

Automating Data Transfer from SAS® to Microsoft Access®

Kayhan Najmabadi, STATPROBE, Inc., Ann Arbor, Michigan

Carl R. Haske, STATPROBE, Inc., Ann Arbor, Michigan

ABSTRACT

Transferring data between the SAS System and other database platforms is commonplace in the management of clinical data. Some features built into Windows allow the SAS System to communicate and share data with other Windows applications. These tools include OLE Automation, Dynamic Data Exchange (DDE), and Open Database Connectivity (ODBC). By taking advantage of these tools, the programmer can develop reusable systems to render the transfer of data between platforms.

A problem arises with how to handle formatted variables. When SAS writes the values of variables, it uses the format associated with a variable to print the values. However, when SAS data are transferred to a derived database in another platform, the decoded values of data are often lost. The resulting exported data contain cryptic codes that are not meaningful to the end user who receives the derived database.

STATPROBE, Inc., has developed a specialized system to export data from SAS to MS Access that automates the decoding of the SAS data through the format library and executes a verification procedure to guarantee the validity of the data transfer. Although the system specifically targets MS Access, the techniques presented in this paper can be adapted for other systems.

INTRODUCTION

This paper discusses a system to transfer data between SAS and MS Access. The motivation behind developing this system was to provide one of our customers, QLT PhotoTherapeutics Inc., with a MS Access database, the content of which was clinical data that had been constructed and managed in a SAS environment.

Four main problems needed to be addressed in the design of this system. First, the SAS database was constructed so that many of the variables were coded, and the customer wanted the decoded values stored in the MS Access database. Second, because the problem involved the conversion of several SAS databases, the process would need to be repeated multiple times. Third, for each database that is converted, a very fast turnaround time was needed after the SAS database was declared final. Finally, a test procedure was needed to ensure the validity of the data translation.

This paper is divided into three parts. In the first part, we present the general outlines of the methods used. The second part discusses the nature of SAS macros systems and a SAS/AF® interface that are used by the data transfer and validation system. These macro systems are generic and stand on their own merit. The application macros and AF interface are discussed in detail in the authors' paper "A SAS Macro System to Automate the Decoding and Validation of Data," the second paper of the present group of three. The third section discusses the use of OLE to initiate and control a SAS session and the implementation of ODBC. We refer the reader to the authors' "Interfacing Microsoft Access and SAS Using OLE and ODBC," the last

paper in the present group of three, for details of the Visual Basic coding behind the OLE and ODBC implementation. The third section discusses the integration of the system. Concluding remarks follow.

The development of the system took place in Windows 95. Although many of the techniques used in MS Access are platform dependent, the SAS component could be easily ported to all operating systems, likely with no additional coding.

PROCESS FLOW

The system consists of four primary stages that correspond to four physical databases. The databases used are:

- Original SAS database
- Decoded SAS database
- Customer's Access database
- Validation SAS database

Figure 1 illustrates this process.

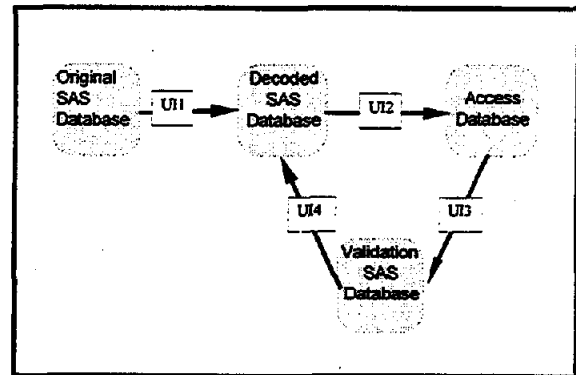


Figure 1. Process flow

The *original* SAS database is decoded to create the *decoded* SAS database. The decoded SAS database is converted to the *Access* database to be delivered to the customer. The Access database is converted to the *validation* SAS database, which is validated against the decoded SAS database.

The entire process involves four key stages, comprising three data conversions and one validation. Each stage is handled by an independent interface object denoted in the diagram by the interface codes UI1, UI2, UI3, and UI4. By UI we mean a User Interface or automation tool that is acting during the process. Table 1 describes the type and nature of these objects.

The systems UI1 and UI4 allow the options of an interface in SAS/AF or an automated OLE process directed from an application external to SAS. The AF interface extracts the decoding and validation systems into an independently executed process within the SAS system. This provides a useful database management application separate from the entire SAS-to-Access system. Alternatively, the decoding and validation system can be automatically executed from

MS Access by using OLE. UI2 and UI3 are systems run in MS Access that leverage the functionality of ODBC to transfer data between SAS and MS Access formats.

