Producing a Format Library and Test Data for Case Report Forms using a Data Definition Table
Producing a Format Library and Test Data for Case Report Forms using a Data Definition Table

Suad El Burai Félix, MPH1, Ellen J. DeMatt, MA1, Rebecca A. Horney, BA1
1Cooperative Studies Program Coordination Center, Office of Research & Development, U.S. Department of Veterans Affairs, Perry Point, MD

ABSTRACT
In Clinical Trials, a Data Definition Table (DDT) is a document that lists a variety of information about each case report form (CRF), such as form number, variable name, variable label or description, data type, length, and codes. When imported into SAS® it can be used to accomplish a variety of tasks. We aim to provide a workflow for creating a format library and test data for CRFs by using SAS and the DDT in Microsoft Excel. Both creating a format library and test data are necessary pieces to gather at the beginning of a study to eliminate the need of re-creating formats in each of your SAS programs (reduce code replication) and to start programming when real study data is not available yet (quicker delivery reporting time), respectively.

INTRODUCTION
Statistical programmers have the vital role of developing SAS coding in early stages of a study to ensure efficient and timely implementation of analyses. Consequently, the prompt development of SAS programs will allow being “ahead of the game” by becoming familiarized with the database (i.e., case report forms) content and producing reports, tables and listings in a timely manner. Two SAS programming practices that a statistical programmer can use to safeguard efficient and timely implementation of analyses are creating a centralized format library and test data.

We aim to describe a SAS macro that reads in an Excel format DDT file that comes from case report forms, which creates SAS data sets. In addition, we will walk you through the process of using SAS programs that utilize these SAS datasets from the macro to actually generate a format library and test data.

SETTING UP THE EXCEL DDT BEFORE RUNNING THE MACRO
In order to use the SAS macro that reads in the Microsoft Excel formatted DDT, the DDT must be one workbook containing a separate worksheet/tab for each case report form. The DDT in appendix A illustrates the columns needed in each DDT worksheet.

RUNNING THE MACRO TO READ IN THE DDT
The macro Read_DDT (appendix B) takes the data from the DDT and creates two SAS datasets per case report form (i.e., Fxx and Fxx_formats – where “xx” represents the form number). These datasets will be used to create a format library and test data. The following code calls the Read_DDT macro.

CREATING A FORMAT LIBRARY
Run the program below, which uses the SAS datasets Fxx_formats created when running the macro to read in the DDT. This will generate a new SAS program called ProcFormat.sas located in the directory folder specified in the %let outdir statement.

DATA allfmt_data;
  length label $300 form $8;
  set ddt.fxx_formats
  ddt.fxx_formats; *Continue for each CRF form as needed;
RUN;

PROC SORT data = allfmt_data; by form fmtname; RUN;
CREATING A FORMAT LIBRARY (CONT.)

DATA allformats;
  set allfmt_data end=eof;
  by form fmtname;
  file "&outdir.ProcFormat.sas";
  retain val;
  if _n_ = 1 then do;
    put "libname outdir 'C:\SAS\Formats\';";
    put;
    put "PROC FORMAT library = outdir cntlout = outdir.Formats;";
  end;
  if first.fmtname and last.fmtname then do;
    len=length(fmtname); val=len+11;
    put/ @3 "value " @9 fmtname @val value @val+5 "= " @val+8 label +(-1) "';";
  end;
  else if first.fmtname then do;
    len=length(fmtname); val=len+11;
    put/ @3 "value " @9 fmtname @val value @val+5 "= " @val+8 label +(-1) "';";
  end;
  else if last.fmtname then do;
    put/ @3 "value " @9 fmtname @val value @val+5 "= " @val+8 label +(-1) "';";
  end;
  if eof then put "RUN;"
RUN;

After the proc format program is generated, run it to create a centralized format library. A format will now be assigned to each codelist and checkbox variable.

