ABSTRACT
A marriage of SAS/AF and SQL to provide seamless ad hoc access to a collection of databases for the computer illiterate.

This menu driven system developed uses SAS/AF and SQL to enable the computer illiterate user to access a collection of DB2 Tables and SAS datasets. Data from various datasets are joined, without user intervention, on the basis of control information contained in SAS datasets.

The user selects the desired query dimensions and query variables. The system submits the sequence of SAS steps that are necessary to gather the requested data and present it in a report or graphic form.

The user can sort the data, customize the output format, develop new variables as expressions using extracted data, and summarize the extracted data through additional menu selections.

The paper describes the use of SAS/AF and SQL techniques that were used to develop this system under SAS 6.06. Subsequent improvements to the system, that were made possible under SAS 6.07, are discussed. Specific reference is made to the use of the SQL Passthrough feature provided in SAS 6.07.

APPLICATION REQUIREMENTS
The following is a list of the system's requirements:
- organize a collection of data in DB2 tables and SAS datasets from available data resources to respond to identified and foreseeable business information requirements
- provide a set of standard, on-demand reporting functions to satisfy identified, common, information requirements
- provide an ad hoc method of access for data selection, computation control, presentation format design, and hard copy output control
- provide results interactively, on-line, in a text or graphic mode, with the ability to obtain local or remote hard copy
- offer an iterative process for refining or broadening the parameters of analysis to obtain a desired result
- present the system in a way that is self documenting for the occasional user, and provides efficient access to results for the experienced analyst
- provide a framework in which an experienced user can create and retain data and procedures that are unique to the business requirements.

INTRODUCTION
The paper describes the SAS/AF and SQL techniques that were used to develop this system under SAS 6.06. Subsequent improvements to the system, that were made possible under SAS 6.07, are discussed. Specific reference is made to the use of the SQL Passthrough feature provided in SAS 6.07.

This menu driven system, developed using SAS/AF and SQL, enables a computer illiterate user to access a collection of DB2 Tables and SAS datasets. Data from various datasets are joined, without user intervention, on the basis of control information contained in SAS datasets. The control information is extracted primarily via PROC Contents and from DB2 System tables.

The user selects the desired query dimensions from menus presenting identified dimensional variables and available values. The user selects the variables from menus presenting lists of available variables. The system submits a sequence of SAS steps that are needed to gather the requested data and present it in a report or graphic form.

The user can sort the extracted data, customize the output format, develop new variables as expressions using extracted data, and summarize the extracted data through additional menu selections.

GENERAL DEVELOPMENT APPROACH
SAS/AF provides the overall structure for the system. High-level navigation is provided via Block menus which are constructed dynamically from information in a security list. The security list correlates the users with the functional components of the system. Block menus pass control to AF Programs controlling the flow of each component within the system. CBT screens, corresponding to each Program screen, provide the primary HELP capability and explain screen usage.

The ad hoc reporting process is accomplished in three basic steps:
1) selecting the dimensions or scope of the desired result
2) selecting the variables or content
3) specifying the presentation format of the output.

The user specifies the parameters of a query request via a sequence of Program screens which present appropriate selection lists.

Extended tables present the selection lists. HELP information for specific items in the selection lists is integrated into the lists. Help is made available via the HELP key when a list is displayed, through simple COMMAND capture logic in the program controlling the extended table.
Standard reporting functions reflect the same three steps; scope, content, and format. For standard reports, the content and format are preprogrammed and only the scope is requested from the user. It is entered through the same dimension screen used in the ad hoc process. Standard report functions are provided where constant use requires the efficiency of a preprogrammed procedure or where the complexity of data gathering is beyond the grasp of the casual user. Each standard report process prepares a dataset, containing the results of its data gathering step, from which results are reported. These datasets become part of the database available to the user through ad hoc analysis.

The database is a collection of DB2 tables and SAS datasets which is identified to the ad hoc system through lists of the datasets and their columns. Information in the lists is extracted from the Corporate Information Asset Directory. Some additional parameters, required to control the joining of datasets, are entered into the lists. This control information is crucial to the ability of the user to view a complex collection of datasets as a seamless whole, without needing to understand the technical process used to join data in multiple datasets.

