Paper 204

Using SAS/AF® to Create Applications for Looking Up and Coding Adverse Events and Generic Drug Names in Clinical Trials

Karen B. Fowler, Emily A. Mixon, UAB Department of Pediatrics, Birmingham, AL

ABSTRACT

Most clinical trials collect medical adverse events (AEs) and usage of medications from participants during the study. Study personnel record this information and corresponding codes onto case report forms for data entry into study databases. The recording of specific medical terms for AEs and/or generic names of medications usually requires reviewing bulky manuals. We developed two applications that allow the user to see a list of medical terms or drug names and their respective codes at one time and the ability to subset the list of terms using either the medical term or drug name. Both applications use a Toolbar, an Extended Text Entry field and a Data Table combined to allow the user to display a list of medical terms or drug names and codes. The Data Table displays a subset of the data based on the selected letter on the Toolbar. A row in the Data Table may be selected by the user to display a frame with a Data Form that contains further information about the medical term or drug name. These SAS/AF® (v. 6.12) applications reside on the computer desktops of the study personnel making it simple to look up medical terms or drug names.

INTRODUCTION

Clinical trials collect information about any medical AEs and concomitant medications from participants at specified intervals throughout the duration of the trial. Upon obtaining this information the clinical study personnel record these data onto case report forms (CRFs) for data entry into study databases. Collection of AEs requires the clinical personnel to describe in medical terms the specific AE and the severity of the AE. To assist in the original description of AEs, the Food and Drug Administration (FDA) developed the terminology called COSTART (The Coding Symbols for a Thesaurus of Adverse Reaction Terms). The COSTART terms provide a uniform method for reporting adverse events to the FDA. The COSTART term also links to a primary and secondary body system classification allowing for summary of AEs by body system when reporting events to the FDA. Similarly, recording and summarizing concomitant medications taken by participants during the trial is needed for reports. However, many times the participant only reports the brand name of the drug or drugs taken during the trial requiring study personnel to lookup and record the generic drugs for summary reports. Recording AEs and concomitant medications on the CRFs required the study personnel to review bulky manuals each time they completed a CRF to identify COSTART terms and generic drug names. To simplify these activities for study personnel, we developed two portable SAS/AF® applications. The first application searches COSTART terms and the other application searches drug names to identify the generic drug. Both applications use a Toolbar, Extended Text Entry and Data Table combined to display a list of AEs or drug names. The list may be subsetted by specifying a letter on the Toolbar or by typing characters on the Extended Text Entry to further subset the display of the table. Double clicking on the row in the Data Table displays a frame with a Data Form displaying the selected row allowing the user to browse the row.

COSTART TERM SEARCH APPLICATON

This application uses four frames. The first frame displays two Icons and a Command Push Button to exit the application (Fig 1). These Icons allow the user to choose to either search by COSTART term or by COSTART number. The only SCL behind the frame uses a CALL DISPLAY routine to open either the COSTART Term frame or the COSTART Number frame depending upon the Icon pressed.

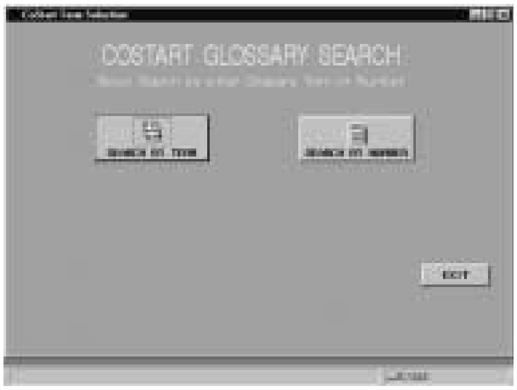

Figure 1

COSTART TERM FRAME

The COSTART Term frame contains a Toolbar, Extended Text Entry and Data Table created from three subclasses of the Toolbar, Extended Text Entry and Data Table classes, respectively (Fig 2). These subclasses were created following the description, methods, and SCL listed in an Observations article published in the First Quarter, 1996 (Brideson, 1996).