CREATING TEST DATA

Once the format library is created, run the macro TestData_CreateEmpty (see appendix C) which will output a new SAS program for each case report form called EmptyFormxx.sas located in the directory folder specified in the %let testdata statement. Run the following code to call the TestData_CreateEmpty macro.

%let blank = 999; *Blank value for checkboxes;
%let testdata = C:\SAS\TestData;

%%TestData_CreateEmpty(frm_num=xx);
%%TestData_CreateEmpty(frm_num=xx); *Continue for each CRF form as needed;

Execute the EmptyFormxx.sas programs that were created when calling the TestData_CreateEmpty macro to create an empty dataset for each form. Then run the TestData_CreateCode macro (see appendix D). This macro will output another SAS program for each case report form called TestData-Formxx.sas located in the directory folder specified in the %let testdata statement. Run the following code to call the TestData_CreateCode macro.

%let line1="seed=2794383;"
%%TestData_CreateCode(frm_num=xx);
%let line1="seed=2694383;"
%%TestData_CreateCode(frm_num=xx); *Continue for each CRF form as needed;

The programs TestData-Formxx.sas created from the code above can be run to populate the empty datasets with test data.
Producing a Format Library and Test Data for Case Report Forms using a Data Definition Table

Suad El Burai Félix, MPH\(^1\), Ellen J. DeMatt, MA\(^1\), Rebecca A. Horney, BA\(^1\)

\(^1\)Cooperative Studies Program Coordination Center, Office of Research & Development, U.S. Department of Veterans Affairs, Perry Point, MD

### APPENDIX A

<table>
<thead>
<tr>
<th>FORM FIELD NAME</th>
<th>DATA ELEMENT TYPE</th>
<th>LENGTH</th>
<th>FIELD LABEL</th>
<th>MIN</th>
<th>MAX</th>
<th>DEPENDENT ON</th>
<th>IF VALUE</th>
<th>FIELD #</th>
</tr>
</thead>
<tbody>
<tr>
<td>CENTER</td>
<td>Integer</td>
<td>3</td>
<td>Title #</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>PARTNO</td>
<td>Integer</td>
<td>4</td>
<td>Participant ID</td>
<td>1001</td>
<td>1200</td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>ALPHA</td>
<td>Char</td>
<td>4</td>
<td>Alpha Code</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VISITDATE</td>
<td>Date</td>
<td>10</td>
<td>Data Points Completed</td>
<td>7/1/2000</td>
<td>7/27/2018</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOB</td>
<td>Date</td>
<td>10</td>
<td>Date of Birth</td>
<td>7/1/2000</td>
<td>7/27/1945</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HEIGHT</td>
<td>Integer</td>
<td>2</td>
<td>Height (inches)</td>
<td>62</td>
<td>76</td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>WEIGHT</td>
<td>Integer</td>
<td>3</td>
<td>Weight (pounds)</td>
<td>100</td>
<td>200</td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>SEX</td>
<td>Gender</td>
<td>3</td>
<td></td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>RACE</td>
<td>Other</td>
<td>20</td>
<td>Other Race, Specify</td>
<td>RACE</td>
<td>6</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARMY</td>
<td>Army</td>
<td>999</td>
<td>Go, Army</td>
<td>1</td>
<td>11</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>AIRFORCE</td>
<td>Airforce</td>
<td>999</td>
<td>Go, Airforce</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>NAVY</td>
<td>Navy</td>
<td>999</td>
<td>Go, Navy</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>MARINES</td>
<td>Marines</td>
<td>999</td>
<td>Go, Marines</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
</tr>
</tbody>
</table>

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.

### APPENDIX B
Producing a Format Library and Test Data for Case Report Forms using a Data Definition Table

Suad El Burai Félix, MPH¹, Ellen J. DeMatt, MA¹, Rebecca A. Horney, BA¹
¹Cooperative Studies Program Coordination Center, Office of Research & Development, U.S. Department of Veterans Affairs, Perry Point, MD

APPENDIX D