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/AF provides 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 copying 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 FS VIEW 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 FS VIEW 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. **LIB**
   File Edit Search View Locales Globalis Help
   - PROTOCOL: 111
   - TITLE: Study Protocol
   - DATE: 1/1/93
   - LIBNAME: SASDB
   - ENGINES: V607
   - SYSTEM:
   - SPACE: 1024 (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. **DAT**
   File Edit View Globals Help
   - TITLE: Study Protocol
   - ENGINES: V607
   - SPACE: 1024 (PRIMARY AND SECONDARY)
   - DATASET NAME: EXPIRIA.PROD.D600018.SAS
   - DATASET DESCRIPTION:
   - DATASET NAME: TECHNICAL CRITERIA
   - PATIENTS
   - PATIENTS' ABNORMALITIES
   - DRUGS
   - DRUG SENSITIVITY
   - DRUG ANTIBIOTIC
   - DRUG HISTORY
   - DIAGNOSIS
   - DISEASES
   - MEDICAL HISTORY
   - HOSPITALIZATION
   - SURGERY

The db Specs system is designed to support 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 LlBSPEC 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.

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

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

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. Its information is output after the DS data and contains detailed description of all the variables forming each dataset (EXHIBIT 9).

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.

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

The OSSPEC dataset (EXHIBIT 5) contains information that further describes each dataset when necessary.
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 COM 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 SASIAF program using the LIB dataset (EXHIBIT 10). The new system streamlines the initialization process explained in the previous paragraph.

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.

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.

### 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.

SAS, SAS/AF, SAS/FSP, are registered trademarks or trademarks of SAS Institute, Incorporate in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.