CompareXLS – a Powerful Comparison Tool for Spreadsheet Difference Resolution

Kirill Tchernakov, Boehringer Ingelheim Pharmaceutical, Inc., Ridgefield, CT
John H. Adams, Boehringer Ingelheim Pharmaceutical, Inc., Ridgefield, CT
Nancy G. Bauer, Boehringer Ingelheim Pharmaceutical, Inc., Ridgefield, CT

Abstract

All pharmaceutical companies are required by regulatory agencies to report the overall safety and efficacy drug profile resulting from years of data collection from all phases of clinical trials. If the substance was licensed in from another company or university that collected clinical data, then this data must be gathered and homogenized to fit into the company’s data structure. Likewise, if the drug under study is novel or a new therapeutic area for the company, the knowledge gained after each clinical trial results in a better understanding of the drug mechanism which in turn can modify the underlying data structure. In an effort to address the need for overall reporting as quick as possible and in a manner that allows the pooling of data from all phases of the drug development process, Boehringer Ingelheim has developed the “Extensible Project Database Loading and Table Translation Program” (XPDL) process (1). XPDL uses SAS® to remap and combine multiple data sources into one data structure called the Project Data Base (PDB). The process is dynamic and table driven. The tables are edited using Excel® and are maintained by Data Managers.

The recent enhancement to XPDL was the development of the CompareXLS SAS® macro program. This macro was written to ease the burden of comparing multiple Excel spreadsheets and resolving the differences between them. The spreadsheets contain metadata that documents the source-to-target relationships of the project database and serves as an input for building the project database using the XPDL ‘load program.’

The CompareXLS macro converts the spreadsheets into datasets and compares them by their variables and attributes, offering different ways of problem resolution. After the differences are resolved, CompareXLS can effectively transform the resulting dataset into a new spreadsheet. The CompareXLS macro has proven to be a very effective and efficient method when resolving multiple spreadsheets.

Introduction

The clinical evaluation of a chemical or biological entity must undergo rigorous testing and successfully complete each phase of the clinical development process in order to
build an overall safety and efficacy profile. The combined clinical trials from the various phases constitute a drug project. Source data from the trials may be kept in a variety of formats such as SAS® datasets, relational databases (Clintrial v4, DLB systems, Oracle Clinical), flat files etc. The data mapper and loader portions of the XPDL macro use EXCEL® to create meta-data maps that are utilized to create and maintain the project database. These meta-data maps are initially created by XPDL and then easily modified by the user with EXCEL®. Thus, the meta-data maps provide the documentation of the project database and drive the ‘Loading’ process of the XPDL macro.

Depending on the size and sequencing of the developing project data base, multiple meta-data maps may be created. Multiple data maps can be potentially dangerous and needs to be controlled and synconized to assure the quality of the developing data base. Creating a program to combine several meta-data tables into one could be time-consuming, extremely lengthy and difficult to validate and maintain.

CompareXLS is a dynamic SAS® macro program that makes this task easy. The program can read and convert any number of meta-data EXCEL® spreadsheets into SAS® data sets. Manipulating SAS datasets is much easy than with the spreadsheets in terms of resolving difference or updates. The difference between two or more SAS datasets is obtained through the usage of different options of PROC COMPARE, which is incorporated inside the CompareXLS macro. This method proved to be more robust, safer and more efficient. After all the differences are resolved in the final SAS dataset, CompareXLS can convert this dataset back to meta-data EXCEL® spreadsheet. The final data map now provides the documentation of the updated PDB and drives the XPDL ‘LOADING’ process. PDB update times are reduced to a few days and updates can be done by non-programmers.

CompareXLS can also be used to re-map existing data sources into new layouts or table structures. The process of re-mapping or translating existing or updated data sources into a new table is similar to the XPDL ‘loading’ process when creating the PDB. For instance, you can re-map your existing data into the Clinical Data Interchange Standard Consortium (CDISC) compliant domain data structure for a FDA submission or you may use it as a dynamic translator to adapt your data as an input for any data processing.

In addition, CompareXLS may compare and resolve difference between any two or more EXCEL® spreadsheets without any regard to clinical data meta-files. Thus, it may have quite a wide usage through Pharmaceutical and other industries.

