A Data Access Challenge:
Build a SAS® Frame Application for Access to an Informix Data Warehouse

John E. Bentley, First Union National Bank, Charlotte, NC

ABSTRACT

First Union National Bank has built a two-terrabyte data warehouse containing current and historic data at varying levels of granularity. The driving force behind the project was the need for improved statistical modeling to support the Bank's marketing efforts as it moves into the 21st century. The marketing data analysts were already proficient with SAS® statistical procedures but needed a user-friendly tool to extract and transform data from the warehouse. Due in part to the data warehouse architecture, the final application design requirements were complicated. This paper presents an overview of the project and of how SAS System software was used to meet the design challenges.

INTRODUCTION AND BACKGROUND

In 1984, First Union National Bank set as its strategic goal to become a major competitive force in U.S. banking. Since then, guided by a strategic plan that included a component calling for a disciplined program of investments in technology, First Union has become the nation's sixth-largest bank, growing its assets from $7 billion to almost $150 billion.

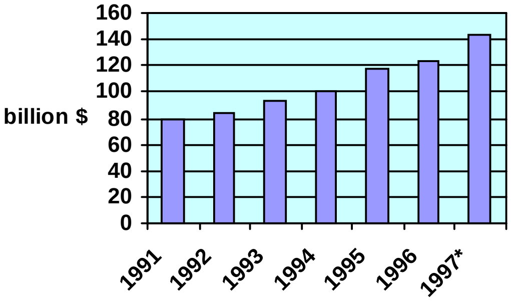
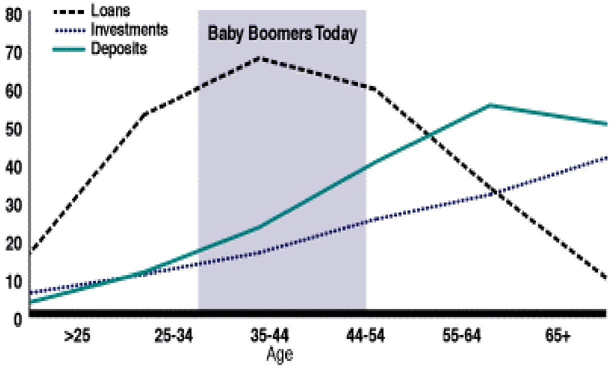


Chart 1. Year End Earning Assets of First Union National Bank
* Through 3d quarter. Does not include Signet or CoreStates mergers.

In 1996, First Union updated its strategic plan to take the bank into the 21st century. A major initiative under the new plan is the "Consumer Bank of the Future", also know as the Future Bank Initiative. This initiative takes advantage of major demographic shifts now underway and the resulting changes in customer preferences relating to both banking products and channels of delivery. For example, over 77 million "baby boomers" are currently approaching age 50 and studies of financial holdings by age show that aging consumers rely less on loans and traditional deposits but increase their demand for investment products. Graph 1 shows one estimate of how people's needs for bank-provided financial products and services change as they age. The scale on the left is the percentage of persons estimated to use the product/service.



Graph 1. The Changing Needs of the Baby Boom Generation

The Consumer Bank of the Future initiative will allow First Union to better serve its customers by providing a comprehensive range of financial services that help them prepare for the future. As should be expected, technology plays an essential role in the implementing the initiative.

While most of First Union's competitors are making investments in technology to achieve expense efficiencies, First Union's has already achieved a technological edge to become one of the most efficient high-transaction banks in the nation. First Union is now leveraging the advantages of its integrated automation systems to generate real, measurable increases in revenue and customer satisfaction.

First Union's solid technology foundation—which of course includes SAS software—allows rapid adaptation to changing customer demands and market forces and enables more customization of offerings to customers. A single, integrated, company-wide view of individual customer relationships is crucial for taking advantage of the emerging technologies of data mining, OLAP and decision support systems, executive information systems, and enterprise-wide reporting systems. SAS System software provides a critical toolset for the bank's efforts to implement a single company-wide view of its data via data warehousing.

THE CORPORATE MARKETING DATA WAREHOUSE

In the summer 1996, First Union partnered with an enterprise systems development company to design, build, and implement a Corporate Marketing Data Warehouse (CMDW) to hold data from nineteen legacy systems. Design and research into the legacy systems was completed in early 1997 and CMDW development was essentially completed in the 4th quarter 1997. Due to First Union's intensive merger and acquisitions activity in the second half of 1997, however, full roll-out was delayed until the 1st quarter 1998 so that the hardware platform be doubled in size.

The development environment is now (December 1997) being used for training and the production environment should be released to the users in the first quarter 1998. When released, the CMDW will initially contain about two terrabytes of historic and current customer data—multiple records for about 100 million accounts, individuals, and households along with transaction activity for certain entities.

The initial release of the CMDW provides a customer-focused database that supports data analysis and data mining for marketing purposes. In this effort, the Information Technology Group has been successful in providing a totally new data infrastructure and data extraction tools that will substantially improve First Union's ability to:

- Quickly develop and evaluate new market strategies;
- React to marketplace changes in a timely and effective manner; and
- Measure the effectiveness of actions taken.

In addition to supporting marketing analyses such as identifying and targeting likely buyers of specific products and services, the CMDW was architected to allow the creation of subject-specific data marts for OLAP and decision support purposes.