Once the dimensions, variables, and presentation format of a desired result are specified, the user submits the request either on-line or in batch mode. The request consists of a sequence of steps which are built and submitted interactively in the on line mode, or are built and bundled in appropriate Job Control Language statements for batch submission.

PROC SQL is the vehicle used for gathering and joining the selected variables into a dataset for subsequent processing. PROC PRINT is used under SAS 6.06 to present results. PROC REPORT, available under SAS 6.07, is expected to provide a more flexible method for dynamic report generation. PROC PLOT is used to provide graphic presentation of results. PROC SORT and various DATA steps are used to manipulate the data for presentation.

When the results of a query request are displayed, the user can produce a hard copy, save the procedure and the resulting data for future use, and send it via remote printing, or email services to other users. The user can also return to the dimension, variable, or presentation selection menus to modify any portion of the request for resubmission. This iterative process enables the user to create a report and run it for various dimension selections or to create a complex analysis by building up a series of relatively simple and manageable steps.

**DIMENSION SELECTION PROCESS**

The DIMENSION SELECTION screen, Fig 1, is common to ad hoc and standard report processing requests. For ad hoc requests, the screen accepts entry of two report title lines (standard reports have preset titles). For the Sales and Marketing organization in a pharmaceutical company, the primary business dimensions are timeframe, product, customer, and sales organization. These are the dimension specifications accepted for entry in the dimension selection process. These dimension specifications represent the basic selection criteria imposed on the data-gathering process. Other criteria, further qualifying the process and relating to variables other than dimensional variables, are entered in the Presentation step.

Dates, indicating the start and end of the period to be analyzed, may be entered directly into available fields or selected through available menus providing lists of months, quarters, fiscal, or calendar years, and specific periods such as current date, most recent full quarter or month, etc. Date Menus are activated via the "SELECT" ACTION button on the dimension screen Examples of each level, the selection method, Fig 2, and the resulting detail selection list, Fig 3, are presented for the Timeframe dimension.

![Fig 1. DATE SELECTION SCREEN](image1)

![Fig 2. DATE SELECTION METHOD SCREEN](image2)
Product, Account, and Sales Organization selections are also made through extended tables, in a procedure similar to that for Timeframe. Available selection methods are presented from lists maintained in a SAS dataset. Lists of specific Products, Accounts, or Sales Organizations from DB2 tables are conditioned and manipulated on the basis of the method selected so that the user can select dimensions by whatever manner the entities are known within the database.

VARIABLE SELECTION PROCESS
The VARIABLE SELECTION screen, Fig 4, is unique to the ad hoc analysis process, and replaces the list of variables included, explicitly or implicitly, in the data-gathering step(s) of a report process. Through this screen and the additional screens presented from it, the user selects those variables that will be included in the report, or are required, as qualifiers, components of expressions, or otherwise, in generating the report.

Variables are selected from lists presented in an Extended Table. The list of available variables, (those variables identified from among all variables existing in the corporate data repository as being of interest to the Sales & Marketing business function) is presented in dataset groupings or in categorical groupings. These groupings are used to subset the list into manageable portions. The dataset groupings are convenient for a user who has some familiarity with the organization of data in the repository. The categorical grouping is based on an arbitrary assignment of the variables into meaningful business categories such as price information, account information, product information, sales amounts, and quantities, etc, and are more meaningful to the novice user.

HELP information is available for each variable shown in a selection list. The information is based on descriptive information stored in the repository and acquired via Proc SQL initiated from within the HELP screen. Some data centers are more or less diligent in keeping this kind of data repository information current and complete. Its accuracy depends primarily on whether or not it is actually used. It is, however, the logical place to acquire HELP information at the dataset or variable level and it would be counterproductive to provide a parallel source.

PRESENTATION SPECIFICATION
The PRESENTATION SPECIFICATION screen, Fig 5, presents five options:
1) Selection Criteria
2) Sort/Break Specification
3) Presentation Format
4) Expressions
5) Summarization
Each of these options presents an aspect of customizing the user's request beyond the simple Dimension and Variable selections already made.
The SELECTION CRITERIA screen, Fig 6, displays existing selection criteria, already imposed on selected variables. The user has the ability to construct, through a hierarchy of menus, additional selection criteria needed to qualify the desired result. Various selection methods, conditional on data type, are presented. Selection of a selection method, such as 'Between limits', causes the presentation of a sequence of screens to identify the value of upper and lower limits. The result of this process is the generation of a valid SAS conditional expression for the WHERE clause of the Proc SQL step which is used later to gather data.