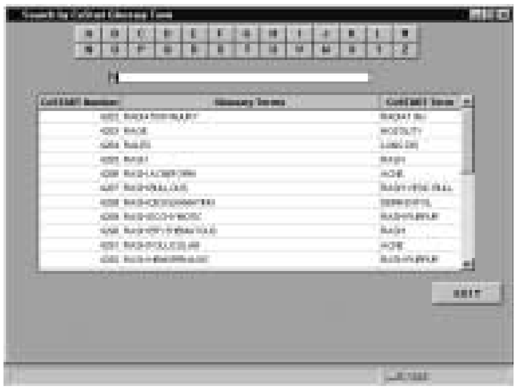

Figure 2

For the Toolbar subclass the _UPDATE_VALUE_ method was overrided in the methods window and the SCL for the Toolbar subclass was written to control the _UPDATE_VALUE_ method. In this method, the label of the pressed Toolbar button is retrieved and the DATA_CHANGE event is sent out with the text. The Extended Text Entry will use the text sent by the DATA_CHANGE event to subset the COSTART terms data in the application. After defining the Toolbar subclass the Toolbar was created and placed on the frame. Attributes for the Toolbar were entered and each Toolbar button was set to a letter in the alphabet (Fig 3).

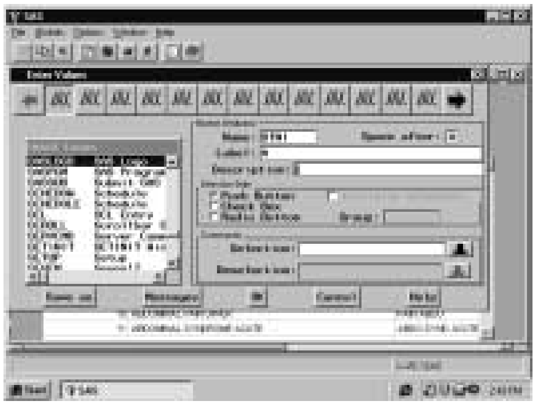

Figure 3

The new Extended Text Entry subclass was built from the Extended Text Entry class with three methods, _INIT_, _FEEDBACK_, and _SET_TEXT_, overrided in the methods window of the subclass. The methods were then defined in the SCL code for the Extended Text Entry subclass. In the INIT method when the DATA_CHANGE event is received the _SET_TEXT_ method is run so that the Extended Text Entry modifies based on the text passed in the event. If text is typed into the Extended Text Entry the _FEEDBACK_ method is run. Using a TABLE_WHERE_DATA event and the _GET_TEXT_ method the Data Table is subsetted by COSTART terms based on the text passed in the event. After creating the Extended Text Entry subclass, the Extended Text Entry was added to the frame (Fig 4).