The combination of computer software and hardware chosen for the general system architecture was "challenging" when it was selected:

- Informix XPS® version 8.1 RDBMS software
- A 58-node IBM RS/6000 SP® computer (increased to 104 nodes in January 1998);
- Parallelization software;
- Hard drive storage providing four terrabytes of mirrored disk space;
- Ethernet LAN and TCP/IP communications software;
- Firewall devices and technology for security;
- AIX version 4.1 UNIX operating system; and
- SAS software running under UNIX, Windows 95, and Windows NT.

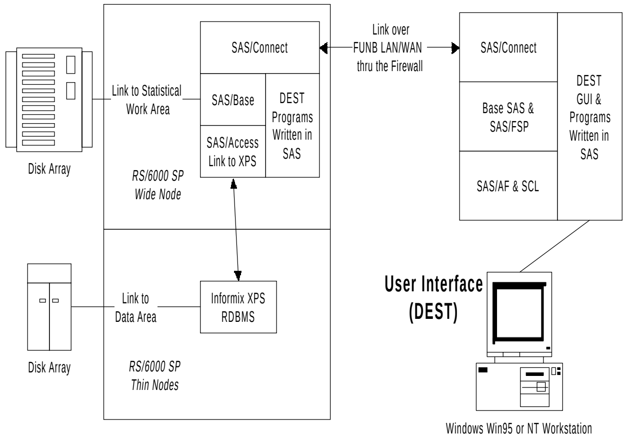


Figure 1. The High-Level Architecture of the CMDW

The warehouse has over two terrabytes of disk space usable for data and contains over 130 tables and 2,200 fields in a normalized physical data model. Records can be at the household, individual, account, or transaction level. Each record has effective and expiration date fields for point-in-time selection and Key fields that allow aggregation (if appropriate) and merging with records at the same, higher, or lower level.

THE CMDW FRONT-END: THE DATA EXTRACT AND SAMPLING TOOL

As mentioned earlier, the driving force behind the CMDW was the need for improved statistical modeling to support First Union's marketing efforts. The demand for the services of First Union's Knowledge Based Marketing department (KBM) has dramatically increased since the mid-1990s. With the Bank's strategic transition to a customer-centric marketing vision, the demand for KBM's expertise will continue to increase. Furthermore, as corporate information consumers become more sophisticated due to improved information availability, the questions posed to KBM will become increasingly complex.

One of KBM's primary tasks is to build statistical models that, for example, accurately predict the propensity of an existing bank customer to purchase a new bank product. Using the coefficients generated by the model, an algorithm assigns a numeric score to each customer. In this example, we'll assume that the higher the score the more likely the customer is to purchase the new product. By restricting the product's marketing effort to customers with the highest scores, available resources can be most effectively used and the highest success rate guaranteed. In this paper, the process of building a predictive model is referred to as "modeling" and assigning those scores is referred to as "scoring".

In the past, KBM's data analysts, modelers, and statisticians based their modeling on a vendor-provided flat-file that was updated monthly and contained about 450 input variables. The vendor also provided a proprietary data extraction tool for the data set but the data still left much to be desired in terms of timeliness, coverage, and ease-of-use. The time to extract and transform data, create and validate a model, and score the customer database was measured in weeks.

The CMDW was initially designed to provide a central repository for data from legacy systems that KBM would find useful. The KBM staff was already proficient with SAS's statistical procedures but clearly needed a tool to retrieve data from the warehouse if the modeling process was to become bogged down in data extraction. The Data Extract and Sampling Tool (DEST) was envisioned as the primary tool for selecting and extracting data from the warehouse, thereby reducing the overall time that is needed for statistical modeling. By making the data more readily accessible, the data analysts and modelers could focus their energy and expertise on developing models and identifying market opportunities.

The final design requirements of the DEST, however, were complicated:

- SAS can access Informix databases only with SQL Pass-Through;
- Users would not write SQL code;
- The volume of data in the warehouse necessitates that samples be extracted for exploratory data analysis and model development;
- The data warehouse contains mixed levels of granularity, which mandates multiple SQL statements and multiple table joins to aggregate and merge data prior to extraction or sampling ;
- "Flattening" has to be performed, in which multiple rows are aggregated and transposed into a single variable.
- Samples and statistical models must be written back into the warehouse for reuse;
- Based on the models, scores must be assigned to records in the warehouse and then written back into the warehouse for later use;
- Production scoring jobs could be run only overnight; and
- The application must be scaleable and portable to other types of RDBMS and hardware platforms.

THE SAS SOFTWARE SOLUTION

To meet the design challenges, the decision was to use SAS software to build an application with a graphical user interface (GUI) that incorporated automated SQL code generation. Because SAS's Enterprise Data Miner was not available in the late spring 1997, it couldn't be considered as a possible solution.

Specific functionality was identified that the application had to fulfill:

- Data extraction from both single and multiple tables according to user-specified selection and subsetting criteria;
- Data aggregation and reduction including sum, average, minimum, maximum, count distinct, and absolute value;
- Flattening (transposing) by product, time, and time and product;
- Sample Universe data extraction;
- Sampling, to include stratification;
- Sample registration in the data warehouse;
- Modeling data extraction using one or more samples as filtering criteria;
- Model registration in the data warehouse; and
- Production scoring of the data warehouse.

