Validation Techniques in Clinical Trials Using SAS Enterprise Guide
Venkata R. Darbha, Quintiles, Thousand Oaks, CA

ABSTRACT
SAS programmers in the pharmaceutical industry will find that SAS’s Enterprise Guide® makes it easier, quicker and efficient in validating SAS data sets, tables, and graphs. This paper presents some of the most commonly used edit checks during validation, using SAS Enterprise Guide. Comparing two data sets, creating summary statistics, creating frequency reports are few of the validation techniques that are addressed besides listing other tips and techniques. Demographic and Medical History example data sets will be used in this presentation. The paper will feature SAS Enterprise Guide® version 4.1.

INTRODUCTION
Validating clinical data, reports and graphs in the pharmaceutical industry has always played a significant role in delivering quality results. A common generic approach in statistical validation in clinical trails is to develop independent code as per the specifications and then cross validate the results. This technique however, is considered to be time and resource consuming. SAS Enterprise Guide® greatly reduces the complexity involved in the above drawback and provides simple, efficient and easy to use task orientated interface for validation. This paper discusses few validation techniques that are commonly used in the traditional programmatic approach such as creating formats, comparing data sets, writing SQL queries, frequency reports, characterize data etc. using Enterprise Guide®.

VALIDATION APPROACHES IN CLINICAL TRAILS
Demographic data for subjects in a clinical trail is considered in this paper. Below is a snapshot of 10 records extracted from a demographic data set (demo). There is one unique record per subject with only a few key variables chosen for the sake of simplicity.

<table>
<thead>
<tr>
<th>OBS</th>
<th>STUDY</th>
<th>SUBJECT</th>
<th>RACE</th>
<th>AGE</th>
<th>AGECODE</th>
<th>SEXCODE</th>
<th>SEX</th>
<th>FDRUGDT</th>
<th>EVALSAFE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20071018</td>
<td>5001</td>
<td>Caucasian</td>
<td>16</td>
<td>3</td>
<td>M</td>
<td>Male</td>
<td>1/7/2007</td>
<td>Y</td>
</tr>
<tr>
<td>2</td>
<td>20071018</td>
<td>5002</td>
<td>Caucasian</td>
<td>6</td>
<td>2</td>
<td>F</td>
<td>Female</td>
<td>1/21/2007</td>
<td>Y</td>
</tr>
<tr>
<td>3</td>
<td>20071018</td>
<td>5003</td>
<td>Caucasian</td>
<td>2</td>
<td>1</td>
<td>M</td>
<td>Male</td>
<td>3/4/2007</td>
<td>Y</td>
</tr>
<tr>
<td>4</td>
<td>20071018</td>
<td>5004</td>
<td>Hispanic</td>
<td>4</td>
<td>2</td>
<td>M</td>
<td>Male</td>
<td>9/2/2006</td>
<td>Y</td>
</tr>
<tr>
<td>5</td>
<td>20071018</td>
<td>5005</td>
<td>Hispanic</td>
<td>15</td>
<td>3</td>
<td>F</td>
<td>Female</td>
<td>1/14/2007</td>
<td>Y</td>
</tr>
<tr>
<td>6</td>
<td>20071018</td>
<td>5006</td>
<td>Other</td>
<td>1</td>
<td>1</td>
<td>M</td>
<td>Male</td>
<td>5/13/2007</td>
<td>Y</td>
</tr>
<tr>
<td>7</td>
<td>20071018</td>
<td>5007</td>
<td>Hispanic</td>
<td>7</td>
<td>2</td>
<td>M</td>
<td>Male</td>
<td>1/27/2007</td>
<td>Y</td>
</tr>
<tr>
<td>8</td>
<td>20071018</td>
<td>5008</td>
<td>Hispanic</td>
<td>1</td>
<td>1</td>
<td>M</td>
<td>Male</td>
<td>2/24/2007</td>
<td>N</td>
</tr>
<tr>
<td>9</td>
<td>20071018</td>
<td>5009</td>
<td>Hispanic</td>
<td>8</td>
<td>2</td>
<td>F</td>
<td>Female</td>
<td>4/14/2007</td>
<td>N</td>
</tr>
<tr>
<td>10</td>
<td>20071018</td>
<td>5010</td>
<td>Hispanic</td>
<td>15</td>
<td>3</td>
<td>F</td>
<td>Female</td>
<td>3/17/2007</td>
<td>Y</td>
</tr>
</tbody>
</table>

Figure 1.1 Example demographic data.

