PAPER DM06

IDENTIFYING AND SURFACING DATA CHANGES FOR EFFECTIVE QUALITY MANAGEMENT

W. Stetson Line, Genentech, Inc., South San Francisco, CA
Scott A. Bahlavooni, Genentech, Inc., South San Francisco, CA

ABSTRACT

Manual data review is an iterative, resource-intensive, and often inefficient process. This paper describes a new approach to highlighting new and modified data in clinical review listings, provides a technical implementation walkthrough, and presents a case study that illustrates how this approach has reduced costs and expedited one project's data review tasks. The presentation is based on SAS version 9.1 and is system agnostic. The technical material is suited to intermediate to advanced SAS programmers who are familiar with PROC REPORT / ODS PDF.

KEYWORDS: SAS, PROC REPORT, ODS, PROC COMPARE, MACRO

INTRODUCTION

Good data management plans seek to enforce quality via a three pronged attack: Automated edits serve a pivotal role, as does well-defined and executed clinical monitoring: The third essential component is manual data review.

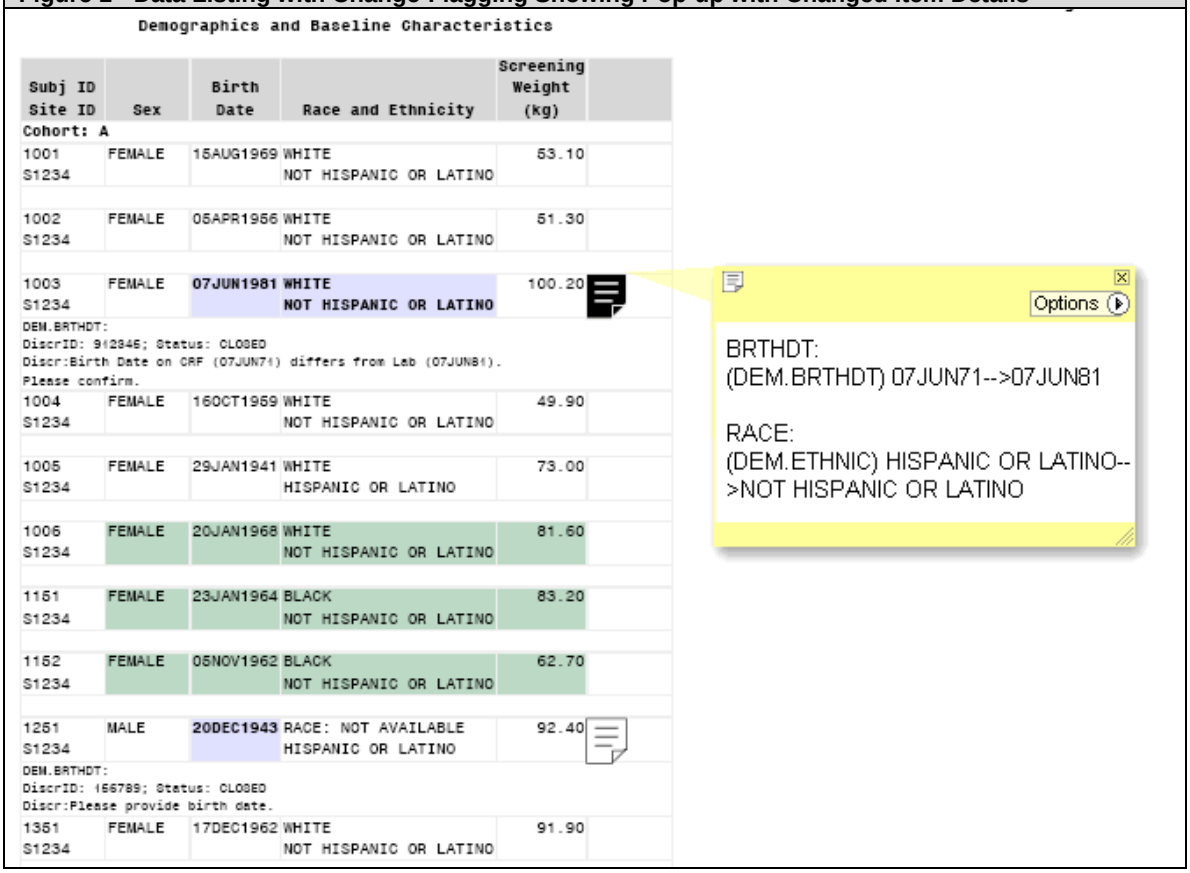
One of the challenges in manually reviewing trial data is that it involves expensive, highly-skilled personnel in what is often perceived as a low-value activity. This is certainly the case when the scope of review is too broad or when the data listings are poorly designed. Listing design innovations have sought to address these drawbacks by increasing data dimensionality. For example, in contrast to one-dimensional 'dumps' of adverse event or medications data, "smart" listings combine and order data from multiple domains to help provide all relevant data in a concise and reviewer-friendly format. Tools that present a graphical summary with the ability to 'drill-down' to the details are another example of adding dimensionality. One data dimension that has been largely neglected - is : time.

