

Computerized Monitoring Aids to Data Validation

David S. Focht

Pharmaceutical Research Associates
Charlottesville, Virginia

ABSTRACT

Electronic data bases have become an integral portion of clinical trials and hence the drug development process. The medical review (monitoring) and validation of the data is time-consuming and expensive. Final analysis cannot be performed and no conclusions can be drawn before the medical review and validation of the data. Any errors or problems with the data need to be identified, examined, and resolved. In an effort to make the process as efficient and productive as possible, the Data Base Programming (DP) Department at Pharmaceutical Research Associates (PRA) spends a considerable amount of time developing Computerized Monitoring Aids (CMAs) for each of our data management projects.

Written completely in SAS®, the programs that compose the CMAs consist of two primary parts; a generic driver program and a series of individual programs tailored to the specific data being reviewed. The complexity of the error checks performed range from the simple identification of missing data to the validation of complex visit progressions and date sequencing. Standard programmatic checks are performed by using a tool box of generic macros.

The use of CMAs increases the efficiency of data entry and monitoring, resulting in the production of a final data base (DB) in less time and with greater accuracy.

INTRODUCTION

Clinical trials are a major facet of the drug development process. Data from a clinical trial are recorded on Case Report Forms (CRFs). The CRFs record the information associated with a given clinical trial. This includes routine information, such as demography, medical history, concomitant medications, adverse events experienced during the study, as well as study-specific information pertaining to the specific aspects of the compound being studied. The information contained on the CRF is eventually entered into an electronic data base (DB) to facilitate analysis and reporting.

In order to avoid the consequences of the adage "Garbage In, Garbage Out," all of the data in the DB must be reviewed from a medical point of view. This process is known as clinical monitoring. Monitoring typically takes place during the study at the actual site of the trial, as well as after the data have been entered into the DB. The latter process is often referred to as "in-house monitoring" and is undertaken to ensure the accuracy of the data and validate the information entered into the DB.

The quantity of information collected in a clinical trial can be staggering. Trials can include thousands of patients, each with a CRF consisting of hundreds of pages. Reviewing the data manually is not a cost effective option. With this in mind, the Data Base Programming (DP) programmers at Pharmaceutical Research Associates (PRA) develop a collection of programs for each clinical trial to assist with the medical review and validation of the DB. These

programs are referred to as the Computerized Monitoring Aids (CMAs). The output from these programs guides and assists the clinical monitors as they wade through the data in an effort to identify and resolve problems from the collection, recording and processing of the data.

DATA CLARIFICATION POLICIES

The first step in the development of the CMAs is a meeting between the clinical monitors at PRA and the client or sponsor of the clinical trial. The focus of the meeting is the development of the Data Clarification Policies (DCP). These policies form the basis of how the data will be reviewed and corrected when needed. The DCPs vary widely from client to client and study to study. Clearly delineated in the document are those checks that will be performed by a manual review of the CRFs and those that will be performed programmatically by the CMAs.

The checks performed by the CMAs range from simple to complex. Simple checks are performed to identify missing data, such as the age, height or gender of a patient. Relational checks are performed to confirm data indicating that there were no adverse events or concomitant medications recorded. More complex checks are performed to evaluate the validity of visit progressions, date sequencing or time windows during which certain activities were to have taken place.

Some checks are too complicated to be effectively performed solely by the CMAs and require the judgement and experience of a clinical monitor. In these cases, the monitor can be alerted to the existence of a certain set of circumstances and therefore the need for clinical review of the relevant CRF section(s).

DESIGN/IMPLEMENTATION

Driver Program

Once the clinical monitors and the sponsor have agreed on what data will be reviewed and how that data will be reviewed, a programmer from the DP department meets with the clinical monitor to fine tune exactly what will be checked by the CMAs. The programs are written in SAS 6.08 on a VMSCluster running on DEC VAX machines and consist of two main portions. The first is a generic driver program that includes all the overhead programming (libnames, include files), the error messages, and the DATA _NULL_ that generates the formatted output. The second portion is a series of individual, study-specific programs, custom written to the specific data being examined.

The generic driver contains all the overhead programming. This includes specification of LIBNAME and definition of all include files and macros. Many of the standard generic checks performed in virtually every study have been written as macros capable of having the required study-specific information passed in single-line macro calls. An example of this type of checking is the search and identification of flags inserted into the data fields by the entry technicians. These flags identify data that were deemed illegible during entry and will require careful examination and possible querying back to the site for clarification before being entered into the DB. Also flagged during the entry process are fields for which data were not collected or were unavailable. This is an important distinction from missing data. There is no efficiency in requesting data from a site when it has been clearly indicated as unavailable. The check for the presence of these flags in virtually every field in the DB has been reduced to a single line macro call that results in the

identification of all flag-related errors.

