

SAS® System Powers Web Measurement Solution at U S WEST

Bob Romero, U S WEST Communications, Technical Expert - SAS and Data Analysis
 Dale Hamilton, U S WEST Communications, Capacity Provisioning Process Analyst

the most recent data. An example is shown on how a SAS view is created:

ABSTRACT

The CP Process Analyst Team implemented a performance measurement system on U S WEST's internal web in order to meet service commitments on time and within budget. The summarized reports are used by management to review performance results, while operational data at a service order level is required to apply business rules and to produce detail reports for the engineers to run the business. Some of these measures impact team awards and performance reviews. The flexibility, speed, and large scale ability of SAS6® to access and compile report information solves numerous data merging and reporting constraints experienced in the past. This paper discusses the process flow of extracting the data from multiple ORACLE® databases and legacy systems, and using some of the power and flexibility of the Base SAS Language to apply business rules, handle out-of-process issues, and then use other SAS procedures and products to produce graphical and tabular reports on the Web. The complete end-to-end process, using SAS, is automated to produce the reports hourly, daily, weekly, and monthly, by using the UNIX "cron" facility. A significant improvement in performance has resulted from this dynamic solution using the SAS System.

INTRODUCTION

The components of Base SAS such as:

1. Statements and functions of the Data Step;
2. Canned SAS Procedures;
3. MACRO Variables and Programs;

have provided flexibility and speed to the automation of the CP Measurement System and Operational Data Reporting System at U S WEST. After data is extracted from ORACLE using SAS/ACCESS to ORACLE, the business rules and criteria are applied, and calculations are done at multi-level summarizations and stratifications. Very high speed processing of millions of rows of data using SAS data sets, provide tabular reports and summarized data sets directly to the link on a timely basis. Color graphical reports are created using SAS/GRAPH®. The Web Publishing Tools make the static reports immediately available for Web Browsers. Figure 1 shows the system architecture to produce reports for users to view on the web. Figure 2 shows how SAS plays a major role in the process. The following sections describe the functionality of the SAS Language to provide a solution.

SAS/ACCESS® to ORACLE

To connect to ORACLE and set up either SAS data sets or SAS views, the "SQL Procedure Pass-Through Facility" is most efficient. PROC SQL is used for the pass-through facility. A view takes minimal space and can be used the same way as a SAS data set in a SET statement of the DATA STEP, or in a SAS procedure. The view only has to be created once and always has

```
libname mylib 'pathname';

proc sql;
  connect to oracle as currcon(path="@dbname");
  create view mylib.sv as
  select * from connection to currcon
  (
    SELECT *
    FROM
    ORTABLE
  )
  ;
  disconnect from currcon;
quit;
```

The CONNECT statement connects to ORACLE with the database defined after the "@" and is in quotes. The name of the connection is an alias that points to the connection (in this case, CURRCON). The first SELECT statement is part of PROC SQL that selects items from the connection, which in this case is everything. The second SELECT statement is ORACLE SQL that is being passed from SAS to ORACLE. The ORACLE SQL is contained in the () so that any valid ORACLE SQL can be used here including joins. In this example, everything is selected which is probably not wise to do, unless the developer wants everything from the table, ORTABLE, in the SAS view, sv. The query is much faster by specifying only the columns and rows that are needed.

However, every time the view is accessed, an ORACLE query is submitted, which takes 10 to 20 times longer than processing a SAS data set. If a join is performed in the view, then the processing can take even longer. To expedite processing, a view is generated of each individual table, and a SAS data set is generated from each view through the SET statement and then merged with the other data sets in a DATA STEP.

Note that SAS truncates the ORACLE column names down to 8 characters and places the ORACLE names into the labels of the SAS variables. If there are any duplicates as a result, SAS truncates the name even further and starts numbering the variables in the names themselves. A new name can be given for each variable in the SELECT statements to produced the desired names. Also, if a SAS data set is created from a SAS view, the variable can be renamed in the assignment statement in the DATA step or in PROC DATASETS.

Components of Base SAS

The Display Manager System of Base SAS is an easy-to-use environment for quick development of SAS Applications and provides a seamless interface among SAS products.