Adding a temporal dimension to data listings or more specifically adding information about what has changed in a given time period is of great benefit. Clinical trial data quality is a moving target: items are constantly being entered and amended. Further, some changes impact the assessment of related data. Highlighting new or changed items enables targeting specific rows during incremental reviews, saving time both in initial review and in subsequent identification of items that may require further investigation. The reason this "change flagging"' is not a wide-spread practice is that it presents numerous operational and technical obstacles. This paper will demonstrate how to surmount these obstacles and implement a flexible, lightweight, and powerful solution.

OUTPUT EXAMPLE

An example of a listing surfacing data changes, also known as “change-flagging”, is shown below (see Figure 1). A comparison date provides the baseline for determining what gets flagged or highlighted. New data items new are highlighted in green. Data items that have changed since the comparison date are highlighted in blue. In addition the listing is populated with PDF notes that contain details of what has changed, and these will “pop-up” with mouse actions when viewed electronically. As a concession to hard-copy users changed items are also bolded to help distinguish them from new items in black-and-white output.

Figure 1 - Data Listing with Change-Flagging Showing Pop-up with Changed Item Details



Demographics and Baseline Characteristics

Subj ID Site ID	Sex	Birth Date	Race and Ethnicity	Screening Weight (kg)
Cohort: A				
1001 S1234	FEMALE	15AUG1969	WHITE NOT HISPANIC OR LATINO	53.10
1002 S1234	FEMALE	05APR1956	WHITE NOT HISPANIC OR LATINO	51.30
1003 S1234	FEMALE	**07JUN1981**	**WHITE** **NOT HISPANIC OR LATINO**	100.20
DEM.BRTHDT: DiscrID: 912845; Status: CLOSED Discr:Birth Date on CRF (07JUN71) differs from Lab (07JUN81). Please confirm.				
1004 S1234	FEMALE	16OCT1959	WHITE NOT HISPANIC OR LATINO	49.90
1005 S1234	FEMALE	29JAN1941	WHITE HISPANIC OR LATINO	73.00
1006 S1234	FEMALE	20JAN1968	WHITE NOT HISPANIC OR LATINO	81.60
1151 S1234	FEMALE	23JAN1964	BLACK NOT HISPANIC OR LATINO	83.20
1152 S1234	FEMALE	05NOV1962	BLACK NOT HISPANIC OR LATINO	62.70
1251 S1234	MALE	**20DEC1943**	RACE: NOT AVAILABLE HISPANIC OR LATINO	92.40
DEM.BRTHDT: DiscrID: 156789; Status: CLOSED Discr:Please provide birth date.				
1351 S1234	FEMALE	17DEC1962	WHITE NOT HISPANIC OR LATINO	91.90

PROCESS OVERVIEW

As users of PROC COMPARE are aware it is a relatively simple matter to provide a general listing of everything that has changed in data from one time point to the next, assuming of course, that one has archived the older data. The difficulty lies in integrating this information in the context of a clinical review listing in a format that is visually appealing and elegant. To illustrate how this is done an example of retrofitting an existing PROC REPORT program with change-flagging will be presented.

The following steps for implementing change-flagging will be discussed in detail:

i. Create the change metadata look-up table
ii. Add indicators/link ids to all eligible SAS datasets
iii. Modify code in report programs to keep record link ids
iv. Add macro calls to create highlight support variables
v. Modify proc report step to add change flagging

STEP I: CREATE THE CHANGE METADATA LOOK-UP TABLE

Change-flagging metadata is created by mining Clinical Data Management System (CDMS) audit trails or by comparing SAS data library archives, depending on what data is available for a given study. Because the comparison date options are limited in a SAS data library to whatever dates the data have been archived a CDMS audit trial is the preferred data source. CDMS databases can also provide other data of value to clinical reviewers such as query text and status metadata.