In addition to Base SAS software,

- SAS/CONNECT® allows TCP/IP connectivity between the user desktop/workstation and the IBM SP. The GUI on the desktop invokes SAS programs and Informix processes that execute on the SP in either interactive or batch mode;
- The SAS/ACCESS® SQL Pass-Through Facility provides read/write access to the Informix XPS database;
- SAS/AF FRAMES was used to build the GUI interface. The DEST is an extension of the normal SAS environment;
- Screen Control Language (SCL) controls the DEST application, the interaction between the user and the application, communication between application windows, code execution, etc.; and
- SAS/FSP® accepts user data selection criteria and specifications and saves them flatfiles and SAS-format data sets.

The DEST is a distributed client/server application that takes advantage of SAS's remote compute and data transfer service capabilities. All data extraction, manipulation, and analysis programs used or generated by the DEST are transparently submitted to execute on the IBM SP. All data sets—work as well as permanent—are written there as well. Master copies of the metadata data sets are stored on the SP. Each time the DEST starts, a startup routine compares the creation dates of the local versions of the metadata with those of the masters and a PROC DOWNLOAD routine updates the local versions if necessary. In effect, SAS's MultiVendor Architecture design and connectivity capabilities allows the IBM SP to act as both an application server and a data server for the desktop.

In addition to the SAS software installed on the IBM SP, the DEST requires that several dozen SAS Macro programs and scripts be located in a common subdirectory there. Other than the SAS software on the desktop, only the Frame application itself in a SAS catalog about 2.5 megabytes in size, the SAS/CONNECT script, and a copy of the metadata are stored locally. Regarding disk space on the SP, almost 100 gigabytes of "playground" is reserved for the work data sets that SAS may generate, and each DEST user has a personal directory of several gigabytes.

THE DEST WORKFLOW

The DEST runs on top of SAS and is invoked in a manner that allows the user to work switch between it and the normal SAS environment. In the same SAS session, for example, a user can extract a Sampling Universe data set from the data warehouse with the DEST, switch to the Program Manager to run an exploratory data analysis routine to identify sample segment breakpoints, and then go back into the DEST and draw a segmented sample. The flowchart in Figure 2 presents the DEST workflow and shows the user's ability to move between the DEST and the normal SAS environment.

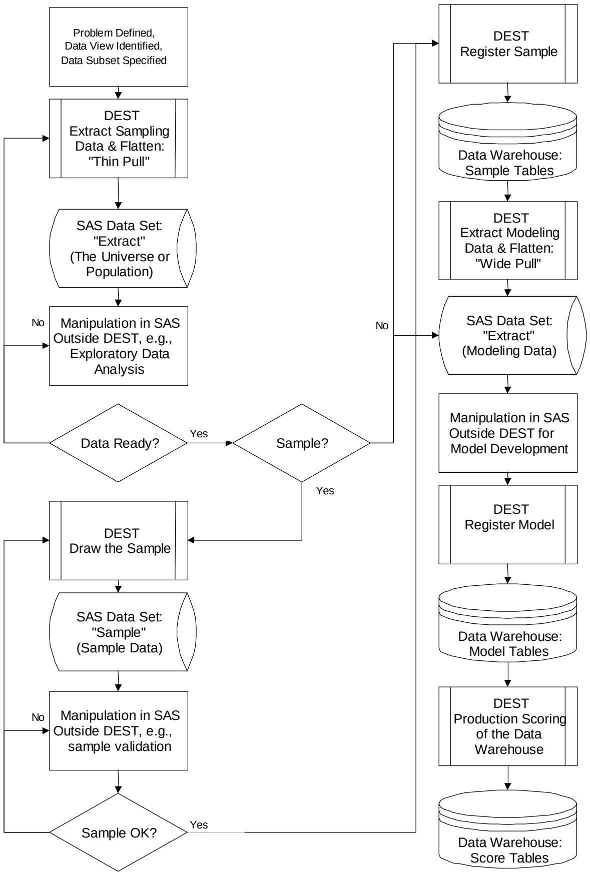

Problem Defined, Data View Identified, Data Subset Specified
DEST Extract Sampling Data & Flatten: "Thin Pull"
SAS Data Set: "Extract" (The Universe or Population)
No
Manipulation in SAS Outside DEST, e.g., Exploratory Data Analysis
Data Ready?
Yes
Sample?
Yes
No
DEST Draw the Sample
SAS Data Set: "Sample" (Sample Data)
No
Manipulation in SAS Outside DEST, e.g., sample validation
Sample OK?
Yes
DEST Register Sample
Data Warehouse: Sample Tables
DEST Extract Modeling Data & Flatten: "Wide Pull"
SAS Data Set: "Extract" (Modeling Data)
Manipulation in SAS Outside DEST for Model Development
DEST Register Model
Data Warehouse: Model Tables
DEST Production Scoring of the Data Warehouse
Data Warehouse: Score Tables

Figure 2. The Data Extract and Sampling Tool Work Flow

The process begins with a well-defined business problem. From this, the appropriate data presentation-level or "view" for the solution is identified and the customer sub-group to be sampled is defined. The presentation-level (from lowest to highest) can be the transaction, account, individual, or household level. For each presentation level, the DEST metadata defines whether a variable is at, above, or below the presentation level. Each table is associated with a "natural" presentation level and the DEST metadata contains Key fields for joining tables across presentation levels. (More on metadata later.)

