTD Study Data

- Data Tabulation Datasets - SDTM
- Annotated Case Report Form
- Case Report Tabulation, data definition specification Define.xml
- Analysis Datasets - ADaM
eCTD Study Data

- Data Tabulation Datasets - SDTM
- Annotated Case Report Form
- Case Report Tabulation, data definition specification Define.xml
Presentation - Overview

- eCTD
- Overall Architecture
- Define.xml
- Annotated CRF
- SDTM Generation
  - SDTM Step 1: Mapping
  - SDTM Step 2: Generic Macros
  - Conclusion
Create Define.xml

SDTM

Traditional SAS metadata (variable names, dataset names etc).

Define.xml

Mapping Specifications

CRF Page numbers, comments regarding mappings and derivations
Annotate CRF with SAS

MORE INFORMATION: “Using SAS to Speed up Annotating Case Report Forms in PDF Format prepared by Dirk Spruck and Monika Kawohl (PharmaSug 2004 paper CC02)”
Presentation - Overview

- eCTD
- Overall Architecture
- Define.xml
- Annotated CRF

SDTM Generation

- SDTM Step 1: Mapping
- SDTM Step 2: Generic Macros
- Conclusion
SDTM Generation

**Step 1**
- Source Data
- SAS Metadata Engine
- Pre SDTM

**Step 2**
- SAS SDTM Macros
- SDTM

Mapping Specifications
SDTM Generation

Step 1

- Use ETL tool ie SAS (Clinical) Data Integration Studio
- Use traditional SAS code
- Use spreadsheet with mappings entered to auto-generate the transforming SAS program
## SDTM Generation

### Step 2

<table>
<thead>
<tr>
<th>Macro name</th>
<th>Macro name</th>
</tr>
</thead>
<tbody>
<tr>
<td>%Usubjid</td>
<td>%Relrec</td>
</tr>
<tr>
<td>%Domain</td>
<td>%Studyid</td>
</tr>
<tr>
<td>%Dtc</td>
<td>%BI</td>
</tr>
<tr>
<td>%Dy</td>
<td>%Stc</td>
</tr>
<tr>
<td>%Seq</td>
<td>%Res</td>
</tr>
<tr>
<td>%LongChars</td>
<td>%SDTMify</td>
</tr>
<tr>
<td>%SuppqualCo</td>
<td>%PhysicalTransform</td>
</tr>
</tbody>
</table>
%Usubjid

Creates the USUBJID variable (Unique subject ID) as a concatenation of SUBJID and Study ID
SDTM Generation – Macros (2/14)

%Domain

Creates the variable DOMAIN for all Pre SDTM datasets
SDTM Generation – Macros (3/14)

%DTC

Creates all --DTC variables in ISO8601 format based on date/time components stored as individual variables in Pre SDTM
SDTM Generation – Macros (4/14)

%Dy

Creates all --DY variables for each --DTC variable based on the RFSTDTC variable in the DM domain. The algorithm used is:

\[-DY = (date \text{ portion of } --DTC) - (date \text{ portion of } RFSTDTC) + 1 \text{ if } --DTC \text{ is on or after } RFSTDTC\]

\[-DY = (date \text{ portion of } --DTC) - (date \text{ portion of } RFSTDTC) \text{ if } --DTC \text{ precedes } RFSTDTC\]
%Seq

Sorts datasets and assigns --SEQ variables as a unique sequence number within each USUBJID
%LongChars

Identifies text strings longer than 200 characters and splits them up into multiple variables
SDTM Generation – Macros (7/14)

%SuppqualCo

Creates supplementary datasets from extra columns on Pre SDTM datasets

Also creates CO datasets where relevant
### SDTM Generation – Macros (7/14)

#### AE (Pre SDTM)

<table>
<thead>
<tr>
<th>STUDYID</th>
<th>DOMAIN</th>
<th>USUBJID</th>
<th>AESEQ</th>
<th>AETERM</th>
<th>AESER</th>
<th>...</th>
<th>SQ_AESOTHYC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCUB01</td>
<td>AE</td>
<td>123</td>
<td>1</td>
<td>TRAUMA</td>
<td>Y</td>
<td></td>
<td>Neck Injury</td>
</tr>
</tbody>
</table>