The SORT / BREAK SPECIFICATION screen, Fig 7, displays existing sort and break specifications, if any, and enables the user to modify them. Sort specifications generate a Proc SORT step which is submitted prior to printing the report. Break specifications generate BY, PAGEBY, and SUMBY clauses in the Proc PRINT which is used to generate the report for display or printing.

The PRESENTATION FORMAT SPECIFICATION screen, Fig 8, displays existing specification of the report format such as the order of columns across the page, whether or not a column is to be totaled, the column format, and column label. These specifications also affect the Proc PRINT step.

The EXPRESSION SPECIFICATION screen, Fig 9, displays existing expressions and enables modification and additions. Expressions create new variables in each row of the gathered dataset based on expressions constructed via menus presented from this screen. Generalized expressions can be created through a controlled sequence of menus. Specific expressions such as 'X as a percent of Y' and 'Group the values of a variable into Ranges' are also constructed from menus. The variables created in this way are added to the gathered dataset, after the completion of the initial Proc SQL step, in a DATA step. This method was adopted because of the greater richness of functionality currently in the SAS DATA step as compared with that in SQL.

The FUNCTION SPECIFICATION screen, Fig 10, displays existing function specifications and enables modification and additions. Functions summarize the gathered dataset, grouping those variables for which a FUNCTION has been specified, by the specified SORT variables. The process creates a summary dataset which, after enabling the user to...
save the gathered detail, replaces the detail as the working dataset. Subsequent reporting and further query iterations are performed on the summarized dataset. Functions, including SUM, AVERAGE, COUNT, etc, are imposed through menus on selected variables.

![Fig 10. SUMMARIZATION SCREEN](image1)

**REPORT PRODUCTION PROCESS**

When the user has completed selecting dimensions and variables and specified the desired presentation characteristics of the request, a screen, Fig 11, is displayed, showing estimates of the time and size requirements of the request together with a recommendation to submit the query in on-line or batch mode, or to review the request before submission. Estimates of size and time are based on the known sizes of tables and datasets involved in the request, how they are joined, and on the imposed selection criteria.

![Fig 11. REPORT MODE SELECTION](image2)

The REVIEW option presents a screen enabling the user to select the point of return into the query specification process, at DIMENSION SELECTION, VARIABLE SELECTION, or PRESENTATION SPECIFICATION. This screen is also presented after viewing of a resulting report, and enables the query and resulting data to be saved for future use.

The ON-LINE option initiates the interactive construction and submission of a sequence of SAS procedures (discussed briefly above and more fully in this section) intended to accomplish the user's desired result. The result is represented by the gathered (and summarized) dataset and the user's captured specifications in two small SAS datasets containing dimension and variable information. The report, Fig 12, and/or graph which can be viewed, printed, or conveyed to other users via e-mail, is created from these three datasets. They exist in WORK until saved and they are copied into a permanent dataset belonging exclusively to the active userid.

![Fig 12. REPORT PRESENTATION](image3)

The BATCH option constructs the same sequence of SAS procedures, but surrounds them with appropriate Job Control Language and submits the lot to the MVS initiators. JCL causes the results to be placed in a dataset, available inside the Analysis System, to the submitting userid. SAS SHARE facilitates the user's ability to continue interactively on other tasks until the Batch Job is completed.
In the current ad hoc analysis system, six steps are used to accomplish all of the actions allowed in a query. Each step is constructed from the information gathered in the Dimension Selection, Variable Selection, and Presentation Specification processes. Each step is written to a unique FILE with PUT statements involving fixed text, variables saved in dimension and variable datasets, and the value of established macro variables.

In ONLINE mode, each FILE is made the object of a %include within a SUBMIT block in the controlling AF program. In BATCH mode, each individual FILE is concatenated to the others within JCL to run a Batch Job.