The user specifies filter variables to define the Sampling Universe and uses flattening variables where aggregation of bank products or transformation over time is needed. For example, a data analyst might want to work with individuals in households with both a mortgage and a car lease financed by First Union and having at least one interest-bearing checking account balance averaging over $1,000 each month for the past three months.

Based on the criteria for including records in the Universe, the DEST can aggregate data to the next highest level or attach higher level data to records at a lower level. For example, if a household has multiple checking accounts, the end-of-month balances can be aggregated and then that sum attached to the household record. It's also possible for variable manipulation/transformation to be accomplished outside the DEST and the resulting dataset then brought back into the DEST. It may be, for example, that a ratio must be calculated to use as the basis for sample stratification.

The DEST provides basic information on the Universe, such as the number of observations. The user-specified sample size can be any number up to 50 percent of the Universe. The Sampling Data Extract is a “narrow pull” containing only the variables needed for generating the sample—perhaps 12 or 15.

The DEST performs simple random, Nth, and stratified random sampling. After the sample is pulled and verified, the user enters descriptive information about the sample. A unique sample identification number is assigned to the sample and attached to each observation.

The descriptive information and the sample, composed of only the record identifier and the sample id, are first output to flat files and then written into the warehouse’s sample tables by using SAS’s SQL Pass-Through Facility to execute UNIX and Informix functions and commands. Figure 3 illustrates the Sampling process, and Code Segment 1 shows the macro-driven SQL code for writing to a sample table in the CMDW.

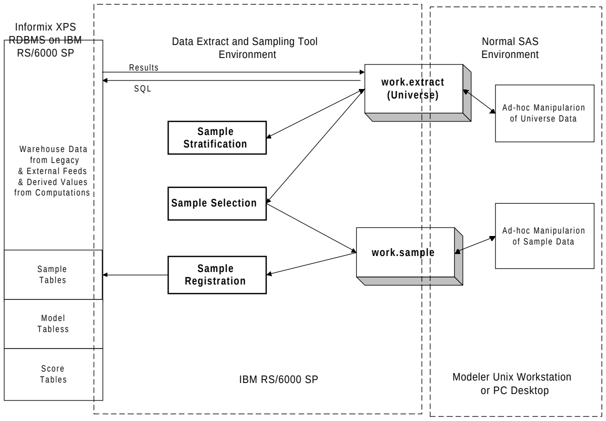

Informix XPS RDBMS on IBM RS/6000 SP
Data Extract and Sampling Tool Environment
Normal SAS Environment
Results
SQL
work.extract (Universe)
Ad-hoc Manipularion of Universe Data
Sample Stratification
Warehouse Data from Legacy & External Feeds & Derived Values from Computations
Sample Selection
work.sample
Ad-hoc Manipularion of Sample Data
Sample Tables
Sample Registration
Model Tabless
Score Tables
IBM RS/6000 SP
Modeler Unix Workstation or PC Desktop

Figure 3. The Sampling Process

```
proc sql noprint;
    connect to informix (database=&DATABASE);
    execute ( set lock mode to &locmode) by informix;
    execute ( begin work) by informix;
    execute ( lock table &SMPL_TBL in &loctabl mode) by informix;
    X echo '#!/usr/bin/ksh' > &PATH.loadscript;
    X echo RC=&SQLXRC >> &PATH.loadscript;
    X echo 'if [ $RC -eq 0 ]; then' >> &PATH.loadscript;
    X echo "rsh &TND " "" "if [ ! -p &PIPES/PIPE_&SMPL_TBL ];
                   then mknod &PIPES/PIPE_&SMPL_TBL p;
                   fi" "" >> &PATH.loadscript;
    X echo "rcp &SAMPLES/&df TND:&PIPES/PIPE_&SMPL_TBL;
                   &" >> &PATH.loadscript;
    X echo fi >> &PATH.loadscript;
    X chmod ugo+x &PATH.loadscript;
    X &PATH.loadscript 2> &PATH.err;
    X sleep 2;
    execute ( insert into &SMPL_TBL (smpl_id,sgmt_nbr,&FIELD)
                     select "&SMPL_ID", sgmt_nbr, &FIELD
                     from    ext_&SMPL_TBL) by informix;
    execute ( commit work) by informix;
    X rsh &TND rm &PIPES/PIPE_&SMPL_TBL 2>> &PATH.err;
    disconnect from informix;
  quit;
```

Code Sample 1. SQL Code for inserting sample records into the CMDW’s Sample Table

To create the data set needed for model development, the sample identification number previously loaded into the warehouse is used as the subsetting variable for a “wide pull” of perhaps hundreds of variables extracted from the CMDW. The modeling data set is either stored on the IBM SP or transferred via PROC DOWNLOAD to a UNIX server.

Model building is accomplished outside the DEST, allowing users to take advantage of SAS’s powerful data manipulation, analysis, and modeling capabilities. Using SAS/Connect, users write code locally and then use Remote Compute Services to execute it on the IBM SP or UNIX server where the data reside.

After the model has been successfully fitted, validated, and tested and coefficients generated for the independent variable(s), the user enters descriptors about the model into an DEST form and the DEST assigned a unique model id. The DEST then writes that information and the locations of SAS programs containing the data extract, manipulation and transformation code, and model coefficients code to the model tables of the warehouse, again by using SAS’s SQL Pass-Through facility.