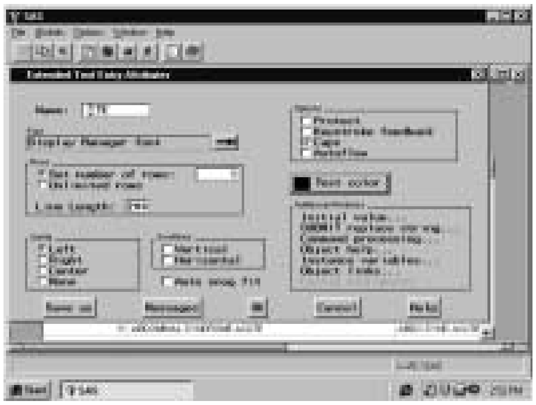

Figure 4

The Data Table subclass was created from the Data Table class with several new instance variables created. The instance variables were selected and defined in the Instance variables window in the Data Table subclass. The _INIT_ method was overrided and new methods were added to the subclass by defining them in the methods window. These newly created methods and instance variables are handled in the SCL for the Data Table subclass. These methods and instance variables were used to define the components of a WHERE clause such as the column selected, where operators (i.e., like, equal, etc.), and ending text options (i.e., %, _, etc.). The Data Table was added to the frame with the columns and rows customized in the Data Table attributes window. The SCL for the frame defines the column (COSTART term), operator ('LIKE') and ending text ('%') to be used for the WHERE clause for subsetting the COSTART terms within the Data Table. To display the Data Form when a selection is made from the Data Table an instance override of the Data Table's _SELECT_ method was created. The modified _SELECT_ method in the SCL executes when a row is selected in the Data Table. The SCL for the COSTART Term frame calls another SCL program that uses a SCL list that contains the text for a _FIND_ROW_ method. This SCL list is passed to the frame with the Data Form executing the _FIND_ROW_ method resulting in the display of the selected row from the Data Table within the Data Form frame.

COSTART NUMBER FRAME

The COSTART Number frame contains an Extended Text Entry and a Data Table created from the two subclasses of Extended Text Entry and Data Table classes (Fig 5). These subclasses and methods used for this frame are described above in the COSTART Term frame. The SCL for the frame defines the column (COSTART number) and operator ('EQ') to be used for the WHERE clause for identifying the row of the selected COSTART number within the Data Table. The _FIND_ROW_ method that displays the selected row from the Data Table within the Data Form frame is the same SCL described in the COSTART Term frame.

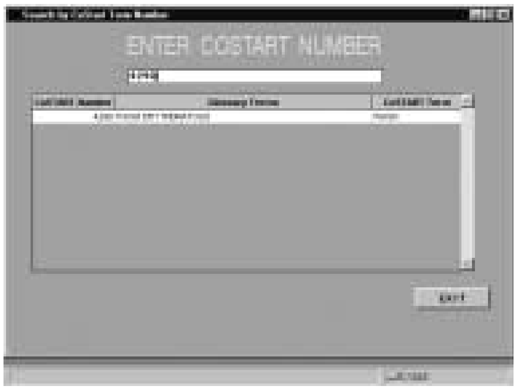

Figure 5

COSTART DATA FORM FRAME

The final frame in the application contains the Data Form. The Data Form displays the row selected in the Data Table from either the COSTART Term or COSTART Number frames (Fig 6). The Data Form attributes are set so only browsing of the data may occur. There is no SCL associated with the Data Form. The frame SCL uses the FIND_ROW_ method to locate the record selected in the Data Table as described in the COSTART Term frame.

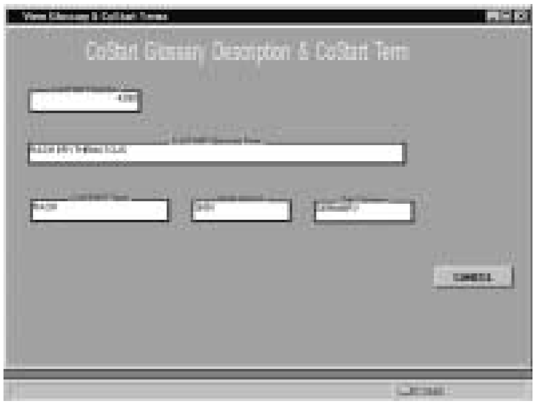

Figure 6

DRUG NAME SEARCH APPLICATION

The four frames and corresponding SCL used in this application are basically identical to the COSTART Term Search application except for the Graphics Text, Data Table and Data Form used. The Drug Name Search application uses the same object subclasses and methods described in the COSTART Term Search application. The Drug Name Search application calls the data set that contains the brand drug and generic names and drug codes. The Search Drug

Name frame that contains the Toolbar, Extended Text Entry and Data Table allows the user to search by drug name (both generic and brand names) as seen in Figure 7.

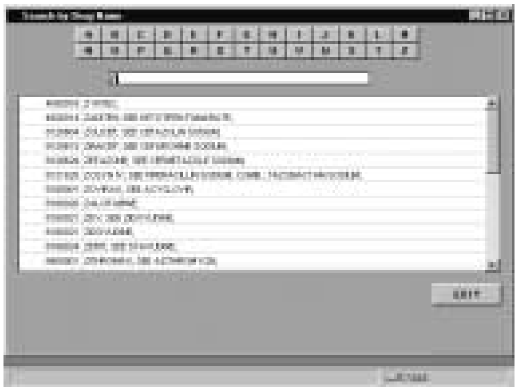

Figure 7

The Search Drug Number frame contains only the Extended Text Entry and Data Table using the same methods and similar SCL as the Search Drug Name frame. The frame containing the Data Form is accessed using the _FIND_ROW_ method previously described in the COSTART Term Search application (Fig 8).