Base SAS is also a powerful fourth generation language that has numerous functions and statements to apply business rules and

criteria, as well as to generate computed columns. For this application, not only is detail data at a service order needed, but the history of each service order provides valuable information at

different stages of the process. Its different statuses during its lifetime enable the developer to apply the business rules

System Components for CP Measurement and Operational Data Reports

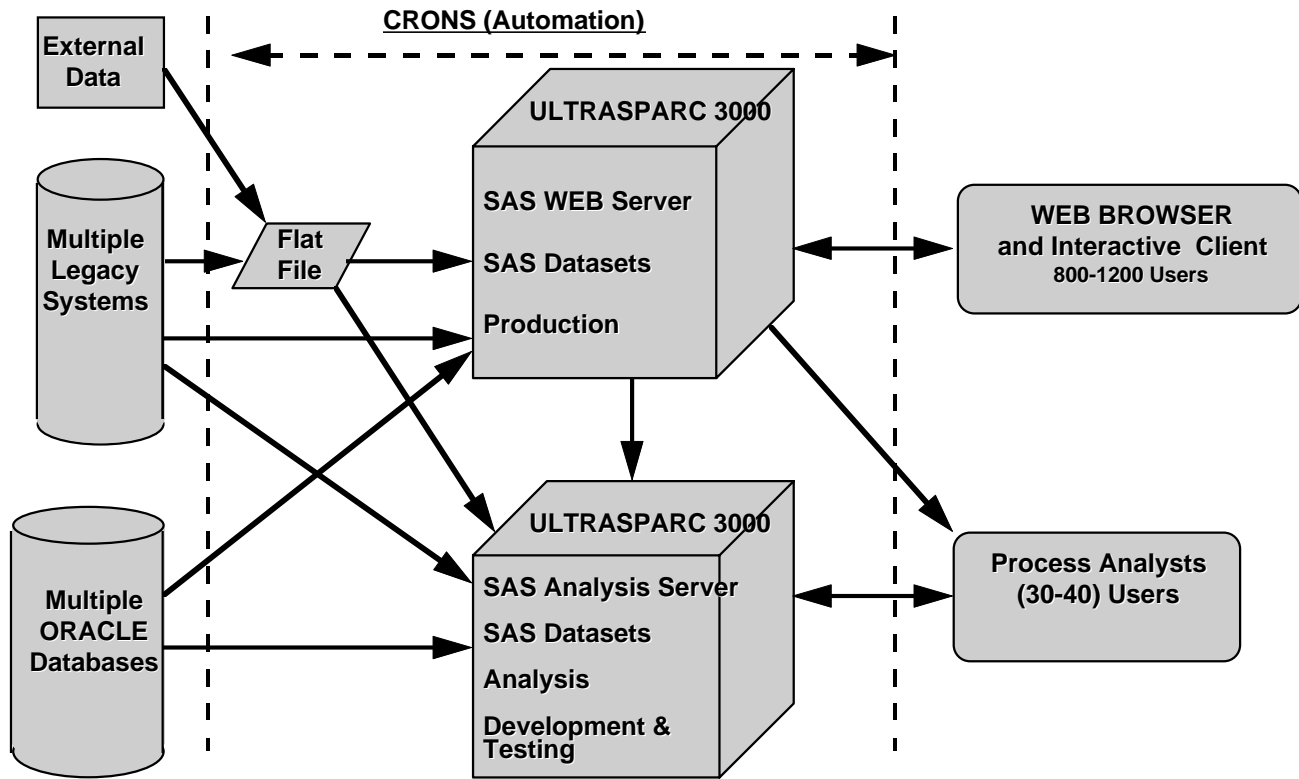


Figure 1.

Specifications of Hardware

Each Sun Ultra™ Enterprise™ Server is equipped as follows:

- 6 CPUs at 250 MHz
- 3 GBytes RAM
- 126 GBytes Disk Space
- Disk partitions mirrored

Production Data Disk Directories are shared by the Analysis Server as read-only.