1) Dimension selections are used to screen datasets containing the universe of available time frame, product, account, and sales organization entities reflected in the data repository. The screening creates appropriate subsets of selected dimensional values to be used later in conditioning the selection of gathered data (Fig 13).

```sql
LET START = 01JAN91;
LET END = 30JUN91;
PROC SQL; CREATE TABLE PRODSLCT AS
SELECT " FROM BCANDATA.PRODLKUP
WHERE DDDPRDNO = '1101010'
OR DDDPRDNO = '1102010';
QUIT;
PROC SQL; CREATE TABLE PRODSLCT AS
SELECT DISTINCT " FROM PRODSLCT;
QUIT;
PROC SQL; CREATE TABLE ACCTSLCT AS
SELECT DISTINCT A." FROM BCANDATA.ACCOUNTS A
, BCANLIBP.PGXA004 B
WHERE A.ACCOUNT = B.BUYGRPMEM AND A.ACCCTCAT = 'HI'
AND B.BGEFFDT <= '1102010'
AND B.BGEXPDT >= '01JAN91'
AND ( B.BUYGRP1D = '22222'
AND PGXA004.BUYGRPID = FORO00.BUYGRPID
AND PGXAO04.BUYPREM = ACCOUNTS.ACCOUNT
AND ACCOUNTS.ACCOUNT = DDDSALES.BIDACCT
AND PRODLKUP.dddprdno = DDDSALES.PRODGRP
QUIT;
```

Fig 13. DIMENSION PROCESSING STEP

2) Variable selections are used to construct a Proc SQL step which SELECTs the indicated columns or variables from the DB2 tables or SAS datasets in which they reside. WHERE clauses arise from three separate sources. First, those datasets implied by the list of selected variables are joined on common columns in the datasets. Second, dimensional variables such as SALES DATE, PRODUCT NUMBER, ACCOUNT NUMBER, and SALES TERRITORY, are conditioned against the dimension subsets created in step 1. Third, the explicitly stated selection criteria, specified by the user, are imposed (Fig 14).

```sql
PROC SQL;
CREATE TABLE WORK.GBCAO00 AS
SELECT ACCOUNTS.ACCCTFRM FORO00.BUYGRPM
, TIMFRAME.QUARTER
, DDDSALER.BIDACCT AS ACCOUNT DDDSALER.DDDSAL
, DDDSALER.DDDSALDT
, PGXAO04.BGEFFDT
, PGXAO04.BUYGRPID
, PGXAO04.BUYPREM
, DDDSALER.PRODGRP AS DDDPRDNO
, PRODLKUP.PRICEGRP
FROM BCANDATA.TIMFRAME
, BCANDATA.PRODLKUP
, BCANLIBP.PGXA004
, BCANDATA.ACCOUNTS
, SCANLISP.PGXAO04
, BCANDATA.DDDSALER
WHERE ACCOUNTS.ACCOUNT IN (SELECT ACCOUNT FROM ACCTSLCT)
AND PRODLKUP.PRICEGRP IN (SELECT DDDPRDNO FROM PRODSLCT)
AND DDDSALER.DDDSALDT BETWEEN 'START'D AND 'END'D
AND DDDSALER.DDDSALDT BETWEEN STARTDT AND ENDDT
AND PGXAO04.BGEFFDT = 'START'D
AND PGXAO04.BGEXPDT = 'END'D
AND PGXAO04.BUYGRPID = '22222'
AND PGXAO04.BUYPREM = ACCOUNTS.ACCOUNT
AND ACCOUNTS.ACCOUNT = DDDSALER.BIDACCT
AND PRODLKUP.dddprdno = DDDSALER.PRODGRP
QUIT;
```

Fig 14. DATA GATHERING STEP

3) Variables created through EXPRESSIONS are included into the gathered dataset through a DATA step generated from the expressions specified by the user and saved in the variable dataset (Fig 15).

```sql
DATA GBCAO00;
SET GBCAO00;
GPO =TRIM(BUYGRPID));'||TRIM(BUYGRPID) ;
RUN;
```