The PROC FORMAT statement with the error messages is also contained in the driver program. These messages provide the clinical monitor with the information needed to identify the problem quickly. The use of specific error messages and grouping them all in one location provides good self-documentation as to exactly what is being checked by the CMAs.

The final portion of the driver program is a generic DATA _NULL_ that provides the formatted output.

The driver program is easy to maintain and prepare for each data management project. It follows a standard format, and modification consists of identifying the study-specific programs as files to be included and the study-specific error messages associated with those programs. The DATA _NULL_ is generic and requires only minor title and header modifications. The standardization of the format and structure of this program makes maintenance and execution by members of the department other than the author more efficient.

Study-Specific Programs

The creation of study-specific programs is the most challenging and time-consuming step in the development of the CMAs. The CMAs for an average clinical trial will require 40 - 80 programming hours. This amount of time is easily recovered in time saved during the monitoring of 150,000 CRF pages. Study-specific programs are generally written on a form-by-form or file-by-file basis. That is to say, the checks for the adverse event form or the medical history form will usually compose a single study-specific program. These study-specific programs can become very complex, requiring data to be pulled in from many or all of the files

associated with the DB. It is these complex cross checks that save the clinical monitors the most time.

Complex programs, such as checking the sequence of the visit dates for a study, involve many data files. This division of tasks keeps the level of complexity in any given program under control. For example, visit dates for a year-long study during which patients were seen every eight weeks (+/- 3 days) can be easily cross-checked and sequenced. Suppose that, at each visit, an ECG, physical exam and lab data form were completed. The manual verification of data on these CRFs would require that the monitor cross-reference, in sequence, twenty to thirty pages as well as perform the necessary interval calculations. This task is perfectly suited to the CMAs.

Save Sets

An important aspect of the CMA design is to eliminate the clinical monitors' review of a potential error more than once while still processing all of the data all of the time. Processing all of the data is required because patients can receive updated information at virtually any time during the project. This updated or additional information can contain errors or be inconsistent with existing pages. It would be unreasonable for the monitor to review all of the patients all of the time, but not for the CMAs. Following each run of the CMAs, data sets containing the details on each anomaly identified are created and saved (save sets). On subsequent runs, the identified problems are compared with the previous run, and new occurrences are indicated. This eliminates the need for the monitor to repeatedly evaluate the same data anomaly.

Program Validation

The primary programmer performs cross-checks during the development of the programs to verify their accuracy. The final test, however, is greatly assisted by the clinical monitors. After meeting with the programmer and identifying all the checks that can be performed by the CMAs, the clinical monitor will enter data into a test DB. This data intentionally contains as many of the errors being searched for by the CMAs as possible. Our experience has shown that the more convoluted and seemingly ridiculous the data entered by the clinical monitor, the more rigorous and accurate the resultant programming package is. Regardless of how well the CRFs are written, how well the sites were instructed or the site monitoring performed, data anomalies and errors will still exist and require evaluation.

ADVANTAGES

The clear advantage provided by the CMAs is the increased efficiency it affords the clinical monitors. Obvious errors are readily identified and monitors are alerted to complex sets of circumstances requiring their clinical evaluation. All data for a given study are examined every time the CMAs are run with the monitors only having to evaluate anomalies that have developed since the last time they examined the data.

The less obvious advantages present themselves during the monitoring of on-going studies. Widespread problems with data collection at the sites are easily identified in this format. This provides an opportunity to follow up with the sites and fine tune the data collection process, resulting in cleaner CRF data coming in.

Finally, since the CMAs are run on all the entered data, they provide a check

for the entry process, as well as a simple method to identify fields requiring critical examination by the monitors through the use of special character flags.

CONCLUSIONS

The implementation of the CMAs has resulted in more efficient and accurate monitoring of clinical trial data bases at PRA. This allows us to process the data faster with an increased level of accuracy, while providing the means to supply our clients with final products in less time than was previously possible.

Contact

For further information regarding this paper, please contact:

David S. Focht
Group Leader, Data Management
Data Base Programming Department
Pharmaceutical Research Associates
2400 Old Ivy Road
Charlottesville, Virginia 22903

Phone: (804) 977-2772
FAX: (804) 293-7084

SAS is a registered trademark of the SAS Institute, Cary NC.