1.0 PRELIMINARIES

Here are the main steps of this extensible project database loading and table transformation process to which the CompareXLS macro was added as a necessary supplement:

1. identification of various data sources (all trails) and their locations, what project domains (SAS® datasets) will be created and which data source (dataset/view for
each trial) will be combined into which domain(s).

2. decision how to combine these data sources within each domain (e.g. should they be merged or append).

3. selection of variables that will be kept in each domain and their attributes.

4. design and mapping of trial treatments and regimen codes to common base.

5. possible re-mapping of variable values across the trial if needed.

6. determination if any new derived variables are required.

7. when planned properly, all required information comes together in the two spreadsheets called the ‘all-variable’ and ‘domain-to-view’ maps. These maps are an imprint of original data sources.

8. most importantly, the two spreadsheets from item 7 provide the main input parameters for the NEWDOMAIN and MODDOMAIN functions of the XPDL macro. These functions create new or modified domain meta-files (spreadsheets), each of which defines the entire structure of a specific PDB domain (SAS® dataset). These EXCEL spreadsheets are called ‘domain variable’ maps and present the set of domain variables and their attributes. Figure 2 depicts the typical structure of ‘domain variable’ map (meta-data spreadsheet of domain).

9. all three types of spreadsheets (‘all-variable’, ‘domain-to-view’ and ‘domain variable’ maps) are what the user works with. However, only the latter two are modified by user, if needed.

Figure 1 schematic displays an overview of the typical NEWPDB environment, where creation of a new domain (NEWDOMAIN) is a major point. Please note there is no difference in structure of the ‘domain variable map’, created either by NEWDOMAIN (brand new domain) or by the MODDOMAIN (modified domain) XPDL functions. The ‘domain variable map’ serves as a major source for actual loading of the domain into the PDB. Thus it is the most important meta-data file in PDB, defining the ‘latest’ and ‘greatest’ domain status. However, during the PDB life time cycle and maintenance periods may happen that a user may keep several versions of the same domain maps. The need for combining several of these tables for a domain (or versions of the ‘domain-variable-map’) into one spreadsheet is the purpose of the CompareXLS macro.

2.0 FUNCTIONALITY

CompareXLS macro has a number of major functions, all dealing with reading from domain variable maps, writing to them or simply comparing them. To that extend it is imperative to illustrate the structure of this map in details. The ‘domain variable map’ is created by the NEWDOMAIN XPDL function which contains all the variables and their
attributes that will be sourcing the domain. A description of the contents is depicted in figure 2. There are two types of columns defined in this map: fixed – those that are always present, and study - those that are trial specific. Fixed columns define the name of the domain (adm, for instance, for Domain column), the names of the project domain variables (ptno, actevent, visdt, sex, etc.. for Prjvar columns) and their attributes - Prjvtyp, Prjvfrm, Prjvlen, Prjvlbl columns for project domain variable type, format, length and label, respectively.

**Figure 1** Typical NEWPDB environment.

The ‘domain-variable-map’ is then edited by the user to reflect the planning for the domain. The Prjvcomp column reflects the results of the verification that the NEWDOMAIN function performed for that variable. Specific problems are flagged and must be resolved by the user editing the appropriate project domain variable attributes. This is where the differences tend to occur and multiple spreadsheets are
created. The user may decide to only keep certain variables from the some of the
view/data sets to specific trials by deleting others (rows).

