Database Specifications and Data Dictionary System

Ala Sweidan and Bernd Doetzkies
Warner-Lambert Parke-Davis, Ann Arbor, Michigan

ABSTRACT

The development of the Clinical Data Management (CDM) Database Specifications & Data Dictionary System (db Specs) at Parke-Davis has improved the SAS ® database development process by applying current technology in constructing the database specifications.

The CDM db Specs system uses SAS/AF® and Screen Control Language (SCL) as application development tools for the user interface. SAS/FSP® procedures, FSEDIT and FSVIEW, provide the user with a direct and easy way to edit and maintain the database specifications and corresponding documents. By normalizing the output from PROC CONTENTS the system provides the user with a flexible way to store and process the data elements needed to support data dictionary and database specifications functions.

db Specs stores the database specifications in a SAS library, rather than in a word processing document file. The system generates database specification documents, initializes the SAS database libraries at the time the database specifications are done, documents the database update and maintenance process, validates and cross-references the database to the original specifications, and aids in the standardization of the database development process.

INTRODUCTION

The development of the CDM Database Specifications & Data Dictionary System (db Specs) at Parke-Davis has improved the database development process by applying current technology in constructing the database specifications. Through this system we have been able to:

* Eliminate the use of word processing files in generating database specifications document.
* Initialize the database at the time the database specifications are done.
* Enhance productivity by saving time spent on manual tasks.
* Aids in the standardization of the database development process.

Traditionally, the database specification document was created using word processing software. The programmer would type in the necessary information to be contained in the specifications. The process of editing and coping specification documents was time consuming. Furthermore, the specifications document could not be used for any other purpose, other than to show on paper the design of the database.

The db Specs stores the database specifications in a SAS library, rather than in word processing document files. The system generates database specification documents, initializes the SAS database libraries, documents the database update and maintenance process, and validates and cross-references the database to the original specifications.

GENERATE DATABASE SPECIFICATION DOCUMENTS

The db Specs system supports the use of standard dataset and variable naming conventions. The database specifications is no longer a document, it is a database that can be used for other functions such as initializing SAS libraries and datasets and it is very easily copied to create specifications documents for other protocols.

The CDM db Specs system uses SAS/AF and SCL as application development tools for the user interface. FSEDIT and FSVIEW provide the user with a direct and easy way to edit and maintain the database specifications and corresponding documents. Users of the system take advantage of FSEDIT and/or FSVIEW to populate the specifications library because the specification library datasets have been structured to include all of the elements within the original database specifications document.

The database specifications are stored in a SAS library containing six datasets. Each dataset contains specific information, which when combined, form the database specifications document. The LIB dataset (EXHIBIT 1) contains the name of the SAS library that will contain the clinical data, the SAS engine, the study drug number (CI),the clinical protocol number (PROT), and a brief description of the protocol.

1

```
+FSEDIT CDMDD.LIB-------------------------------------------
| File Edit Search View Locals Globals Help

        CI:       111
  PROTOCOL:       111

     TITLE: Study database title_____________________
            ________________________________________

      DATE: 4/1/93

   LIBNAME: SASDB
  FILENAME: KAERESAA.PROD.D960818.SAS
    ENGINE: V607

    SYSTEM: Operating system information_____
     UNITS: _______   (CYL,TRK)
     SPACE: _______   (PRIMARY AND SECONDARY)


+-----------------------------------------------------------
```

The CI, PROT, and a brief description of the protocol are output at the top of every page in the specification document (EXHIBIT 2).

2

```
+FSLIST: AA#RN05.SASOUT.LISTING
  File Edit View Globals Help

   TITLE:     Open-Label Tolerance Study

 SYSTEM:    MVS TSO SAS VERSION 6.07
  ENGINE:   V607
  FILE NAME: KAERESAA.PROD.D960818.SAS
 @DATASET NAME  FORM  DATASET DESCRIPTION
 ____________   ____  ____________________________________
 @INCLEXCL      1     Inclusion/Exclusion Criteria
  PTINFO        2     Patient Demographics
  RXSENS        2     Drug Sensitivities
  PRIORANT      2     Prior Antibacterial Medication
  PRECOND       3     History of Predisposing Conditions
  DIAGPNEU      3     Current Episode of Nonsocomial Pneumonia
  SURG          3     Surgical Procedures
  HOSPITAL      3     Hospitalization
  MEDSURG       4     Medical/Surgical History
  INTUBATE      5     Prior and Concurrent Intubation
  THERAPY       5     Concurrent Nondrug Therapy/Procedures
  SURGPROC      5     Concurrent Elective Surgeries/Procedures
```

The LIBSPEC dataset (EXHIBIT 3) contains general information about the clinical database library. This information is stored as text describing calculations, dataset sort keys, general design and processing specifications.