Table 1. Phase Interfaces

Interface	Type	Description
UI1	OLE	OLE automation: Access launches the decoding macros system in SAS.
	AF	User launches decoding macros from an AF frame.
UI2	ODBC	ODBC is used for importing SAS database to MS Access.
UI3	ODBC	ODBC is used for exporting MS Access tables to SAS database.
UI4	AF	AF frame is used to compare two databases.
	OLE	OLE automation: Access launches SAS process to compare two SAS databases and subsequently ends the process.

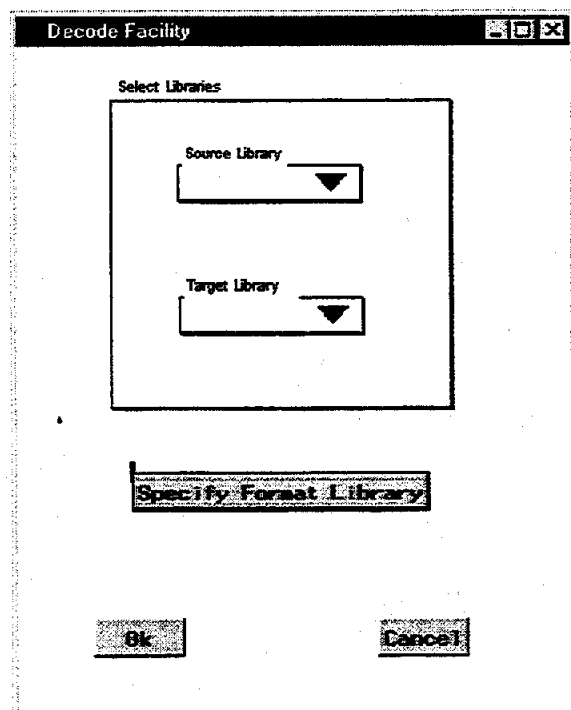


Figure 2. Decode facility

SAS MACROS AND INTERFACE

The system in SAS to decode a database and validate a database uses a macro library and provides an AF interface to the macros. The macros are divided into decoding

macros and validating macros. Figure 1 shows the AF interface to the decode macro that gets the input output data libraries. Figure 2 shows the facility that interfaces to the macro that decodes a single data set.

The decoding macros essentially translate the data by scanning the database to determine the variables that have associated formats. This scanning process is achieved by using utility macros that access the SAS dictionary tables through PROC SQL. The system then creates new data sets and applies the formats to the variables, outputting the results to a new data library.

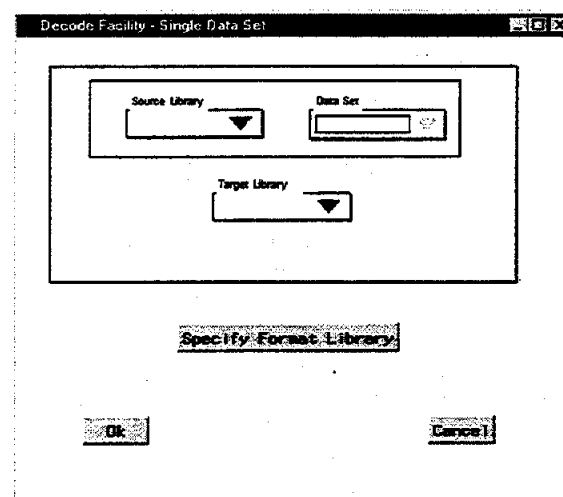


Figure 3. Decoding a single data set

The validation system compares two data libraries for differences and produces a report. This system also has an AF interface similar to the decode facility, and it can also be run via an OLE interface from an external application. This system is used to guarantee that the transfer of data to a file format other than SAS is valid. The details of the underlying system are available in the authors' paper "A SAS Macro System to Automate the Decoding and Validation of Data."

OLE AND ODBC

ODBC: This technique is used for the data transfer. ODBC, or Open Data Base Connectivity, was developed by Microsoft and allows ODBC-compliant software packages to exchange information. By using the ODBC API, applications can access data stored in a variety of personal computer, minicomputer, and mainframe database management systems (DBMSs), even when each DBMS uses a different data storage format and programming interface. Since both MS Access and the SAS system are ODBC compliant, we have used ODBC twice. We use it first when we export the decoded data set to a Microsoft Access table. Second, in order to validate the process, we need to compare the original data set with its transferred version, and we use ODBC once again to send the Access table back to the SAS System.

OLE is Object Linking and Embedding. By implementing this technique into our work we were able to exploit the server ability of the SAS System version 6.12. We submitted the decoding macro system from Access.

The following diagrams give a more detailed account of the construct.