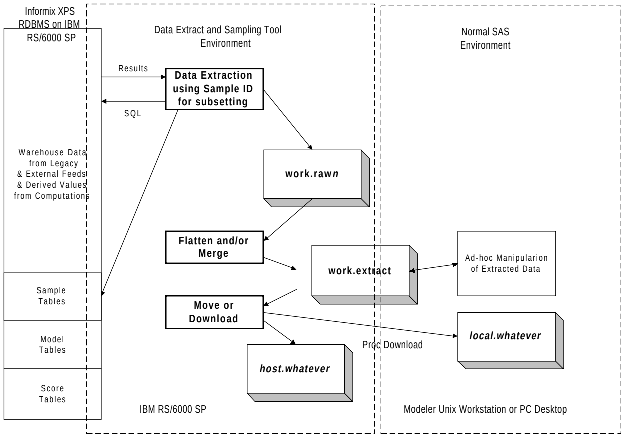

Informix XPS RDBMS on IBM RS/6000 SP
Data Extract and Sampling Tool Environment
Normal SAS Environment
Results
SQL
Data Extraction using Sample ID for subsetting
Warehouse Data from Legacy & External Feeds & Derived Values from Computations
work.rawn
Flatten and/or Merge
work.extract
Ad-hoc Manipularion of Extracted Data
Sample Tables
Move or Download
local.whatever
Proc Download
host.whatever
Model Tables
Score Tables
IBM RS/6000 SP
Modeler Unix Workstation or PC Desktop

Figure 4. The Modeling Data Extraction Process

When a model is ready to score the records in the CMDW, descriptive information about the scoring run is entered via the DEST into model tables in the data warehouse. The DEST is then used to generate and schedule an over-night production-scoring SAS job to be run via automatic scheduling software.

Scores are assigned only to records that meet the criteria for inclusion in the original Universe. If the scoring run produces multiple scores, a record is output for each score. Output records are first written to a flat file and then inserted into the score tables of CMDW, again by using SAS’s SQL Pass-Through. Descriptive information about the score is written to a table built for that purpose.

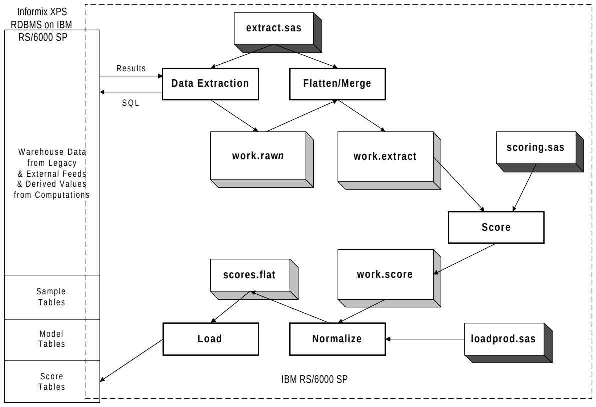


Figure 5. The Production Scoring Process

KEY POINTS OF THE DEST DESIGN

The DEST interface is organized by the particular workflow elements (discussed earlier). These elements are identified by their presence on the Main Menu Bar and its sub-menus. A few workflow functions require activities outside the DEST, such as sending email to the DEST Administrator requesting that a production-scoring job be run.

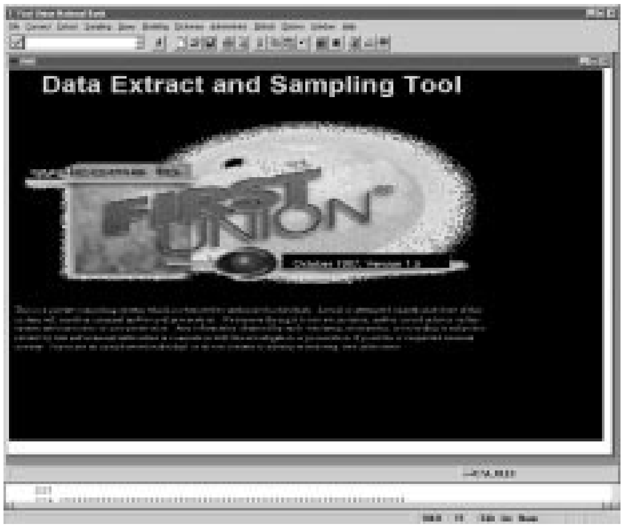


Figure 6. The DEST Splash Screen and Main Menu Bar

Metadata

For the DEST, access to the Informix data warehouse is metadata driven, and this feature contributes to its straightforward portability to other RDBMS. As long as the metadata accurately reflects the contents of the database and the relationships between the tables the DEST can easily be modified to access any data warehouse or data mart on any RDBMS platform—an Oracle-based data mart, for example, derived from the Informix data warehouse.

Information about the structure of the data warehouse, the relationships of the tables to each other, and the user-specified data extract query is held in static and dynamic SAS-format metadata data sets. Table 1 shows the metadata used by the DEST.