CREATING USER DEFINED FORMATS
Creating formats for the variables during a SAS data set validation scenario is necessary to compare with the variables generated during production programming. For example, consider the data in Figure 1.1. If you were to create variables such as ‘SEXCODE’ and ‘AGECODE’ for the subjects from their parent variables, ‘SEX’ and ‘AGE’, in the traditional approach, you would write a SAS code having a proc format and a data step to accomplish the task as shown below.

```sas
proc format ;
value $sex
'Male'='M'
'Female'='F';
value age
```
2

1-2='1'
3-11='2'
12-16='3';
run;

In Enterprise Guide®, you can quickly create either temporary or permanent formats by following the steps below. Referring to Figure 1.2a below

1. Choose 'Create Format' from the Task List menu.
2. Give the format name as 'sex' in the box under Format name, character push button in the Format type, and 'Local' and 'Work' in the Location of the format section.
3. Highlight Define Format in the selection pane and assign format labels (M and F) and values (Male and Female) by choosing New Label and New Range successively.
4. Clicking Run creates the format and is seen in the Project Designer window.
5. Assign the format to variable ‘sex’ by navigating to its properties in the update mode. You could as well assign this format to the derived variable ‘sexcode’ by first creating the variable using the wizard and then assigning the format by navigating to its properties.

Figure 1.2a Creating character format.

Figure 1.2b Creating numeric format.
Figure 1.2b depicts a similar way to create numeric format for three different age categories. Since the numeric variable format is designed to handle a range of values, **Range** should be selected from the pull down menu under **Type** in the **Define Formats** window.

Note: 1. To create a permanent format, choose any other library other than 'Work' in the Location of the format section.
2. Permanent formats created on a particular SAS server are only available for the tasks on that server.
3. There are a plenty of most commonly used character, numeric, date, currency formats which are built in within the format wizard in Enterprise Guide.

**COMPARING DATA SETS**

Proc **Compare** is a widely used procedure to compare and identify data level discrepancies in two data sets. For example, assume that

1. Data represented in Figure 1.1 (demo1) is the base data set and a competing demo data set (demo2) is created by a validation programmer.
2. There are however, differences in the corresponding variable values of the two data sets.

A traditional comparison code:

```sas
proc sort data=demo1 out=demo1;
    by study subject;
run;

proc sort data=demo2 out=demo2;
    by study subject;
run;

proc compare base=demo1 compare=demo2
            method=exact
            id study subject;
run;
```

Figure 1.3 Front end of Compare Procedure in Enterprise Guide.

Utilizing **Compare Data** wizard in Enterprise Guide, to implement exactly the same comparison intension of using tradition coding, resulted in producing exactly the same discrepancies as noted in assertion 2. However, the flexibility and simplicity in using Enterprise Guide for this technique can be summarized as:
1. Point and select option of selecting the ID variables in the compare wizard.
2. There is no need to exclusively sort the base and compare data sets by the ID variables before using the compare wizard. The compare wizard automatically does the sorting, thereby saving time.
3. Comparison criteria; Method of judging equality such as exact, absolute, percent, relative are provided as menu driven options.
4. Point and create output data set to include all observations from base data or from compare data or report only the discrepancy observations as a separate data set.
5. Besides creating a SAS listing of the compare results, Enterprise Guide by default creates output delivery such as HTML/RTF/PDF/TEXT of the compare results to a user defined destination. Note that in BASE SAS programming you have to exclusively invoke the ODS package.

REPLACING PROC SQL BY USING THE QUERY BUILDER

Query Builder in SAS Enterprise Guide is a handy tool that is extremely useful in data manipulations.

Significant tasks that can be accomplished using query wizard include:

1. Simple to complex SQL queries,
2. Data filtering and sorting.
3. Compute new and derived variables.
4. Join data sets.

For example, in the demo data set, you would want to accomplish the following

1. Create a new variable (USUBJECT) which is defined as a concatenation of STUDY and SUBJECT.
2. Select and list only the variables as seen in Figure1.1 along with the new computed variable USUBJECT.
A traditional approach using a Proc SQL would be

```sql
proc sql;
create table demo as select demo.study,
    demo.subject,
    (compress(study||'-'||subject)) as usubject,
    demo.birthdtv,
    (put(age,age.)) as agecode,
    demo.age format=best12.,
    demo.sex format=$8.,
    (put(sex, $sex.)) as sexcode,
    demo.fdrugdt format=date9.,
    demo.ldrugdt format=date9.
from demo as demo;
quit;
```

In Enterprise guide, you could use the Query Builder wizard to accomplish the above task with ease. Below are the steps that guide you through the process.

1. Chose Select Data from the right pane of the Query Builder.
2. Drag and Drop all the required variables for the listing from the left pane.
3. Choose Computed Columns from the horizontal Menu bar to initiate Computed Columns window.
4. Choose Build Expression under New which opens up the Advanced Expression Editor window.

![Figure 1.5 Computing new variables in the Query Builder.](image-url)
5. Write the expression in the **Expression Text** window as ‘compress (study||‘-‘||subject)’ and click OK.