To create the metadata a SAS program evaluates data item status using the available data source and comparison date. A dataset containing records for each data item change item is stored in the target SAS data library. Each record contains a record id, data set and item names, and the date and nature of the data changes (see Figure 2). The advantages to storing this data in one file are that its normalized structure is efficient and that it provides a convenient source for generating metrics of all data changed during the target time-frame. This metadata is created either on-demand or as part of batch process.

Figure 2 – Sample Metadata Created to Support Change-Flagging

	Record ID	Dataset Name	Flagged Variable Name	Previous Value-->Current Value	Date Value Changed
1	50607~S3219999999~NON-SERIOUS~1	AECO	AEREC	NO-->YES	03JAN2008
2	50556~S3109999999~NON-SERIOUS~2	AEDEAEDE	AERDT	.-->19APR07	03JAN2008
3	50603~S3069999999~NON-SERIOUS~2	AEDEAEDE	AERDT	.-->30MAR07	18DEC2007
4	50603~S3119999999~NON-SERIOUS~2	AEDEAEDE	AERDT	.-->29JUN07	18DEC2007
5	50760~S3229999999~NON-SERIOUS~1	AEDEAEDE	AERDT	.-->30DEC07	14JAN2008
6	50853~S3049999999~NON-SERIOUS~1	AEDEAEDE	AERDT	.-->14DEC06	04JAN2008
7	50859~S3069999999~NON-SERIOUS~2	AEDEAEDE	AE	TROCHANTERIC BURSITIS-->TROCHANTERIC BURSITIS LEFT HIP	04JAN2008

STEP II: ADD INDICATORS/ LINK IDS TO ALL ELIGIBLE SAS DATASETS

Using PROC CONTENTS output and macros the next step programmatically determines the SAS library tables eligible for change-flagging. To qualify, a dataset must contain certain variables that uniquely identify each record. A change-flagging indicator variable is added to the selected tables. The indicator is set to 'NEW' for observations new since the comparison date or as a unique tilde-delimited id string for rows *with at least one* changed data item. The tilde-delimited id and field name links records in the respective source SAS table to the changed value metadata in the look-up table created in Step I. Records that are 'NEW' do not use the look-up table as all their fields have no prior history or changes to surface.

NOTE: To avoid potential SAS field name contention the unique id variable added to each eligible dataset is named after its source dataset with a prefix '_', e.g. the id field for dataset 'DEM' is '_DEM'.

STEP III: MODIFY CODE IN REPORT PROGRAMS TO KEEP RECORD LINK IDS

This is a relatively simple but important step. The change-flagging indicators created in Step II need to be preserved throughout the data steps used in the reporting programs. This may be as straightforward as adding these fields to KEEP statements. The indicators must be preserved as they indicate as observation is new or serve as the link to the look-up table created in Step I.

STEP IV: ADD MACRO CALLS TO CREATE HIGHLIGHT SUPPORT VARIABLES

Change-flagging will not be appropriate for all fields, e.g. key or sort variables. To mark the desired fields for change-flagging a SAS macro, CSPV_DELTAFLG, matches the source SAS table record link ids to the look-up table and returns change metadata via two new flag support variables. The first, DELTAFLG, contains a tilde-delimited string of reporting variables for which the report-level data item status evaluates to "new" or "changed". This is used to support data-item highlighting. The second, DELTAVAL_xxx (where xxx is the dataset name) supports implementing PDF NOTES containing the changed data values. If these additional fields already exist in the dataset the macro appends the change values to their contents using a 'variable name : values' format separated by tilde delimiters.

This process is the same for flagging derived fields where any number of source fields contributing to the derivation may have changed. An example of this is shown in the discussion of the CPSV_DELTAFLG macro (see Figure 3).

Figure 3 – %CPSV_DELTAFLG Syntax, Description, and Code Sample

%**cpsv_deltaflg**(ListingColumnName,
SourceTable1: SourceColumn1 SourceColumn2...SourceColumnN
SourceTable2: SourceColumn1 SourceColumn2...SourceColumnN...
SourceTableN: SourceColumn1 SourceColumn2...SourceColumnN) ;

Parameter	Description
ListingColumnName	A single reporting column for which data item status will be evaluated. This may be the source dataset variable or a derived, composite variable.
SourceTables: SourceColumns	A formatted string of source SAS table(s) and column(s) comprising the reporting variable. A colon delimits the source dataset from a space-delimited list of source variables. A space delimits each source dataset/variable group.