#### SUPPAE

<table>
<thead>
<tr>
<th>STUDYID</th>
<th>RDORAIN</th>
<th>USUBJID</th>
<th>IDVAR</th>
<th>IDVARVAL</th>
<th>QNAM</th>
<th>AESOTHYC</th>
<th>QVAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCUB01</td>
<td>AE</td>
<td>123</td>
<td>AESEQ</td>
<td>1</td>
<td>AESOTHYC</td>
<td>Desc of Other Event</td>
<td>Neck Injury</td>
</tr>
</tbody>
</table>

#### AE (SDTM)

<table>
<thead>
<tr>
<th>STUDYID</th>
<th>DOMAIN</th>
<th>USUBJID</th>
<th>AESEQ</th>
<th>AETERM</th>
<th>AESER</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCUB01</td>
<td>AE</td>
<td>123</td>
<td>1</td>
<td>TRAUMA</td>
<td>Y</td>
<td></td>
</tr>
</tbody>
</table>
SDTM Generation – Macros (8/14)

%Relrec

Creates RELREC dataset describing relationships between records in Pre SDTM datasets
Relrec example:

CM number 11&12 is related to AE number 3

<table>
<thead>
<tr>
<th>RELREC</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>STUDYID</td>
<td>RDOMAIN</td>
<td>USUBJID</td>
<td>IDVAR</td>
<td>IDVARVAL</td>
<td>RELTYPE</td>
<td>RELID</td>
</tr>
<tr>
<td>SCUB01</td>
<td>AE</td>
<td>123</td>
<td>AESEQ</td>
<td>3</td>
<td>AECM1</td>
<td></td>
</tr>
<tr>
<td>SCUB01</td>
<td>CM</td>
<td>123</td>
<td>CMSEQ</td>
<td>11</td>
<td>AECM1</td>
<td></td>
</tr>
<tr>
<td>SCUB01</td>
<td>CM</td>
<td>123</td>
<td>CMSEQ</td>
<td>12</td>
<td>AECM1</td>
<td></td>
</tr>
</tbody>
</table>
SDTM Generation – Macros (8/14)

Relrec example:

AE number 3 is related to CM number 11&12

Pre SDTM CM dataset:

<table>
<thead>
<tr>
<th>STUDYID</th>
<th>DOMAIN</th>
<th>USUBJID</th>
<th>CMSEQ</th>
<th>RELREC_AE</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCUB01</td>
<td>CM</td>
<td>123</td>
<td>11</td>
<td>AESPID=3</td>
<td></td>
</tr>
<tr>
<td>SCUB01</td>
<td>CM</td>
<td>123</td>
<td>12</td>
<td>AESPID=3</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

The macro identifies the variables starting with 'RELREC-', then finds the associated records in the AE datasets, and based on that creates the 3 RELREC records.
SDTM Generation – Macros (9/14)

%SStudyid

Creates the variable STUDYID with the relevant study ID based on what is assigned in the Pre SDTM DM domain
SDTM Generation – Macros (10/14)

%Bl

Creates the --BL variables, based on a defined algorithm (for example, by flagging the last assessment prior to RFSTDTC as the baseline value)

Note:
The existence of this statistically derived variable in the SDTM model is under review by the CDISC teams and the variable is expected to disappear from the SDTM model at some stage. Until that happens it is suggested to use a crude method like the above to set it and to note in the define document that it will not in all cases match the advanced baseline flags in the ADaM datasets.
%Stc

Creates the various results variables in the finding domains, based on the Pre SDTM variables --ORRES and --ORESSU (original result and unit)
%Res

Checks if the value of --STRESC is in fact numeric. In that case it is transferred to --STRESN
SDTM Generation – Macros (13/14)

%SDTMify

Assigns labels and exact formats to all the variables in the created datasets and sorts dataset records and columns according to agreed rules.
SDTM Generation – Macros (14/14)

%PhysicalTransform

Turns the created SAS datasets into SAS transport files
Presentation - Overview

- eCTD
- Overall Architecture
- Define.xml
- Annotated CRF
- SDTM Generation
  - SDTM Step 1: Mapping
  - SDTM Step 2: Generic Macros
- Conclusion
SDTM Generation

Step 1

Source Data

SAS Metadata Engine

Pre SDTM

Step 2

SAS SDTM Macros

SDTM

Mapping Specifications