Use of SAS in the CP Process and System Servers

- | |
|--|
| <ol style="list-style-type: none"> 1. Fast Tool Development and Customization <ol style="list-style-type: none"> a. SAS Interface to Web b. Powerful 4th Generation Language c. Object-Oriented Programming to build Graphical User Interface (GUI) tools d. Cross-Platform Capability/Interface |
| <ol style="list-style-type: none"> 2. Business Decisions and Analysis <ol style="list-style-type: none"> a. Optimize Network Placements and Revenue Opportunities (Statistical Analysis) b. Process Improvement (Root Cause Analysis) |
| <ol style="list-style-type: none"> 3. Complete Automation of Reports <ol style="list-style-type: none"> a. Corporate Measurements <ol style="list-style-type: none"> 1. Process 2. Production 3. Financial b. Operational Reports <ol style="list-style-type: none"> 1. Summary 2. Detail 3. Graphical c. Data Gathering to On-Line Reports |
| <ol style="list-style-type: none"> 4. Data Warehousing <ol style="list-style-type: none"> a. Direct Access to ORACLE/Relational and Corporate Databases, i.e. Microsoft and Legacy Systems b. Very High Speed Processing of millions of rows of data using SAS binary files (Ex. One SAS data set with 22 million rows and 8 columns took 7 minutes of real time and 5 minutes of CPU time in a Data Step. The same query in ORACLE took over 5 hours.) c. Multi-level Summarizations and Stratifications |

Keys to Success:

1. Control of and Access to Data
2. Quick Response to Customer Needs
3. Quick and Inexpensive Customization, Automation and Development

Figure 2.

necessary and determine if a commitment was made or missed. The RETAIN statement combined with the FIRST. and LAST. variables in a DATA step allow pertinent information about each service order to be flagged and kept with the service order for the output data set. Special date and time functions are very useful in working with date or time columns.

Canned procedures also reduce the amount of code and time necessary to generate reports, statistics, and other results, which include :

PROC FORMAT, PROC REPORT, PROC PRINT, PROC SUMMARY, PROC SORT, PROC UNIVARIATE, and PROC SQL, to name a few.

PROC SUMMARY produces statistics such as counts, sums, and average cycle times at different levels of stratifications and at different combinations of stratifications. The values of the `_type_` variable illustrates at which level the statistics are given, and a multi-dimensional summarized data set is produced. The number of groupings or levels is 2 to the power of the number of variables in the CLASS statement. When a level shows that the statistics were computed for all values of a variable, the value of the variable is missing. For example, the following CLASS statement produces 4 levels of summarization.

```
class state month;
```

The values of `_type_` go from 0 to 3. Level 0 is across all states and all months, level 1 is each month across all states, level 2 is each state across all months, and level 3 is each month of each state. Thus statistics for the whole company and year-to-date are automatically given. Taken through another DATA step, a value of "USWEST" can be assigned to state where there is no value of state and YTD can be assigned to the value of month where month is missing. Also, correct average percentages can be calculated in this data step. If not all of the values of the `_type_` are desired, the summarized data set can be subsetted with a where statement.

PROC UNIVARIATE is extremely useful in calculating the mean, median, mode, and percentiles of intervals of the company's response to customers.

PROC FORMAT is especially useful in listing coded or descriptive values for a variable. For example, if you want to keep your value for month numeric (1-12), and yet you want to show 3-letter month names on reports and graphs, one way you can accomplish this is by using PROC FORMAT in the following manner:

```
proc format;
value monthfmt 1 = 'Jan'
                2 = 'Feb'
                3 = 'Mar'
                4 = 'Apr'
                5 = 'May'
                6 = 'Jun'
                7 = 'Jul'
                8 = 'Aug'
                9 = 'Sep'
                10 = 'Oct'
                11 = 'Nov'
                12 = 'Dec'
                13 = 'YTD';
```

```
run;
```

The format is assigned in the format statement like in the following statement:

```
format month monthfmt.;
```

The value of 13 is assigned to month for year-to-date totals and then can be used on the same charts with monthly values.

Macro variables and the SAS MACRO language also greatly enhance the flexibility and reduce the amount of code and time. One simple example of a macro variable would be:

```
data _null_;
adate = today()-1;
call symput('enddate',put(adate,date7.));
run;
```

Then the macro variable `enddate` is referenced in a title or footnote statement with an & (i.e. title1 "Totals for USWEST as of &enddate";) or as a date comparison variable (i.e. where reportdt le "&enddate"d;)

If a certain processing task such as a detailed report for each state, time period, or other values of other grouping variables is needed, then a simple use of the SAS Macro Language is to use the value of a grouping variable as an input parameter to a Macro Program.

