SAS® System to IDMS/R™ Interface Overview
Nancy J. Wills, SAS Institute Inc., Cary NC

This paper is designed to provide an overview of the current development being done on an interface between the SAS® System and IDMS/R™. The paper is geared to experienced SAS users who understand the concept of a data base and are familiar with IDMS/R.

IDMS/R is Cullinet Software Inc.’s data base management system (DBMS). It runs under OS, DOS/VSE and CMS operating systems. IDMS/R has both network and relational capabilities. The data bases are controlled by IDI™, the Integrated Data Dictionary.

The interface between IDMS/R and the SAS System is being developed because of the interest expressed by both SAS and IDMS/R users. We have received numerous letters, phone calls, and verbal requests from customers. The interface was a very highly requested item on our SASware Ballot.

Let’s take a look first at the network capabilities of IDMS/R. To begin with, let’s define a set. A set is a relationship established between two or more record types. One record type is the owner of the set; the other(s) is a member. Record types can belong to more than one sets. A record type can be an owner of one set and a member of another. Or, a record type can be a member of more than one set. These sets and their interweaving relationships make up a network.

In order to move through the data base, each record type contains pointers to other record types in its set(s). There are three kinds of pointers: next (required pointer), prior, and owner. The next pointer points to the next record type in the set, regardless of whether the record type is an owner or a member of the set. The prior pointer works the same way, except that it points to the prior record type. The owner pointer points from a member record type to the owner record type. Through the use of these pointers, a program can navigate through the network and travel a specified path through one or many sets.

The Logical Record Facility (LRF™) can be used to make navigation through the data base easier. The data base administrator (DBA) defines logical records in the subschema. Logical records are composed of elements selected from multiple data base records and the paths used to get to the records. The paths are designed to process expected program requests. Programs can then use the Logical Record Facility to access elements from many records as if they were in one record.

Now let’s look at IDMS/R’s relational capabilities. A relational data base contains data tables. Data tables are two-dimensional tables that contain rows and columns. Data tables can be stored tables or views. Stored tables are physically stored in the data base (defined in the schema). Views are real-time images of data stored in the data base. Views can be derived from existing tables, records, or other views. Views do not contain data themselves but contain a definition of the data. Tables and views are defined to IDMS/R through the Automatic System Facility (ASF™), or a DBA can define logical records to represent them. Both can be accessed through the Logical Record Facility.

The interface with the SAS System will use the Logical Record Facility to access data from the data base. The interface will be a SAS procedure that will run under full-screen, line, and batch modes. Currently under development is a procedure to extract data from IDMS/R and build a SAS data set.

To access the procedure, under full-screen mode, type:

`PROC IDMSEX; RUN;`

The first screen (Figure 1) is the Data Base Access Panel, which requires information about the data base and the SAS data set. In the OUTPUT SAS DATA SET field you enter the SAS data set where the extracted data are to be written. In the SUBSCHEMA NAME field you enter the subschema where the record or table is defined. Use the SCHEMA NAME and SCHEMA VERSION NO. fields to indicate the schema and version number that contains the subschema. Use the RECORD/TABLE NAME to indicate the name of the logical record, table, or view that will be accessed from the data base. In the NODENAME field you enter the name of the node or central version that will process the program’s database requests. This field is filled in if you are running a Distributed Database System (DDS™), which is composed of multiple central versions.

When all the required data are correctly entered on the Data Base Access Panel and the ENTER key is pressed, the Extract Data Selection Panel appears (Figure 2). This screen contains all the records and elements defined to that logical record, table, or view. At the top of the screen is information about the logical record, table, or view as well as the SAS data set defined on the Data Base Access Panel. The selection information is broken down into columns. The first column is the FUNCTION column. In this column you would enter an S to indicate the elements that are to be retrieved from the data base and written to the SAS data set. If you enter an N next to a record name, all the elements for that record will be selected. The next column is the IDMS/R RECORD or ELEMENT NAME column. If a record name is in this column, the name is followed by asterisks and its SAS name contains asterisks. The next column is the SAS NAME column. In this column, enter a SAS name next to the element that was selected. If this column is not filled in for selected elements, a SAS name will be generated from the first eight nonblank characters of the element name. All dashes will be removed from the name. The last column, FORMAT, is the SAS format that was derived from the IDMS/R picture and usage of the element. The format can be changed by typing over it. At the bottom of the screen is an area for WHERE CRITERIA. In this area an IDMS/R where clause can be entered. The where clause should follow the PL/I syntax rules. When all the information is entered and the END or ENTER key is pressed, SAS names will be generated, duplicate or incorrect SAS names will be flagged as errors, and incorrect SAS formats will be flagged as errors. When all the information is correct and the END key pressed, the data will be extracted from the data base and written into a SAS data set and the Data Base Access Panel will reappear.

From this panel the newly created SAS data set can be browsed by typing a B on the command line. More SAS data sets can be created or the procedure can be ended. Now the SAS data set is ready to use with other SAS procedures or in the DATA step.

The IDMS/R interface will also offer the ability to break your extraction process into two parts. The first part can be done with the full-screen panels, but the output SAS data set will contain the necessary information to extract the data from the data base. The second part of the extraction can be a batch job. This job will use the SAS data set created in the first part of the process...
as input. The SAS data set will tell the procedure what to select from the data base, the SAS names and SAS format to use, and the where criteria. Another SAS data set will be created with the extracted data. This two-part process will allow large extractions or extractions that are done frequently to be done as batch jobs, thereby saving online resources and CPU time.

All the capabilities available in the full-screen application will have equivalent commands for executing the procedure in interactive line mode or batch mode. There will be commands to select the elements for extraction and assign SAS names, to assign SAS formats, to enter a where clause, and to list various information about the data base and SAS data set.

As you can see, the procedure IDMSEXT makes building a SAS data set from an IDMS/R data base an easy process. The procedure has been designed to make the process easy for the most novice and the most experienced IDMS/R and SAS users. In the future we hope to develop other procedures that will allow you to update IDMS/R data bases from SAS data sets.

Because the procedures are still under development, your suggestions and comments would be appreciated and will be taken into consideration.

SAS and SASware Ballot are registered trademarks of SAS Institute Inc., Cary, NC, USA. IDMS/R, 1DD, ASF, LSF, and DDS are trademarks of Cullinet Software, Inc.