Example

```
data dem_icf_vitals ;
  merge dem (in=in_d) icf(in=in_elig) vitals (in=in_vs)  ;
  by patnum ;

  attrib
  age       label='Age at Informed Consent' ;

  if in_d ;

  if brthdt ne . and icfdt ne . then
   age = floor(yrdif(brthdt, icfdt, 'ACT/ACT'));


  if (_dem ne '' or _icf ne '' ) then do;
    %cpsv_deltaflg(age,dem: brthdt icf:icfdt);
  end;

  if (_dem ne '' or _vsdevsde ne '' ) then do;
    %cpsv_deltaflg(brthdt,dem: brthdt);
    %cpsv_deltaflg(sex,dem: sex);
    %cpsv_deltaflg(race,dem: indalk asian black island white racena ethnic);
    %cpsv_deltaflg(wt,vsdevsde: wt) ;
  end;

  keep cohort patnum subjsite sex age brthdt race wt _dem deltaval_dem deltaflg;
run ;
```

In the highlighted code in the Figure 3 example, the data item status of reporting variable AGE is evaluated by assessing the status of both source variables, DEM.BRHTDT and ICF.ICFDT. If either the ICF.ICFDT *or* the DEM.BRTHDT changed since the comparison date the AGE column would be marked for change flagging by appending its change metadata to the flag support variables: DELTAFLG and DELTAVAL_DEM.

STEP V: MODIFY PROC REPORT STEP TO ADD CHANGE FLAGGING

Change-flagging uses well-documented ODS highlight or stop-lighting techniques, the lesser-known FLYOVER style element, and COMPUTE blocks to produce the enhanced output. The minimal impact of change flagging code used to retrofit a typical report step can be seen in Figure 4: (change flagging code highlighted).

Figure 4 – Impact of Change Flagging-Specific code on Proc Report Example

```
proc report data= dem_icf_vitals headline headskip missing spacing=2 nowindows list;
  columns cohort patnum subjsite deltaflg sex brthdt age race wt deltaval_dem _dem ;

  define cohort   /order order=internal noprint ;
  define patnum   /order                noprint ;
  define subjsite /left width=7        "Subj ID" "Site ID";

  define sex      /left  width=7       "Sex" ;
  define brthdt   /left  width=15      "Birth Date" ;
  define age      /right width=15      "Age" ;
  define race     /left  width=45      "Race and Ethnicity" ;
  define wt       /right width=10      "Screening Weight (kg)" ;

  compute before cohort /style=[background=white font_weight=bold] ;
    line @1 'Cohort: ' cohort $10. ;
  endcomp ;
  break after cohort/page ;

  %cpml_delta(dem);
run ;
```

A SAS macro, CPML_DELTAFLG, generates variable specific COMPUTE block statements within the PROC REPORT step for all variables passed through the CPSV_DELTAFLG macro, i.e. those intended for change-flags that actually had changed data. These COMPUTE statements are used to apply highlights and to create the changed data value PDF note (see Figure 5.) Each COMPUTE block evaluates at a record level whether the data item requires special formatting based on a search of change flagging support fields: DELTAFLG, and DELTAVAL_xxx.

Figure 5 – Example of Compute Statement Blocks for Highlighting Flagged Data Points

```
/* Apply data-change stop-lighting to AGE */
COMPUTE  AGE;
if find(deltaflg, "~AGE:C", 1) then do;
  call define(_col_, "style", "style=[font_weight = bold background = CXEOEOFF]");
end;
else if find(deltaflg, "~AGE:N", 1) then do;
  call define(_col_, "style", "style=[background = CXBCD9C5]");
end;
ENDCOMP;
/* Create changed data value PDF Note */
COMPUTE  _DEM / CHAR LENGTH=4  ;
if DELTAVAL_DEM ne '' then do;
  call define(_col_, "style", 'style=[flyover = "' ||
              translate(strip(DELTAVAL_DEM), "'", '"') || '"]');
end;
ENDCOMP;
```

Obviously there are a lot of technical details that couldn't be conveyed in the space of these few pages. However this overview should have provided a decent understanding of the technical framework and scope of work involved in this approach. To better understand the benefits of applying change-flagging a brief case study will be considered.