3

```
FSEDIT CDMDD.LIBSPEC                                              New
File Edit Search View Locals Globals Help

    CI: 111
PROTOCOL: 111                                    NUMBER: 1
  TEXT: The data will be stored in one SAS database (Library) that
        will contain multiple SAS datasets as defined in the
        Library Specifications. The database will be saved in a
        SAS Version 5.18 format for the MVS (ISO) operating system.
        The SAS datasets will be assigned labels as defined in the
        Library Specifications.
```

This information is printed at the beginning of the specifications document and provides a general overview of the database (EXHIBIT 4).

4

Parke-Davis Pharmaceutical Research Division
Clinical Data Management
Database Specifications

DATE: 07/26/94
CI: 0999
PROTOCOL: 0999
TITLE: Open-Label Study

> THe field GBPDAY is Calculated based on the first day the patient takes the first dose of drug in the 999-999 protocol. This might be the same date as DAY1 or it may be some other day. This is possible due to the fact that patients could be on a different medication at the beginning of the study and then switch to [redacted]. This field will be used to calculate the number of days on [redacted] for any collected date.

> The database for this study will have the same name as that for the 999-999 protocol. The test database name will be KAERESAA.TEST.D999999.SAS and the production database name will be KAERESAA.TEST.D999999.SAS . This is done to ensure easy access to the data by both the Data Coordination and Biometrics departments.

The DS dataset (EXHIBIT 5) contains the name, a brief description, and the CRF number associated with each dataset in the clinical database.

5

```
FSEDIT CDMDD.DS
File Edit Search View Locals Globals Help

                CI: 111
          PROTOCOL: 111
              FORM: 8

      DATASET NAME: AE
     DATASET LABEL: Adverse Events
    DATASET NUMBER: 1

    DUPLICATE KEYS: Y
              NOBS:
           DELOBS:
             CRDATE:
             MODATE:
           MEMTYPE:
           TYPEMEM: DATA
          IDXCOUNT:
```

This information is output following the LIBSPEC data and describes the design and structure of the database (EXHIBIT 2).

The DSSPEC dataset (EXHIBIT 6) contains information that further describes each dataset when necessary.

6

```
FSEDIT CDMDD.DSSPEC                                          Obs 1
File Edit Search View Locals Globals Help

    CI: 111
PROTOCOL: 111        DATASET NAME: AE          NUMBER:      1
  TEXT: Type text in this field that describes the dataset and any
        special instructions.
```

This information is output following each dataset when the last variable in the dataset is printed (EXHIBIT 9).

The VAR dataset (EXHIBIT 7) contains the name, type, label, length, and format of every variable in the clinical database. This dataset also contains information on whether the variable is derived, used in the sort order of the dataset, and the relative order of the variable in the dataset as it occurs on the CRF. This information is output after the DS data and contains detailed description of all the variables forming each dataset (EXHIBIT 9). The VAR dataset constitutes the largest portion of the specifications document. Since each variable description is a record in the VAR dataset, there could be over 1000 observations contained in it.

7

```
FSEDIT CDMDD.VAR
File Edit Search View Locals Globals Help

                CI: 111
          PROTOCOL: 111
      DATASET NAME: AE

     VARIABLE NAME: AE
    VARIABLE LABEL: Adverse Event

              TYPE: 2
            LENGTH: 5
            FORMAT: COSTART
      FORMAT WIDTH: 0
   FORMAT DECIMALS: 0

     SORT/KEY FIELD:
CALCULATED/DERIVED: -
       BLANK FIELD: -
          REQUIRED: -
            VARNUM: 1
```

The VARSPEC dataset (EXHIBIT 8) contains more detailed information about variables when necessary. For example, the specific instructions on how a field is derived or any special instructions on how a field should be processed.

8

```
FSEDIT CDMDD.VARSPEC                                             Obs
File Edit Search View Locals Globals Help

    CI: 111        DATASET NAME: AE
PROTOCOL: 111      VARIABLE NAME: AECONT      NUMBER:      1
  TEXT: Type text in this area that describes the Variable and
        includes any special instructions. An example of a special
        instruction for this field might be: 1=continuing,
        else blank
        This instruction tells the reader that a one in this field
        = continuing. Any other number or character leaves the
        field blank.
```

This information is placed after each variable as it is listed in the specifications document (EXHIBIT 9).