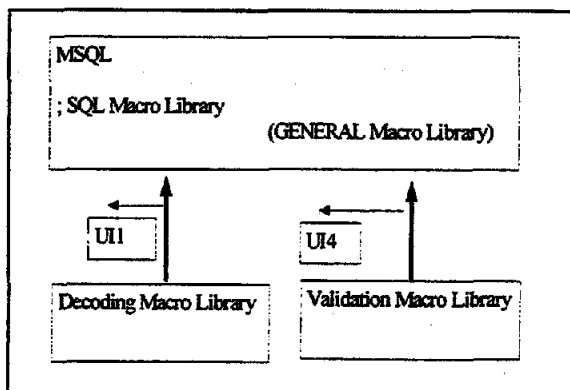


Figure 4. Structure of UI1 and UI4

The system of macros that we have implemented has two distinct levels. In a higher level, a general macro library, which is independent of this specific data transfer and makes use of SQL techniques, is called by macros that are specific to our data transfer case. Figure 2 is applied to both UI1 and UI4. The second paper of this group will discuss this part in more detail.

In the next diagram, ODBC is shown as the agent between MS Access and the SAS system. The third paper of this group will give more explanation of the Visual Basic code that uses this technique.

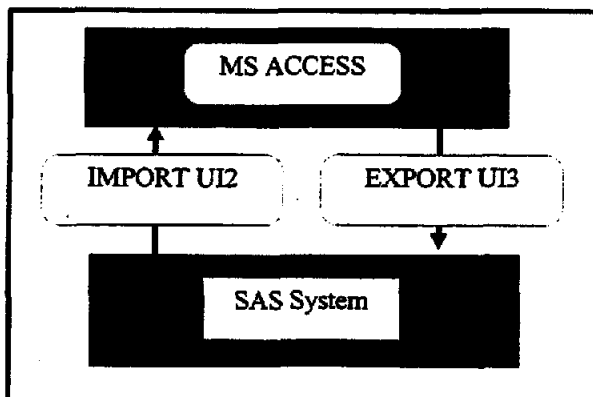


Figure 5. ODBC mechanism

SYSTEM INTEGRATION

The starting platform used by our team is MS Access. Therefore, we have built a system around MS Access, which is a Visual Basic-oriented application. However, we needed SAS coding to "decode" data sets. This SAS macro system was developed at STATPROBE and is part of our macro library. The Visual Basic code behind MS Access calls the macro system by means of OLE Automation. With ODBC, we were able to transfer data from SAS to Access and vice versa. The ODBC facility is well described in earlier work (e.g., Riba and Riba [1996]) and need not be emphasized here. What was new in our approach was the use of OLE Automation for initiating an en-

tire system based in SAS and controlling it from the Access platform. An alternative to this solution would be the implementation of SAS/AF as a user interface. This situation will result in a self-contained SAS application. The Access platform will need only to launch this SAS application. Both cases are revealed to be feasible and rewarding.

CONCLUSION

The system presented here shows how to integrate SAS with another application by using OLE and ODBC. Using this system, we were able to transfer, validate, and package 10 databases, each containing about 30 tables, in approximately 16 hours. This data transfer system was originally developed to solve a specific problem and some generic utilities were produced as a result.

REFERENCES

- Geiger, Kyle (1995). *Inside ODBC*. Redmond, WA: Microsoft Press.
- Laws, Reesa, and Longacre, Mary (1996). "Bridging the Gap: Using ODBC to Grab Data Stored in a SAS® Data Warehouse for Reporting in Microsoft Access," *Proceedings of the Twenty-First Annual SAS Users Group International Conference*, 724-727, Cary, NC: SAS Institute Inc.
- Riba, S. David, and Riba, Elizabeth A. (1996). "ODBC: Windows to the Outside World," *Proceedings of the Twenty-First Annual SAS Users Group International Conference*, 548-557, Cary, NC: SAS Institute Inc.
- SAS Institute Inc. (1994), *SAS ODBC Driver Technical Report, User's Guide and Programmer's Reference*, Release 6.10, Cary, NC: SAS Institute Inc.

ACKNOWLEDGMENTS

The authors would like to thank Alexandra Mancini of QLT PhotoTherapeutics Inc. for generously according her authorization to use data samples from the PHOTOFRIN study. The authors also thank Paul Schwankl for his assistance in the preparation of this paper.

SAS and SAS/AF are registered trademarks of SAS Institute Inc., and MS Access and Windows 95 are trademarks of Microsoft in the USA and other countries. ® indicates USA registration.

AUTHORS' ADDRESSES

Kayhan Najmabadi, Ph.D.
STATPROBE, Inc.
3885 Research Park Drive
Ann Arbor, MI 48108
(313) 769-5000 x161
E-mail: knajmabadi@statprobe.com

Carl R. Haske, Ph.D.
STATPROBE, Inc.
3885 Research Park Drive
Ann Arbor, MI 48108
(313) 769-5000 x115
E-mail: chaske@statprobe.com