**Figure 2** The domain variable source list (‘domain variable’ spreadsheet).

<table>
<thead>
<tr>
<th>COLUMN NAME</th>
<th>COLUMN DESCRIPTION</th>
<th>NOTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain</td>
<td>Name of the project domain</td>
<td>fixed column names</td>
</tr>
<tr>
<td>Prjvar</td>
<td>Name of the project domain variable</td>
<td></td>
</tr>
<tr>
<td>Prjkey</td>
<td>=index if variable has needs to be indexed</td>
<td></td>
</tr>
<tr>
<td>Prjvcomp</td>
<td>Comparison/error edit flag</td>
<td></td>
</tr>
<tr>
<td>Prjvbl</td>
<td>Project domain variable label</td>
<td></td>
</tr>
<tr>
<td>Prjvtyp</td>
<td>Project domain variable type</td>
<td></td>
</tr>
<tr>
<td>Prjvfrm</td>
<td>Project domain variable format</td>
<td></td>
</tr>
<tr>
<td>Prjvlen</td>
<td>Project domain variable length</td>
<td></td>
</tr>
<tr>
<td>Sdyview</td>
<td>Study view/dataset name</td>
<td></td>
</tr>
<tr>
<td>Sdyvar</td>
<td>Study view/dataset variable name</td>
<td></td>
</tr>
<tr>
<td>Sdylbl</td>
<td>Study view/dataset variable label</td>
<td></td>
</tr>
<tr>
<td>STUDY1 - STUDYN</td>
<td>Study1 – studyN folder names</td>
<td>One set Per study</td>
</tr>
<tr>
<td>etc ..</td>
<td>Formats, types and lengths of STUDIES</td>
<td></td>
</tr>
</tbody>
</table>

Additionally, the user can rename domain variables ($Prjvar$), re-type domain variables ($Prjvtyp$), change the length of the project variable ($Prjvlen$), assign project formats to domain variables ($Prjvfrm$), or reference user macros (in $Prjvar$ and/or $Sdyvar$ columns) that derive new domain variables (there are four different types of derivation macros available). The $Prjkey$ column allows the users to specify how to combine the data from various view/data sets, e.g. merge them or append (default setting). Specifying INDEX in this column will cause an index to be created for that project variable.

As a result of these various editing options the user may end up having a multiple copies of the same ‘domain variable map’, which is difficult to maintain and to document changes. Adding new trial data and keeping two different instances of the project data base i.e.: “QC” and “Test” can make this task more tedious.
In order to synchronize these two environments, resolve the differences between them and to create a ‘single verified domain variable map’ the CompareXLS macro is used. Table 1 depicts the functions of this macro.

Table 1  Functions of the CompareXLS macro

<table>
<thead>
<tr>
<th>Function</th>
<th>Function Purpose</th>
<th>Function description</th>
</tr>
</thead>
</table>
| readxls  | creates SAS® dataset based on domain variable spreadsheet | • reads domain variable map  
• converts it into SAS® dataset |
| compare  | compares different SAS® datasets | • Proc Compare incorporated  
• Compares two SAS® datasets by their variables and attributes |
| writexls | creates domain variable spreadsheet | • Creates single verified domain variable map from updated (resolved) SAS® dataset  
• Outputs this map to the desired location |

There are two functions that are used for transition between the EXCEL® spreadsheets (‘domain variable maps’) and the SAS® datasets. The middle function makes the actual comparison of the datasets.

The readxls function reads the domain variable spreadsheet and converts it into a SAS® dataset. The domain meta-data file is converted into a conventional SAS® dataset. By supplying macro parameters with two different ‘domain-variable-map’ names to the CompareXLS macro two different SAS® datasets are created. These SAS® datasets can be easily compared by the SAS® PROC COMPARE procedure incorporated into CompareXLS macro code (compare function). The Proc Compare evaluates the two datasets by their variables (project domain variables) and their attributes. After all difference between the SAS® datasets have been reviewed by the data manager and resolved by means of conventional SAS® BASE data manipulation programming, the final verified meta-data file of the project domain is complete. This final dataset is then converted back to an EXCEL® spreadsheet by using the writexls function, creating the ‘single verified domain variable map’. The writexls function is capable of taking the SAS dataset and output it as a map (spreadsheet) to any desired location. Thus this
verified spreadsheet can be used for a final PDB data load and eventually as a template for the regulatory submissions to FDA. Figure 3 depicts the overview of the typical CompareXLS environment:

**Figure 3   Typical CompareXLS environment**

```
2.0 General Syntax

The general syntax for CompareXLS is quite simple. There are 10 common arguments in Compare XLS call that are needed for most functions, as shown below. The exact syntax for each function, however, varies because of the different requirements for each function (mainly the optional arguments in macro call).

%CompareXLS (function = , - selects 3 types of functions (see above), required for all macro calls

    flnm1 =, - first (base) spreadsheet to compare
```
3.0 Creation of Single Verified Domain Map

To implement a single verified domain map, gather the information discussed in 1.0 PRELIMINARIES section. After a plan is decided, proceed to the creation of single verified domain variable map.