Data Set Name	Contents
Tables	DW_Table Entity Table_Desc Table_Type
Fields	DW_Table DW_Field Attribute Field_Desc Field_Type Selectable Filterable
Views	DW_View View_Name View_Desc View_Eff_Dte View_Exp_Dte
View_Tables	DW_View DW_Table Level_Code Prod_Flat Time_Flat
View_Joining	DW_View Child_Table Child_Field Parent_Table Parent_Field Cardinality
Extract_Select	DW_Table DW_Field Operator
Extract_Filter	DW_Table DW_Field Comparitor Criteria Left_Paren Right_Paren Connector
Extract_Table	DW_Table DW_View Prod_Parm Time_Param

Table 1. The DEST Metadata Data Sets

The static metadata describes the data warehouse to the application. It is the business metadata and is updated as needed, such as when the structure of the warehouse is changed or domains are modified. The Tables and Fields data sets hold information about the database structure, such as the business names and descriptions of the tables and fields. Information in View_Tables defines the subsets of tables that are available within a given presentation level, i.e., transaction, account, individual, or household, and the types of allowable flattening—product, time, product and time. The View_Joining metadata specifies which tables have a parent-child relationship and which field joins them.

Dynamic metadata controls presentation of user-specified tables and fields during extract query formulation. It is also critical for generating the PROC SQL and other SAS code that actually extracts the data and creates the output data set. The metadata data sets prefaced with "Extract_" are dynamic metadata tables that hold the extraction criteria provided by the user via the GUI. They are temporary SAS data sets that receive the presentation level, table and field names, Boolean comparison operators, constant values, parentheses, and connectors between selection criteria (AND or OR).

Extract Specification and Code Generation

The user only specifies the data elements to be extracted from the CMDW—the DEST generates and submits all the PROC SQL and other SAS code needed. To accomplish this, the user specifies the data warehouse source tables and the fields in each table to be retained in the output data set, the filtering (sub-setting) criteria, and any DEST-performed aggregations and transformations. As the criteria are entered, they are written to the dynamic metadata.

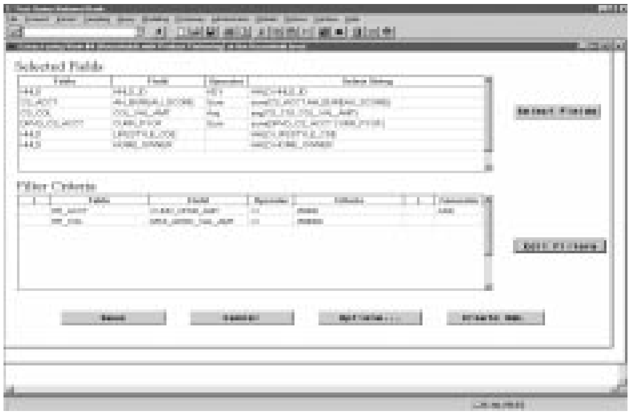
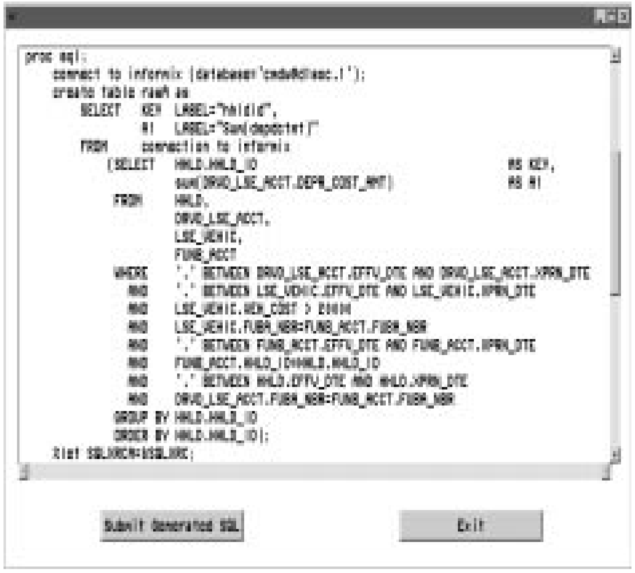


Figure 7. The Data Extract Specification Window

When finished, the extract specifications are copied from the metadata to a structured text file that is permanently saved for possible reuse. When the "Create SQL" button is selected, the text file is parsed to provide input for a macro that generates PROC SQL and SAS code to merge the SQL result sets and flatten/transform the data as necessary. The user has the opportunity to review the SQL code before it is submitted for batch execution if multiple tables are read or interactive execution when a single table is accessed.

SQL and SAS code is generated by a series of complex SAS macro programs that formulate the query and assign unique SAS names to the fields and the work and permanent SAS data sets. Code for inner and outer joins, filtering, flattening/transposing, grouping, and ordering is generated as needed.

Due to the limitations of the parent-child relationships described in the View_Joining metadata data set, one result set is returned as a temporary SAS data set for each Informix table from which a field is selected. The code generator, however, insures that all result sets are merged into a single permanent SAS data set that is written to the user's subdirectory on the SP.