6. This generates a new column as ‘computed1’ in the **Computed Columns** window. Rename the variable to USUBJECT.

7. Similarly, repeat the process to create variables SEXCODE and AGECODE and assign their respective user defined formats if available.

8. Preview the results, code, and log by choosing **Preview** from the horizontal **Menu** bar.

9. Click on the Run tab to run the query. This creates a query data set that includes all the variables intended for the listing including the computed variables.

10. Highlight the query data set and use the **List Data** wizard under **Describe** in the **Menu** bar to create a listing of the data set in any of the ODS file formats as required.

**Advantages of using Query Builder:**

1. Saves time by not having to exclusively write a query and not to worry about SQL syntax.
2. **Functions** tab in the **Advanced Expression Editor** provides you with abundant arithmetic, logical, character, numeric, date, time functions in a drive down menu for use in building any logical expression.
3. **Filter** and **Sort** data wizards inside the Query Builder enhances the scope of building complex queries.
4. Enterprise Guide generated SAS code available to understand the algorithm of the query flow.

**VALIDATING RTF LISTINGS**

SAS Enterprise Guide provides you with features and a variety of options to create RTF listings, summary statistics, and frequency reports etc. which are very common and quite often needed to be generated or validated in a clinical environment.

**List Data** wizard under the **Describe** tab in the **Menu** bar in Enterprise Guide provides a niche in generating SAS listings in a variety of ODS file formats. The features include:

1. Drag and drop flexibility to select reporting variables from the data set into the report.
2. Drag and drop group by, page by, totals, and subtotals of the variables into the report.

![Figure 1.6 List Data Wizard.](image-url)
CHARACTERIZE THE DATA
Characterizing clinical data by looking at the frequency summary report of all the key variables as well as the data's graphical representation at the same instance is possible using the Enterprise Guide. Preliminary insight of understanding clinical data distribution gives you the flexibility to clearly analyze the following:

1. The variables of statistical interest.
2. Type of statistical procedures that should be used.
3. Report discrepant data issues right away.

Characterize Data wizard accomplishes this very useful task of creating summary reports and histogram charts of all the character, numeric, and date variables in the input data set in a single wizard. This wizard is located in the Describe section of the Task List menu. Using the demo data in Figure 1.1

Following the simple instructions of this wizard, summary reports, graphs, and frequency and univariate data sets of the main characteristics of the demo data set were created. As you can notice from the above figure, the Characterize Data block has one input data set (demo) and three output blocks; Frequency counts data set, univariate data set, and an HTML report, that has the summary reports and graphs within.

Figure 1.7a Histogram of age category in demo.

Figure 1.7b Histogram of safety evaluable flag in demo.

Figure 1.7a and Figure 1.7b shows count of three age categories and safety evaluable flag in demo data set which were residing in the HTML report of the wizard. As you can see, it is clearly evident that the counts are appropriate from the data.
Key advantages of using Characterize Data wizard over developing SAS code to perform the same task are as follows:

1. Frequency counts and univariate statistics for all the variables in the data set are automatically delivered as a SAS data set.
2. Frequency summary reports and graphs are simultaneously generated.
3. No need to manually type code for SAS procedures such as Proc Freq, Proc Univariate, Proc Gchart etc to accomplish a similar task.
4. Detailed perception of the data within minutes.
5. Readily accept or reject data quality.
6. All in one utility package for clinical data management.

CONCLUSION
A number of validation techniques in clinical trails are successfully presented using wizards in SAS® Enterprise Guide®. Techniques on creating user defined formats, comparing data sets, data manipulation using queries, listings, frequency tables, graphs as well as addressing the advantages of using Enterprise Guide over traditional programmatic approach are presented. Automatic code generation for all the wizards is an applauded advantage that helps in having a much clearer understanding of a wizard. Numerous programmatic techniques in validating clinical data are replaceable using SAS Enterprise Guide. ‘Point and Click’ feature of all the wizards discussed in this paper is easy to use and saves a lot of time and resource in a clinical environment.

Usage of SAS Enterprise Guide® in pharmaceutical industries is very encouraging in recent times. However, there is a further need to standardize the use of this product in a regulatory validation environment.

REFERENCES
SAS Institute Inc.

ACKNOWLEDGEMENTS
The author wishes to acknowledge Sunil Gupta, whose paper on Enterprise Guide® presented at WUSS, 2006 prompted the idea for this paper.

CONTACT INFORMATION
Your comments and questions are valued and encouraged. Contact the author at:
Venkata R. Darbha
Quintiles, 325 E. Hillcrest Dr, Suite 200
Thousand Oaks, CA 91360
Work Phone: (805) 557-7720
E-mail: ravikanth.darbha@quintiles.com
Web: www.quintiles.com

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.