PROC REPORT and PROC PRINT produce simple to very complex reports. The DATA _NULL_ step with PUT statements can be used to generate extremely customized reports. It can also be used to build UNIX scripts, other SAS programs, comma delimited files and so forth! The X statement or FILENAME statement with the PIPE option can be used to interface with UNIX.

The procedures in SAS/GRAPH produce color graphics such as bar charts, line charts, and pie charts.

Other important procedures include those in SAS/STAT® and SAS/QC® that do more advanced statistical analysis.

SAS Web Tools

The Web Publishing Tools are available for free and are documented on SAS Institute's home page. Four Device drivers to produce color plots and charts at different sizes are available, and generate .gif files for viewing on the Web. The %out2htm tool produces HTML ready reports from PROC PRINT, PROC REPORT, or from the DATA _NULL_ step. The %ds2htm tool converts a SAS data set to an HTML table. The SAS labels of the variables are printed at the top of the page. Titles can be added to the files. The FILENAME statement with the FTP option, and the use of the HTMLFREF, in any of these tools, will automatically send any of these files to the desired paths on the Web server for immediate display by the browser.

Other products needed to support the CP Measurement System are SAS/AF®, SAS/FSP®, SAS/CONNECT®, and SAS/EIS®.

The Dynamic Web Tools in SAS/IntrNet®, SAS/MDDB®, and SAS/SHARE® give users the opportunity to produce immediate graphical and tabular reports on the WEB.

The full process of getting the data, applying the business rules, summarizing, and then generating the graphical and tabular

reports on the Web, are completely automated using the UNIX cron facility. The combination of all of these tools serve as a WEB-driven, Executive Information System.

Littleton, CO 80120
(303) 707 - 7846
rromero@uswest.com

Dramatic Improvements at U S WEST

1. U S WEST exceeded 1997 held order target by 36%.
2. U S WEST exceeded 1997 missed order target by 18%.
3. % Estimated Engineering Complete dates met are up 26% over 2 years ago.
4. % Estimated Ready For Service dates met are up 33% over 2 years ago.
5. Capacity Provisioning managers achieved the highest team award ever.
6. The reports generated by SAS would have required an additional 117 people to do the reports manually, and with less dependability.
7. Prototype results justified the business case approval for full system funding.

Dale Hamilton
700 W. Mineral Ave., Rm OR G12.33
Littleton, CO 80120
(303) 707 - 7825
dhamilt@uswest.com

Using SAS to automate the CP Measurement System and Operational Data Reporting, empowers U S WEST Managers and Engineers to significantly improve on service commitments, and to stay within budget. Batch reports have been replaced with paperless access to data, customized reports and graphical trend charts. The SAS software populates information from multiple systems into powerful business and resource management models that allow accurate and timely business decisions. As a result, SAS enhanced "Visibility Tools" are fast becoming the standard at CP and at U S WEST.

SAS Dynamic web tools allow users to do adhoc reporting and focus on achieving desired results. It provides a powerful search engine with no learning curve. With these tools, decision support and on-line analysis have reached a new level of accessibility.

Summarized reports and the supporting detailed data are updated on an hour or half hour basis. This immediate automated reporting allows rapid intervention at key process points to positively impact U S WEST customers' experience and results. It also supports an "in today / out today" concept of processing service orders. Ongoing development using a powerful mix of software, hardware, and communications technology, continues to "Make Life Better Here" for U S WEST customers, as we prepare for new challenges in the 21st century.

SAS, SAS/ACCESS, SAS/AF, SAS/CONNECT, SAS/EIS, SAS/FSP, SAS/GRAPH, SAS/IntrNet, SAS/MDDDB, SAS/QC, SAS/SHARE, and SAS/STAT are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.® indicates USA registration. ORACLE is a registered trademark or trademark of Oracle Corporation in the USA and other countries. ® indicates USA registration.

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

Bob Romero
700 W. Mineral Ave., Rm OR H12.3