Acknowledgements

I'd like to thank the Data Base Programming Department at PRA and acknowledge their contributions in the development of these processes. I'd also like to recognize the input of PRA's technical writers in the development of this paper.

```

/*
*   FILENAME:
*   AUTHOR:
*   CHECKED:
*   DATE:
*   DATE:
*
*   ABSTRACT:   This is the Main program that runs all of the CMA
*               listings for <study>:
*
*               The following are the programs that will be included
*               in this program:
*               - CMA_<file1>_INCL.SAS
*               - CMA_<file2>_INCL.SAS
*               - ...
*
*   The error datasets generated by each of the INCL files should have
*   the following format:
*
*       prot      (length = $3)
*       inv       (length = $3)
*       pt        (length = $4)
*       visit     (length = $5)
*       volume    (length = $4)
*       form      (length = $8)
*       errcode   (length = $4)
*       field     (length = $8)
*       page      (length = 5.2)
*       star      (length = $1)
*       value     (length = ??)
*
*   ASSUMES:
*
*   REVISION HISTORY:
*
*       Date      Programmer      Edit Description      Checked by
*       -----      -----      -----      (Initials)
*
*   Copyright (C) 1994, Pharmaceutical Research Associates, Inc.
*   All Rights Reserved.
*/

options pagesize=64 linesize=132 missing='' mprint mlogic symbolgen;

libname  DATAPATH <engine> "<directory path>";
libname  PROGPATH <engine> "<directory path>";

/* %let run_num = 1; 05/12/94 */
/* %let run_num = 2; 05/24/94 */
/* %let run_num = 3; 06/20/94 */
/* %let run_num = 4; /* 06/25/94 */

filename no_obs "<utilities path and filename>"; /* no errors found macro
filename get_vol "<utilities path and filename>"; /* get volume informatior
filename get_star "<utilities path and filename>"; /* indicate new occurances

/*
/*   GENERIC CHECKS TOOL BOX
/*   - %aralblk      /* array alpha fields blank check */
/*   - %arnmbblk     /* array number fields blank check */
/*   - %rngchk       /* range check */
/*   - %findflag     /* identifies flagged fields */
/*
*/
%include "<tool box path and filename>";

/*
/*   FILENAME SECTION, INDICATES ALL CM LISTING PROGRAMS
*/

```

```
filename incl1 '<PROGPATH>.cma_demog_01_incl.sas';
filename incl2 '<PROGPATH>.cma_surg_02_incl.sas';
filename incl3 '<PROGPATH>.cma_pain_03_incl.sas';
```

```
filename incl# '<PROGPATH>.cma_<name>_##_incl.sas';
```

```
/* ERROR FORMAT DEFINITION - be sure the errcode variable is 4 chars */
```

```
proc format;
  value $errfmt
    "0000" = "Inquiry field"

    "0101" = "Blank Demog Field"
    "0102" = "Demog Visit Date OOR"
    "0103" = "DOB OOR"
    "0104" = "Weight OOR: 85-280 lbs"
    "0105" = "Weight OOR: 36-140 kgs"
    "0106" = "Height OOR: 48-78 ins"
    "0107" = "Height OOR: 91-213 cms"
    "0108" = "Race Other Discrepancy"
    "0109" = "Preg Test Error"
    "0110" = "Preg Test Data Discrepancy"
    "0111" = "Visit Date ne Preg Test Date"
    "0112" = "Food Time<4hrs frm Surg time"
    "0199" = "NO Demog Errors"

    "0201" = "Blank Surg Field"
    "0202" = "No Procedures Indicated"
    "0203" = "Other Proc Spec Discrepancy"
    "0204" = "Missing Surg Med Data"
    "0299" = "NO Surg Errors"
```

```
run;
```

```
/* RUN EACH CM LISTING PROGRAM TO CREATE INDIVIDUAL DATASETS -!@#$$ */
```

```
%include incl1;
run;
%include incl2;
run;
%include incl3;
run;
```

```
/* COMBINE ALL THE ABOVE CREATED DATASETS INTO ONE -!@# $ */
```

```
data mainset;
  set err_01 err_02 err_03 err_04 .... ;
run;
```

```
/* SORT THE DATASET IN ORDER TO PRINT IT IN THE DATA NULL */
```

```
proc sort
  data=mainset;
  by volume prot inv pt page;
run;
```

```

/* PRINT LISTINGS */
data _null_;
  set mainset end=lastobs;
  by volume prot inv pt page;
  retain pagenum 0 linenum;

  file cmlist
    print
    header = heading
    notitles;
  if (linenum > 60) and (_n_ ne 1)
  then
    do;
      link footer;
      put _page_;
    end;

  put @2 star @7 volume @16 prot @28 inv @39 pt @49 visit @58 page @;
  put @68 form @78 field @88 value @99 errcode $errfmt.;
  linenum = linenum + 1;

  if last.pt
  then
    do;
      put;
      linenum = linenum + 1;
    end;
  if lastobs
  then
    link footer;

  return;

HEADING:
  when = datetime();
  time = timepart(when);
  date = datepart(when);
  if _n_ = 1 then
    pagenum = 1;
  else
    pagenum = pagenum + 1;

  put @51 'PRA CMA Listings - For Internal Use only' @;
  put @115 'Date: ' @122 date date7.;
  put @58 'XYZ PHARMACUTICALS' @;
  put @115 'Time: ' @122 time time.;
  put @60 'Protocol 1234-A' @;
  put @115 'Page: ' @122 pagenum;
  put;
  put @1 'New' @6 'Volume' @14 'Protocol' @24 'Investigator' @38 'Patier'
  put @58 'Page' @68 'Form' @78 'Field' @89 'Value' @99 'Error Message';
  put '-----';
  linenum = 7;
  return;

FOOTER:
  if linenum < 60
  then
    do i=linenum to 60;
      put;
    end;
  put;
  put @4 '@ New Additions';
  return;

run;