For instance, very often there may be two instances of the PDB, one for QC (quality assurance) and the other for TEST (development environment). Imagine now that so far the user was successful in putting the parallel (to the same trial) changes to both environments taking into account user’s consideration that QC process of drug trials, contributing to PDB, will be completed soon. However, as the time passes maintaining of this process becomes more and more complicated since QC process may be far from finishing and the request queue for the additional changes to these drug trials continues to grow. As a result of this redundancy of existing parallel domains (QC and TEST) the user has to apply updates to two sets of domain variable maps at the same time. Application of changes to both environments using different domain variable maps is too time consuming, complicated and unsafe. There is definitely a need for creation of single verified domain variable map. Please note that a similar situation may arise when the user has just two or more different versions of the same domain map and wants to merge the differences into one verified map. It is extremely important at the time of submission, when the user wants to have all changes applied to different versions of the same domain map to be merged into one single map.

Figure 4 presents this process as an example of creation of ADM domain’s single verified domain map. The user has two EXCEL® spreadsheets, proj_db_list_adm_v6.xls, and proj_db_list_adm_v7.xls, which are project domain variable maps for ADM domains.
existing in the QC and TEST environments, respectively. The goal for the user is to find the differences between these two maps, resolve them and create a single verified map that will be used for final load into PDB.

The first step in that process is to convert these two EXCEL® spreadsheets into two SAS® meta-data sets, proj_db_adm_qc and proj_db_adm_test, respectively. That will be accomplished by readxls function of CompareXLS macro.

The second step is their actual comparison by their project level variables, their attributes, number of observations, contributing trials, study views, formats, types, labels and other means. That can be easily done by compare function, which is actually a PROC COMPARE, incorporated into CompareXLS code. The differences should be identified and resolved according to domain specifications and by means of custom SAS® BASE programming that the user will apply to these SAS® datasets.

After all differences will be resolved or merged, the final SAS® dataset will be created. Basically it is a meta-data SAS® dataset, a prototype of single verified ADM domain variable map. The last step in the process will be a conversion of this SAS® dataset into EXCEL® spreadsheet, which will serve as a final verified domain map. That will be easily accomplish by writexls CompareXLS function.

The single verified domain variable map can be used now for immediate load into PDB, optionally into two environments, QC and TEST, with the consequent usage for the further user’s editing or directly for the regulatory submission. That process is out of scope of CompareXLS macro, and will be completed by XPDL macro as outlined it in ‘PRELIMINARIES’ section of the article.

CompareXLS can also be used to re-map existing data sources into new layouts or table structures like, for instance, Clinical Data Interchange Standard Consortium (CDISC) complaint domain data structure, or others.

CONCLUSION

CompareXLS is a powerful comparison tool for spreadsheet difference resolution. The spreadsheets, maintained by users with CompareXLS skills, provide the documentation and drive the PDB loading process. CompareXLS can be applied beyond the scope of domain variable maps; it can compare any two or more EXCEL® spreadsheets, making this macro more widely applicable in Pharmaceutical and other industries. CompareXLS can be useful in re-mapping to the Clinical Data Interchange Standard Consortium (CDISC) for submission to the FDA.

CompareXLS has been a tremendous success. Many resolutions of project database spreadsheet differences have been accomplished in a relatively short time by users with little or no database experience. The higher quality of database documentation, lower maintainability costs and the financial savings for programming resources, make this
macro an additional necessary component of XPDL process of developing a drug project databases.

Figure 4  Creation of single verified domain variable map (example)
ACKNOWLEDGMENTS
Our sincere appreciation for the support and encouragement during the development of this application goes to Vijay Nandi and Amy Capilupi at Boehringer Ingelheim, Inc.

CONTACT INFORMATION
Your comments and questions are valued and encouraged.
Contact the authors at:

Kirill Tchernakov
Boehringer Ingelheim Pharmaceutical, INC.
900 Ridgebury Road
Ridgefield, CT 06877-0368
Work phone: 203.798.4239
Fax: 203.798.4242
E-mail: ktcherna@rdg.boehringer-ingelheim.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.

References.