9

PARKE-DAVIS PHARMACEUTICAL RESEARCH DIVISION
CLINICAL DATA MANAGEMENT
DATABASE SPECIFICATIONS

DATE: 07/26/94
CI: 0999
PROTOCOL: 0999
TITLE: Open-Label Study

FORM: 1
DESCRIPTION: Inclusion/Exclusion Criteria
DATASET NAME: INCEXCL2

SAS NAME		FIELD DESCRIPTION	TYPE	LENGTH
CI	(1)	CI	N	4
PROT	(2)	PROT	N	4
TRIAL	(3)	Site Number	N	3
CEN	**	Site Number	N	3
PTINTL	(5)	Patient Initials	C	4
VISIT		Visit Number (TYPE AND LENGTH DEFINED AS STORED IN THE SAS DATABASE)	C	6
OBSDATE		visit Date(European Date dd/mm/yy)	C	6
SDOBSDY	**	Study Day of Visit Date	N	8
INCLUD01		Inclusion Criteria 1	N	1

All inclusion exclusion variables will have a 1 or zero for a value.

() DENOTES SORT/KEY FIELDS
* DENOTES BLANK FIELD
~DENOTES DERIVED/CALCULATED FIELD

The database specifications document can be created from existing databases by normalizing the output from the PROC CONTENTS to generate the LIB, DS, and VAR datasets. The data elements used to generate the specifications document are easily manipulated and queried through Ad Hoc programming. Because the system uses SAS/AF and SCL, users can access the SAS Display Manager System (DMS) and the operating system (IBM MVS TSO®) to support Ad Hoc queries and programming. Changes that need to be made throughout the entire library can easily be made by a simple program run in interactive SAS.

INITIALIZE SAS DATABASE LIBRARY AND DATASETS

Initialization of the database is the process of defining the SAS library, datasets, and corresponding variables, and under the MVS platform, allocating storage space. This process defines the SAS engine for the database, the dataset names that form the database, and the variables and their attributes that form each dataset.

Previously, initializing the database required three steps. The first step was the allocation of the SAS library. The second step was defining the output specifications from the data entry software Entrypoint 90 Plus®. The third step was to code a SAS program to read in the input data and define the dataset, variables and assign their respective labels.

The CDM Database Specifications & Data Dictionary System compiles the data from the specifications library datasets and initializes the SAS library, datasets, and variables and generates the PROC CONTENT report in one step. The selection list for databases to be initialized is built in the SAS/AF program using the LIB dataset (EXHIBIT 10). The new system streamlines the initialization process explained in the previous paragraph.

10

```
+Initialize Database-----------------------------------------------
 Command ===>

 HELP    GOBACK    SEARCH    BACKWARD    FORWARD

 * Move the cursor to one of the selections and press <ENTER>.

      CI:      111
 Protocol: 111
    Title: [redacted] in the Treatment of Secondary Bacterial Infections
           of Acute Bronchitis
  DB Name: AA@DP55.TEST.D111111.SAS
```

Dataset and variable labels are assigned during the initialization process and are not needed in the SAS programs that updates the database. Also, any changes to the datasets and variables can be made after the database has been populated. The system contains an option (EXHIBIT 11) to re-initialize an existing database after making additions and/or changes to variable attributes, and keep or delete all of the current data.

11

```
+Database Initialization Options-----------------------------------
 Command ===>

 HELP    GOBACK

              WARNING: The selected database already exists.
                       AA@DM05.TEST.D111111.SAS

 * Select the processing options and then select <OK>.

   Do you want to re-initialize the Database ?                YES   NO
   Do you want to keep or delete the existing observations ?  KEEP  DELETE
                                  < OK >
```

The database initialization process no longer requires three different steps. The SAS program used to update the database has been made more efficient by eliminating all the label statements and handling any variable attribute conflicts.

VALIDATE and CROSS-REFERENCE DBs and SPECIFICATIONS

The CDM programmer is responsible for ensuring that the database structure matches that stated in the database specifications document. In the past this process required manual comparison of the database specification document and PROC CONTENTS output. Each field had to be manually checked to ensure its presence in both documents. To do that for over 1000 data points per database was very time consuming.

The CDM db Specs system allows programmers to take full advantage of electronic cross referencing. The system provides for a series of cross-reference reports (EXHIBIT 12) and options (EXHIBIT 13) that assist the programmer in the validation process and substantially increases efficiency and accuracy.