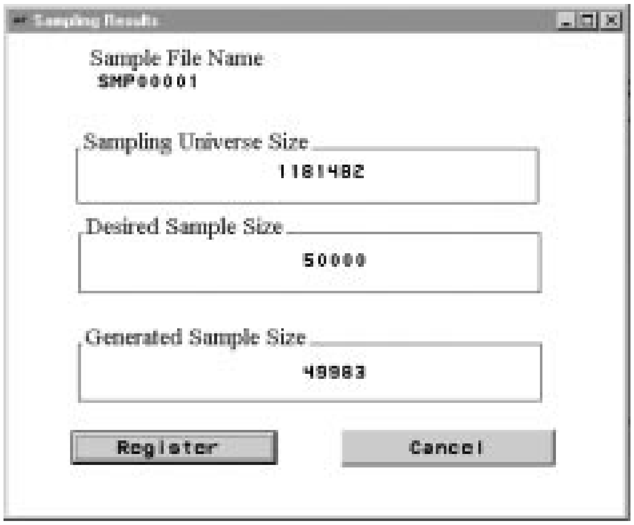
```
proc sql;
    connect to informix (database='cmdw@dlesc.1');
    create table raw1 as
       SELECT   KEY  LABEL="hhldid",
                #1   LABEL="Sum(depdcte1)"
       FROM     connection to informix
          (SELECT  HHLD.HHLD_ID                              AS KEY,
                   sum(DRVD_LSE_ACCT.DEPR_COST_AMT)          AS #1
           FROM    HHLD,
                   DRVD_LSE_ACCT,
                   LSE_VEHIC,
                   FUND_ACCT
           WHERE   '.' BETWEEN DRVD_LSE_ACCT.EFFV_DTE AND DRVD_LSE_ACCT.XPRN_DTE
             AND   '.' BETWEEN LSE_VEHIC.EFFV_DTE AND LSE_VEHIC.XPRN_DTE
             AND   LSE_VEHIC.VEH_COST > 20000
             AND   LSE_VEHIC.FUBA_NBR=FUND_ACCT.FUBA_NBR
             AND   '.' BETWEEN FUND_ACCT.EFFV_DTE AND FUND_ACCT.XPRN_DTE
             AND   FUND_ACCT.HHLD_ID=HHLD.HHLD_ID
             AND   '.' BETWEEN HHLD.EFFV_DTE AND HHLD.XPRN_DTE
             AND   DRVD_LSE_ACCT.FUBA_NBR=FUND_ACCT.FUBA_NBR
           GROUP BY HHLD.HHLD_ID
           ORDER BY HHLD.HHLD_ID);
    %let SQLXRC=&SQLXRC;
```

Figure 8. DEST-generated PROC SQL

Sampling

The DEST supports three sampling strategies: simple random, Nth, and stratified random. Simple random and Nth sampling requires that the user specify only the desired sample size, either approximate or exact. For random sampling, the sample size is divided by the Universe size to yield the selection threshold value. Then SAS's random uniform function (RANUNI/UNIFORM)) assigns a value to each record in the Universe. That value is compared to the selection threshold value and the record is output to the sample when the value is less than the threshold.

Due to the probabilistic nature of random selection, the number of records selected will likely only approximate the desired sample size. An option is available that allows the user to force an exact sized sample. Under this option, the threshold value is inflated to guarantee that a larger than desired sample is initially selected. The sample is then sorted by the randomly generated value and observations with a row number greater than the exact sample size are deleted.



Sampling Results

Sample File Name
SMP00001

Sampling Universe Size
1181482

Desired Sample Size
50000

Generated Sample Size
49983

Figure 9. Results Message from Random Sampling for an Approximate-Sized Sample

The Nth sampling strategy is computationally less intensive than random sampling. Starting with an unordered Universe, the DEST calculates the beginning row and the row interval that will provide the desired sample size and presents them to the user for confirmation. If the user chooses to override either of those values, the DEST validates the user-specified values against the maximum acceptable values and displays a warning message if the values are too large.

Stratification is supported by first sorting the Universe by a user-specified, interval or ratio-level variable. Then the data set is divided into strata according to either DEST-calculated or user-specified breakpoints. The stratification variable can be either a variable in the Universe or a calculated/transformed variable created outside the DEST.

For the DEST to calculate the strata breakpoints (where one segment ends and another begins), the user provides only the desired sample size and number of strata. With this information, the DEST calculates evenly spaced breakpoints and randomly samples each stratum to provide identically sized strata samples.

The user-defined breakpoint option requires that the user identify the breakpoints and the sample size desired from each stratum. The DEST then randomly samples the data set within the strata to provide the desired strata-specific samples.

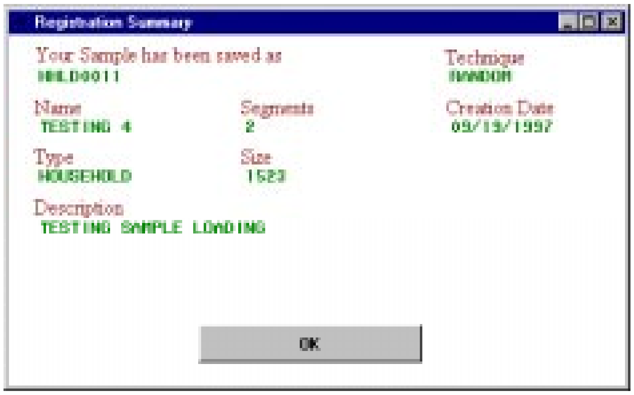
Sample and Model Registration

As noted in the section titled "DEST Workflow" and seen in Figure 2, a key feature of the DEST is its ability to register and load samples and models into the CMDW. By registering samples and models, they are saved for reuse and made available to all users.