Fig 15. EXPRESSION CREATION STEP
4) Summarization is imposed on the gathered detail dataset through a Proc SQL step generated from those FUNCTIONS specified by the user and saved in the variable dataset. When functions are present in a request, a screen is displayed between steps 3 and 4, enabling the user to save the gathered detail dataset, before continuing to the summarization step (Fig 16).

```sas
PROC SQL;
CREATE TABLE GBCAOOO AS
SELECT GPO, BUYGRPNM, QUARTER, ACCOUNT, BUYORPID, DDDPRDRO, PRICEGRP
SUM(DDDSALES) AS DDDQTRSL
FROM G8CAOOO
GROUP BY GPO, BUYGRPNM, QUARTER, ACCOUNT, BUYORPID, DDDPRDRO, PRICEGRP;
QUIT;
```

Fig 16. SUMMARIZATION STEP

5) The gathered (and summarized) data is sorted through a Proc SORT step generated from the specification of sort keys and order saved in the variable dataset (Fig 17).

```sas
PROC SORT DATA=GBCAOOO;
BY GPO, ACCOUNT, QUARTER, BUYORPID, DDDPRDRO, PRICEGRP;
RUN;
```

Fig 17. DATA SORT STEP

6) A Proc PRINT, preceded by a Proc PRINTTO pointing to a work file and TITLE statements utilizing MACRO generated standard headings, is generated from presentation characteristics in the variable dataset (Fig 18).

```sas
PROC PRINTTO PRINT=GBCAOOO NEW; RUN;
OPTIONS NOCENTER BODATE LINESIZE=72 PAGBSIZE=19 PAGENO=1;
TITLE1 "&TITLEA";
TITLE2 "&TITLEB";
TITLE3 "&TITLEC";
PROC PRINT DATA=GBCAOOO WORK.GBCAOOO NOBS LABEL SPLIT="";
VAR BODATE DDDQTRSL
PRICEGRP
DDDQTRSL
LABEL ACCOUNT = 'MEMBER'
GPO = 'GPO'
QUARTER = 'QTR'
DDDPRDNO = 'DDD PRODNO'
PRICEGRP = 'PRODUCT GROUP'
DDDQTRSL = 'QUARTERLY DDD SALES'
FORMAT DDDQTRSL COMMA13.2
BY GPO ACCOUNT QUARTER;
SUM BY QUARTER
PAGE BY GPO;
SUM DDDQTRSL
RUN;
PROC PRINTTO PRINT=PRINT; RUN;
```

Fig 18. REPORT PRESENTATION STEP

SAS 6.07 PROCESS IMPROVEMENTS

The most significant improvements to this system, available through the use of SAS 6.07, are to the components using Proc SQL and to Step 6 of the Report Production Process where Proc REPORT will replace Proc PRINT to format the report.

At presentation of this paper, our data center has not yet installed a 6.07 version of SAS for developmental usage.

Expectations for the indicated improvements arise from observation of the product in a test environment, from preliminary copies of manuals, and from conversation with various SAS personnel.
CONCLUSION

The use of SAS/AF facilitated the development of this CUA compatible application. The application is sufficiently generalized to be applicable to a wide range of business functions, yet customized to a specific business area through data driven menu selection lists. HELP screens provide on-line user assistance as well as system self documentation.

PROC SQL, available in SAS 6.06, provided the ability to access data in DB2 tables directly, and to join it with data in SAS datasets without intermediate data translation or transfer. Additional features, available in SAS 6.07, provide a significant improvement in efficiency when large DB2 tables are processed.

PROC REPORT, available in SAS 6.07, provides a significant enhancement to the flexibility of format as compared to the PROC PRINT feature used under SAS 6.06.

While SAS 6.06 provided breakthroughs in the ability to develop truly user-friendly applications, SAS 6.07 has greatly improved their efficiency and flexibility.

ACKNOWLEDGMENTS

The author acknowledges Anne Soeder, Glaxo, Inc., who cooperated in the design of the system and developed several major components.

During development of this application, we have made use of the wealth of other SAS user's experience found in SUGI Conference Proceedings from prior years, and of the expertise of personnel staffing the SAS Technical Hotline.

We particularly acknowledge the efforts of the following individuals from SAS Institute, John Boling and Yvonne Shelby in reviewing the overall design of this application, and of Paul Kent and Lewis Church in reviewing the DB2 interface.