CHANGE-FLAGGING CASE STUDY

Trial Overview:

560 Patient Phase III Trial
Outsourced Data Management
Internal Medical Monitoring
Shared Data Review of 20-30 Listings

Retrofitting and Roll-Out

Six to nine months prior to an interim analysis, the data review team, a Genentech / CRO collaboration, initiated a monthly, sub-grouped data review plan. The team planned to review specific patient blocks; subsequently re-review certain patients; and near database lock, cumulatively review all patients. The data review team agreed to evaluate changed-flagged reports for their usefulness. Retrofitting the two dozen reports to support change-flagging required approximately two weeks. At first these new features met with some skepticism but the team agreed the enhanced reports had promise and requested they be placed in production.

Ongoing Reviews and Raves

Skepticism quickly turned to acceptance and praise after a few review cycles. The new format "made a torturous task manageable." New listing specifications now included change-flagging as a requirement. The team associated comparison dates with each patient subset; thereby enabling targeted review of different patient sub-groups in parallel with minimal cross-interference or confusion. These new techniques reduced outsourcing costs for subsequent re-reviews: The CRO billed based upon the number of patients in a review cycle. Genentech requested and the CRO agreed to bill based upon the number of patients *with new and/or changed data* in a review cycle. Another benefit was that the change-flagging concept (not process) was extended to patient tracking activities making these more efficient.

A Locking Benchmark

Prior to database lock, the team discovered an anomaly directly impacting crucial interim analysis data points. The rapid review-query-resolve cycle included two key change-flagged reports. The first report highlighted data relative to a baseline comparison date and the second highlights daily additions and updates. These report enabled the data review team to cumulatively monitor expected changes and simultaneously track daily progress. According to one of the principle reviewers, "the highlighted listings…produced for the (data) issue right before the database locks were a godsend to (the reviewers)! We could not have kept our heads straight nor completed the task without the listings".

Case Study Conclusion

Change-flagged reports provided significant benefit to the study data review team. Although, quantifying time savings depends on the reviewer and review type, the Clinical Trial Manager estimated a savings of 50% to 75% for certain, targeted internal reviews. Time savings could not always be quantified especially for cross-report reviews (i.e. reviewing two or more listings); however, the Clinical Trial Manager indicated change-flagging promoted more direct cross reviews. Change-flagging reduced outsourcing costs enabling Genentech to bill only for those patients with new or changed data. Change-flagging conserved data review team resources and promoted more focused, cost-effective data review.

NEXT STEPS

Surfacing data changes advances data review; however, further efficiencies stand to be realized by supplying discrepancy information. Surfacing discrepancies enables reviewers to determine if their concerns have been addressed and if so, addressed adequately. The process presented above easily extends to surface discrepancies. (see Figure 1 for discrepancy data prototype output) Another macro currently being evaluated creates hyperlinks to discrepancy information served up by a SAS/IntrNet program.

Ideally, data review could be fully tracked and audit-trailed. This could provide further efficiencies and valuable metrics. However this would require a fundamental shift from paper-based to online review: the more advanced features of change-flagging provide a gentle nudge to users toward adopting this modality. Key change-flagging concepts, e.g. mining CDMS audit trails, creating linked data and metadata tables, and automating status assessments, could be applied to data review audit-trailing.

As SAS continues to provide reporting options and destinations, reports and processes evolve. The current PDF destination offers limited bookmarking and internal hyperlinking. RTF and Excel tagsets may provide a more efficient and effective alternative to surface change.

CONCLUSION

The approach to surfacing changes outlined in this paper is the culmination of several years of innovation and trial and error. Has there been sufficient return on this investment? It is impossible to exactly quantify this, but if the experiences from our case study can be extrapolated to the nearly three-dozen projects that are currently employing 'change-flagging' then the savings are substantial, potentially enough to fund additional investigational trials. Targeting only un-reviewed data or data requiring additional review conserves the energy and eye-sight of clinicians and data managers and frees up time for them to focus on areas most likely to improve data quality. The likely result is that data review is not only performed faster but with an increased efficacy in identifying and correcting quality issues.

Contact Information:

W. Stetson Line
Sr. Manager, Clinical Programming/CDM
Genentech, Inc.
South San Francisco, CA
stetson@gene.com

Scott A. Bahlavooni
Sr. Clinical Programmer Analyst, Clinical Programming/CDM
Genentech, Inc.
South San Francisco, CA
scottab@gene.com