Registration is accomplished under the Administrator option of the Main Menu Bar. In the interests of manageability and security, administrative functions such as registration (and sample and model deletion) are restricted to a few persons with DEST Administrator rights to the CMDW. Both samples and models are registered in essentially the same manner.

Prior to registration, the user completes an DEST-based form that provides important descriptive information about a newly drawn sample or completed model. The user then sends an email to the Administrator advising that a sample, for example, is ready for registration. The Administrator first reviews the descriptive information for completeness. The next step is to have the DEST generate a SAS batch job composed of a series macro-driven PROC SQLs that use the X command to execute Informix commands to insert the records into the appropriate tables of the warehouse. A example of this code is in Code Sample 1.

A fact table holds the descriptive information about the sample and there are separate tables for household, individual, and account samples. An DEST algorithm assigns a new sample id that incorporates the presentation-level and only this id, the segment number, and the record id are written to the sample table. By limiting the amount of information in the tables they are kept comparatively small even though the samples may be large.


Registration Summary

Your Sample has been saved as
HHLD0011

Technique
RANDOM

Name
TESTING 4

Segments
2

Creation Date
09/19/1997

Type
HOUSEHOLD

Size
1523

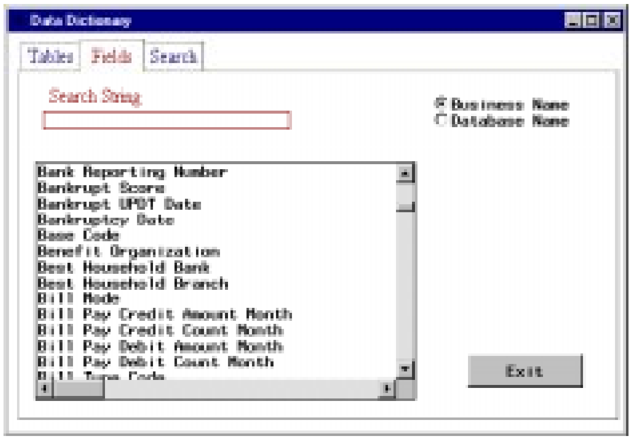
Description
TESTING SAMPLE LOADING

Figure 10. The Sample Registration Message Window

The Data Dictionary

The DEST includes a limited data dictionary of the static metadata that is accessed via the Dictionary option on the Main Menu Bar. It directly reads the metadata data sets, so it is always current.

Organized as a multi-tab window, the first two tabs provide alphabetized searchable lists of tables and fields. The user can search by business or database name. Entering a search string scrolls the display to the first item that begins with the typed string. The third tab also allows entry of a keyword or search string. Here, radio buttons control whether tables or fields are searched by database name, business name, or, most importantly, business definition.


Data Dictionary

Tables | Fields | Search

Search String

● Business Name
○ Database Name

```
Bank Reporting Number
Bankrupt Score
Bankrupt UPDT Date
Bankruptcy Date
Base Code
Benefit Organization
Best Household Bank
Best Household Branch
Bill Mode
Bill Pay Credit Amount Month
Bill Pay Credit Count Month
Bill Pay Debit Amount Month
Bill Pay Debit Count Month
Bill Type Code
```

Figure 11. The Field Search Tab of the Data Dictionary

FUTURE DIRECTIONS

The Data Extract and Sampling Tool will significantly contribute to the First Union National Bank's effort to develop the Consumer Bank of the Future. With the fast access to the Corporate Marketing Data Warehouse that the DEST provides, data modelers, analysts, and statisticians will be able to spend less time extracting and preparing data and devote more time to building accurate predictive models. This capability will provide faster turn-around for the bank's marketing efforts, thereby allowing First Union to better target its products and services to the customers who can best benefit from them.

SAS software is ideal for building applications such as the DEST. With one integrated toolset, users can build a seamless application that:

- Has a user-friendly graphical interface;
- Allows connectivity to other computers for remote compute and data transfer services;
- Provides read/write access to very large RDBMS via SQL;
- Incorporates extensive data transformation, sampling, analysis, and model building capabilities;
- Has a metadata foundation that permits modification for accessing any RDBMS; and
- Can be easily ported to other operating systems and platforms because of the software's multi-engine architecture.

As with the data warehouse itself, the DEST is dynamic and will evolve over time. Phase 2 enhancements may include:

- Parallelization of SAS program execution;
- The ability to perform exploratory data analysis on the Sampling Universe;
- Enhanced data transformation capability;
- Sampling stratification by categorical variables;
- Testing to validate the Sample against the Universe; and
- An expanded data dictionary in native Windows Help.

AUTHOR INFORMATION

John E. Bentley
Systems Programmer Consultant III
First Union National Bank
400 S. Tyron Street, 16th floor
Mailcode NC0094
Charlotte NC 28285
John.Bentley@FirstUnion.Com (no attachments) or
BentleyJ@Mindspring.Com (with attachments)

Trademarks

SAS, SAS/Connect, SAS/Access, SAS/AF, and SAS/FSP are registered trademarks of SAS Institute Inc. in the USA and other countries. IBM and RS/6000 SP are registered trademarks of International Business Machines Corporation. XPS is a registered trademark of Informix, Inc. ® indicates USA registration.

Date Extract and Sampling Tool and DEST are pending copyright and trademark registration by First Union National Bank.

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