```

PRA CMA Listings - For Internal Use only
XYZ PHARMACUTICAL
Protocol 1234-A

Date: 20Jun94
Time: 11:22:15
Page: 1

New	Volume	Protocol	Investigator	Patient	Visit	Page	Form	Field	Value	Error Message
a										NO Med Hx Errors NO Pain Cont Errors NO Term Errors Demog Visit Date OOR
	1	1234-A	001	001		2	Demog	demog_ds		
	1	1234-A	001	002		2	Demog	demog_ds		Demog Visit Date OOR
	1	1234-A	001	003		2	Demog	demog_ds		Demog Visit Date OOR
	1	1234-A	001	004		2	Demog	demog_ds		Demog Visit Date OOR
a	3	1234-A	001	016	05	4	Pain	tphalftm		Half Gone=no;Time ne Blk
a	3	1234-A	001	018	05	4	Pain	tphalftm		Half Gone=no;Time ne Blk
a	4	1234-A	001	019	07	4	Pain	tphalftm		Half Gone=no;Time ne Blk
	4	1234-A	001	025	01	8	AE	aether		Therapy Red'd Check CM Page
	5	1234-A	001	021	02	8	AE	aether		Therapy Red'd Check CM Page
	5	1234-A	001	200	2	4	Pain	Act Time		Actual Time Not +/- 3 min
	5	1234-A	001	200	3	4	Pain	Act Time		Actual Time Not +/- 3 min
	5	1234-A	001	200	4	4	Pain	Act Time		Actual Time Not +/- 3 min
	6	1234-A	001	031		2	SurgInfo	dmmealny		Meal Field Discrepancy
	6	1234-A	001	031	8	4	Pain	Act Time		Actual Time Not +/- 3 min
	6	1234-A	001	032	01	8	AE	aether		Therapy Red'd Check CM Page
a	7	1234-A	001	035	07	4	Pain	tphalftm		Half Gone=no;Time ne Blk
	11	1234-A	001	061		2	Demog	ht_in	81	Height OOR: 48-78 ins
	12	1234-A	001	065		2	SurgInfo	teeth ex		Blank Surg Field
	13	1234-A	001	070		2	Demog	dmpregny		Preg Test Error
	14	1234-A	001	081	01	8	AE	aether		Therapy Red'd Check CM Page
	14	1234-A	001	081	02	8	AE	aether		Therapy Red'd Check CM Page
	15	1234-A	001	085	01	6	Con Med	mdst_ds		Stop Date Prior to Start
	15	1234-A	001	086	01	2	SurgInfo	Meal Ti		Meal Ti < 2h
	17	1234-A	001	094	01	7	ills	ilact		Other Act/Check Comments
a	17	1234-A	001	189	06	4	Pain	tphalftm		Half Gone=no;Time ne Blk
	18	1234-A	001	195	01	8	AE	aether		Therapy Red'd Check CM Page
	20	1234-A	001	106	01	8	AE	aether		Therapy Red'd Check CM Page
	20	1234-A	001	106	02	8	AE	aether		Therapy Red'd Check CM Page

a New Additions