

Introduction

The cleaning of collected data is a huge and essential task in clinical trials which starts with the use of EDC technology *at the earliest stage of the project*. In our company which provides in-house EDC applications, SAS is used for most data management activities, which leads to data managers being faced with new challenges. The validation plan which gives an exhaustive description of all the checks to be applied to the data must be available at the beginning of the eCRF development stage, while SAS programs for edit checks must be implemented within very short timelines to enable queries to be sent as soon as the data collection has started. It is not rare then that the validation plan is changed during the course of the study and subsequent programming changes must be managed as well. In this context we needed a solution to simultaneously allow the definition of the validation plan, fast, accurate and standardized SAS programming and the easy update of the validation plan during the study.

Solution overview...

- Fast, accurate and standardized programming objectives can be met through **automatic SAS code creation**. Regarding issues concerning edit checks, two SAS code templates are defined

- Programming must match specifications and changes of the validation plan: a simple solution is to store the complete validation plan information in a database (**Oracle Table**) and to use the updated information stored to execute SAS edit checks at a later time.

- As automation requires parameters to be entered the need for a user-friendly interface is obvious. It can be easily managed with a single **Web-form**.

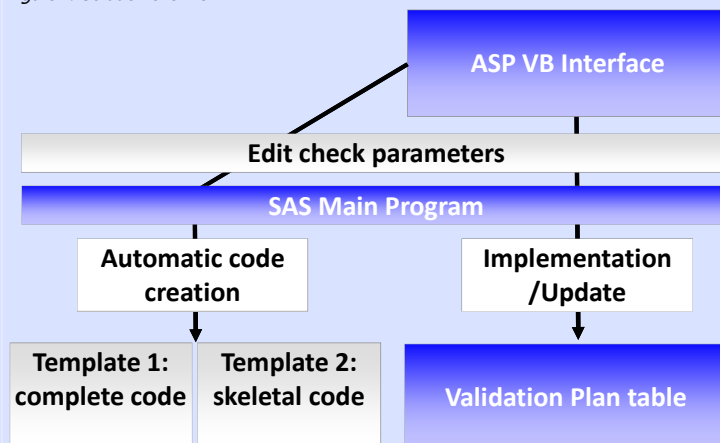
Creating edit checks

For each new check, the description and programming parameters is entered within the web-form. The SAS program that creates a new record in the validation plan table for each check and generates automatically corresponding SAS templates, is executed. **At the end of the specification stage, the SAS programming tasks are more or less complete or well advanced and match the exhaustive validation plan.**

Updating the validation plan

The user modifies the check attributes so that the validation plan table information is updated accordingly while SAS edit check programs remain unchanged.

Figure 1: Solution overview



Deactivating a check

An obsolete status is associated to the identified check in the Oracle table, indicating that the check won't be executed anymore. Again no changes need to be made to existing programs.

...And the SAS side

All functionalities are implemented in a single and rather basic SAS program, as shown in Figure 1. The program loads parameters entered through the web-form and manages the validation plan and automatic code creation with appropriate and classical *if/then* statements, SQL procedures and *put* instructions.

Template 1 – Complete code for basic checks

The template consists of the standard company header and a complete program structure including program comments. Basic checks like missing data or conditional missing data checks can be targeted with a single macro offering huge flexibility. The next macro, aims at loading the appropriate message from the validation plan table and offers the possibility to manage dynamic items:

VISIT	VALIDATION PLAN MESSAGE	FINAL MESSAGE
INCL	At visit <VISIT> Weight is missing. Please complete.	At visit INCLUSION Weight is missing. Please complete.

This results in both the standardisation of programming and advantageous time-saving.

Template 2 – Skeletal code for complex checks

For more complex checks, like consistency checks, the difference with the previous template remains only in the first part of the program, where the macro managing missing items is replaced by a comment indicating that the program needs to be completed by a SAS programmer. Again this approach ensures standardized and efficient programming.

Figure 2: Automatique code (template1):

```

*****
* CLIN DATA MANAGEMENT
* PROGRAM NAME      : <$_MACRO>
* STUDY             : <$_PROJECT>
* PROJECT           : QUERIES MANAGEMENT
* -----
* AUTHOR            : $sysuserid
* CREATE DATE       : $idate
* VERSION           : 1.0
* -----
* DESCRIPTION       : Consistency check <$_MACRO>
* { .....
*****
%MACRO $_MACRO(lib=$_GLIB,delwork=1);

/*1. Find inconsistency*/
%MISSING_ITEM(input=$_lib..<$_PANEL>,
              item=$_ITEM,
              output=%str(TMP(where={<$_WHERE>}));

/*2. Load Message from Validation plan table*/
%FORMAT_TEXT(input=TMP,check=<$_MACRO>);

...
%MEND;
  
```

For better understanding, macro variable names initialised in the main program have been indicated with the following syntax <\$_name>; of course in programs generated, the macro variables are replaced by their values so that the code is ready to be used.

Conclusion

The combination of the ASP VB interface and the SAS program for automatic code generation offers an excellent solution for fast and standardized programming of edit checks, even for non SAS specialists. The workflow and SAS programs implemented are nevertheless quite straightforward and the solution relies mainly on the automatic code template design and associated SAS macros. Similar solutions could be easily developed for other data management processes, particularly in a context where pressure for reducing timelines is constant.