12

```
+DB Specification Cross-Reference Reports--------------------------
 Command ===>

 HELP    GOBACK

 * Move the cursor to an option and press the <ENTER> key.

   Existing Database to Existing Specifications
   Compare Two Existing Databases
   Compare Two Database Specifications
```

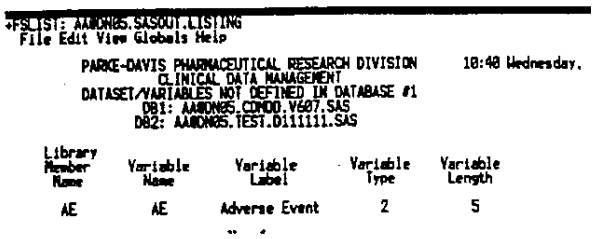
13

```
+Cross-Reference Existing DB and Specs-----------------------------
 Command ===>

 HELP    GOBACK

 * Type the required information (without quotes).

   Specs Library ==> _____________________________________________
              CI ==> _____
        Protocol ==> _____
         SAS DB ==> _____

 * Select at least one of the report options and then select <OK>.

   Dataset/Variables Defined in Database and Specifications
   Dataset/Variables not Defined in Database
   Dataset/Variables not Defined in Specifications
   Variables Defined as Character and Numeric
   Variables Defined with Different Labels
   Variables Defined with Different Lengths
   Variables Defined with Different Formats
                                  < OK >
```

The programmer can use a macro program to do a cross reference check between the database specifications document and the database to ensure that a precise match exists. PROC COMPARE is used to help the programmer do a cross-reference check between two existing specifications libraries to identify valid and invalid differences. Another macro program can be used to do a cross-reference comparison between two existing databases to easily identify similarities and differences.

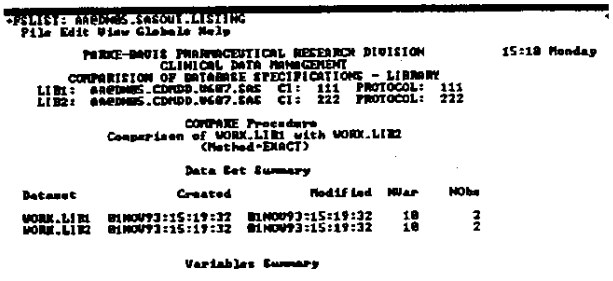
The result from each of these cross-references is presented in a report (EXHIBIT 14,15). The programmer can select the type of comparisons to be performed and listed in the reports. Each cross reference can be performed with several options, matching or non-matching dataset names, variable names, and variable attributes. Thus, conflicting variable attributes, missing variables, or extra variables can easily be identified.

14



```
*FSLIST: AAMDM05.SASOUT.LISTING
  File Edit View Globals Help

        PARKE-DAVIS PHARMACEUTICAL RESEARCH DIVISION      10:40 Wednesday,
                      CLINICAL DATA MANAGEMENT
        DATASET/VARIABLES NOT DEFINED IN DATABASE #1
                DB1: AAMDM05.CDMDD.V607.SAS
                DB2: AAMDM05.TEST.D111111.SAS

  Library
  Member     Variable    Variable      Variable    Variable
  Name       Name        Label         Type        Length

  AE         AE          Adverse Event 2           5
```

15



```
*FSLIST: AAMDM05.SASOUT.LISTING
  File Edit View Globals Help

        PARKE-DAVIS PHARMACEUTICAL RESEARCH DIVISION          15:18 Monday
                      CLINICAL DATA MANAGEMENT
      COMPARISION OF DATABASE SPECIFICATIONS - LIBNAME
LIB1:  AAMDM05.CDMDD.V607.SAS  C1:  111  PROTOCOL:  111
LIB2:  AAMDM05.CDMDD.V607.SAS  C1:  222  PROTOCOL:  222

                    COMPARE Procedure
          Comparison of WORK.LIB1 with WORK.LIB2
                    (Method=EXACT)

                    Data Set Summary

Dataset          Created          Modified         NVar   NObs

WORK.LIB1  01NOV93:15:19:32  01NOV93:15:19:32    10      2
WORK.LIB2  01NOV93:15:19:32  01NOV93:15:19:32    10      2

                    Variables Summary
```

The process of validating the database has been made more efficient, effective and interesting through the use of cross-referencing procedures.

CONCLUSION

One of the best features of db Specs is that while database specifications are developed, the databases are also being developed. When the specifications are finished, the database can be immediately initialized with all the data elements from the specifications. Then the cross-reference tools can be used to check and validate your work.

The cross-reference tools can also be used to find similarities or differences between two sets of specifications or two databases.

Acknowledgements

We would like to acknowledge and thank Vicki Rothmeier for all of her assistance and support in developing the User's Manuals, documentation for the system, and preparing this paper.

We would also like to thank Richard Weinkauf for his support during the development of the system.