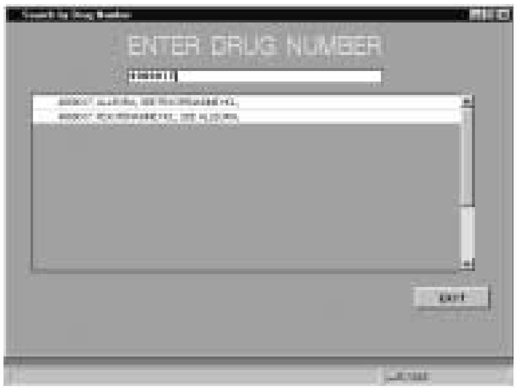

Figure 8

The Data Form frame has been changed to reflect the data (Fig 9). However, all SCL and methods to populate the Data Form with a row from the Data Table are identical to the COSTART Term Search application.

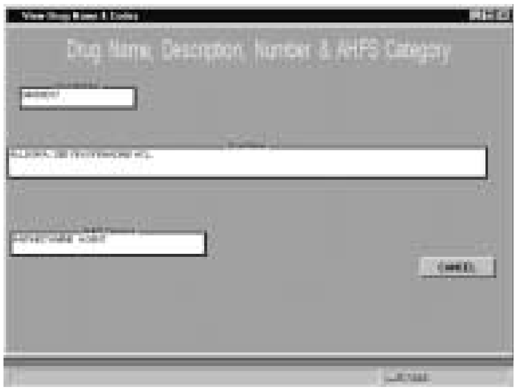

Figure 9

CONCLUSION

The use of the subclasses of Toolbar, Extended Text Entry and Data Table makes it possible to develop search and browse applications for a variety of uses. These subclasses, methods and SCL are the pieces needed to build a searchable SAS/AF® frame application. These components for data searching have become standard in many of the applications developed by our group. For our study personnel having searchable COSTART terms and drug names makes completing study tasks somewhat easier and more time efficient. The portability of these applications make them readily available for all users.

REFERENCES

Brideson, Patti M. (1996), “Creating a Contacts Application Using New Classes in SAS/AF Software”. Observations, First Quarter 1996.

Carpenter, Ann E. and Leone, Lynn P. (1996), “Introducing a GUI Approach to Data Entry Using Data Table and Data Form Objects in SAS/AF FRAME Software”. Observations, First Quarter 1996.

Rocket, Ann Carpenter. (1997), “An Introductory Overview of the Data Form and Data Table Objects in SAS/AF Frame Entries”. Proceedings of the Fifth Annual Southeast SAS Users Group Conference. USA.

SAS Institute Inc. (1997), Building SCL Applications using Frame Entries Course Notes, Cary, NC: SAS Institute Inc.

Wilkins, Scott. (1997), “Data Table and Data Form Enhancements in Release 6.12". Observations, First Quarter 1997.

ACKNOWLEDGMENTS

This work was supported in part by the National Institutes of Health, National Institute of Allergy and Infectious Diseases Grants, 1P01AI43681 & 5U01AI41025 and National Institute on Deafness and Other Communication Disorders Grant 1R01DC02139. The authors thank Paul McNeal and Felicita David for earlier versions of similar applications. The authors would also like to thank Ann Carpenter Rocket and Lynn Leone at SAS Technical Support for their helpful advice and suggestions in the development of the SCL.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Karen B. Fowler
UAB Department of Pediatrics
1600 7th Ave So, CHT 752
Birmingham, AL 35233
Work Phone: 205-939-6685
Fax: 205-975-6549
Email: kfowler@uab.edu