Cross Platform Generation of Reports from Oracle
Using SAS® on MVS, Macintosh and AIX

Morvyn Lindsey, The California State University
Office of the Chancellor, Long Beach, CA

ABSTRACT

The objective of this paper is to discuss the implementation of a system to generate reports from an Oracle database using the unique features and functionality of SAS® on each of three different computing platforms: MVS, Macintosh and AIX.

This paper will discuss how the CSU Chancellor's Office Corporate Information Systems team developed a system to generate over 300 annual reports from an Oracle7 database using SAS/Macintosh. These reports had been previously generated via COBOL from flat files on MVS.

The methodology that was developed makes use of SAS/CONNECT® and Base SAS® on all three platforms, SAS/ACCESS® for Oracle and PROC REPORT on AIX, and SAS/EIS® and FRAME® on the Macintosh.

INTRODUCTION

The project described in this paper demonstrates how SAS can be used in a cross platform manner to generate reports from a corporate data warehouse, The information presented is intended to provide other SAS sites with helpful information and insights that can be used to plan and to implement similar projects.

Additionally, this paper documents the hard work and effort put forth by the CIS project team which included Sandra Elliott, John Martin, Sam Smolker, June Sung and the author.

PROJECT OVERVIEW

The Enrollment Reporting System Conversion project arose out of the need of the Analytic Studies Department to generate the annually produced systemwide and campus enrollment reports for the 22-campus California State University system from the newly created corporate data warehouse residing in an Oracle7 database on an IBM RS/6000.

The enrollment reports had been previously done completely on an MVS mainframe from flat and VSAM files using COBOL and other tools such as Easytrieve Plus. The generated reports were output to flat files. The Analytic Studies Department could then make the report output files directly available to the campuses on the MVS system via FTP or download them to their desktop Macintosh or to their departmental server. If required, the reports could be exported into Excel spreadsheets or into Word documents and made camera ready and made available on our AppleTalk network to the campuses that are directly connected to the AppleTalk network.

The goal of the project team was to reproduce the enrollment reports from the Oracle7 corporate data warehouse instead of from the MVS file system.

THE HARDWARE AND SOFTWARE ENVIRONMENT

Early on, it was decided to use SAS to complete this project because corporate decisions had been made previously to use SAS whenever possible and to avoid writing any more COBOL programs. Also, SAS was currently running on the MVS, AIX and Macintosh platforms and the team was confident that it could easily test and run the code on all three platforms as necessary.

The most critical software components were SAS/CONNECT running on all three platforms and SAS/ACCESS for Oracle running on the AIX system. With these products in place each team member was enabled to work independently in the manner which was most efficient and comfortable for them personally.
Cross Platform Generation of Reports from Oracle
Using SAS® on MVS, Macintosh and AIX

Maryyn Lindsey, The California State University
Office of The Chancellor, Long Beach, CA

FLEXIBILITY IN WORKING STYLE

MVS TSO

A team member could choose to stay on MVS and use ISPF or the SAS Editor to modify code and then test the code via a SAS/CONNECT session from MVS to AIX where the source program would invoke SAS/ACCESS for Oracle on AIX. Due to poor TSO response time this method was eventually abandoned by all team members.

SAS System for the Macintosh

A second approach taken was to run from the desktop via SAS/Macintosh to edit the SAS code and to again test the code via a SAS/CONNECT session to AIX.

SAS/Macintosh was run in either of two configurations. One team member that had the required hard disk space and an FPU ran SAS/Macintosh completely from his own Macintosh. Two other team members that did not have a Macintosh hardware configuration capable of running SAS/Macintosh standalone, ran in a client/server mode with the SAS software resident on a server and their SAS data and program libraries resident on their desktop Macintosh.

AIX

The two team members that converted the bulk of the report programs eventually determined that logging on to AIX directly was the most efficient and easiest way for them to work.

ENABLING TEAMWORK

It is important to note that, due to the flexibility of SAS to run on all three platforms, the individual team members were allowed to work independently within the team with no need to try to force a team member to work on a platform or in a manner in which they were not familiar or comfortable with.

For example, some team members preferred to use the MVS PDS structure or flat files to store source programs and macros, while other team members more experienced with SAS used SAS data libraries. Thus, the method a team member used to move files between the computing platforms was either FTP and the SAS INCLUDE command or the use of PROC UPLOAD and PROC DOWNLOAD based on their individual preference.

This personal flexibility enabled the team members to focus their efforts on getting the project completed and fostered teamwork.

PROJECT STAFFING

The original staffing plan was to use eight staff members for the coding including two managers to be used as needed due to the large number of reports to be converted.

Two months into the project it was determined that only three team members would be needed to do the coding due to the speed of the development process and because of the need to assign the other staff members to other projects. It was determined that three team members Sandra, John and Sam, would dedicate approximately 80%, 95% and 50% of their time respectively to the project until it was completed.

One team member prepared the 13 prototypes and macros and ensured that the programming standards and procedures were adhered to as well as personally coding 165 reports.

A second team member coded 51 reports and coordinated the day to day efforts of the development team. This team member also performed most of the quality assurance duties, completed the end-user production FRAME entries and SAS/EIS menus to
submit the jobs to AIX via SAS/Macintosh and worked to resolve technical problems with other team members or with SAS Technical Support as required.

A third team member did SAS source coding only and programmed 116 reports.

A fourth team member was instrumental in setting up the SAS macros at the start of the project and converted 2 reports.

A fifth team member provided technical assistance and expertise with base SAS, UNIX, SAS/Macintosh, SAS/EIS, FRAME, and SAS/CONNECT as well as occasionally serving as a "referee" when technical problems arose.

PROJECT PLAN

SAS PROGRAMMING GUIDELINES

A SAS Programming Standards Document was prepared and distributed to all team members. It identified nine ERS report categories, defined a naming convention and specified that PROC REPORT was to be used for all programs.

The SAS Programming Standards Document Table of Contents included the following:

Introduction
Getting Started on a New Program
General Coding Rules
Coding rules for Specific Situation/Report Types
Testing of Non-delimited Output Reports
Coding an Export File for Download
Testing of Delimited Output (Export) Report
Warnings and Recommendations
Wrapup
Sample Program and Output

The reporting categories and program naming conventions included in the SAS Programming Standards Document were as follows:

<table>
<thead>
<tr>
<th>Report Category</th>
<th>Program Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age, Sex, Student Level</td>
<td>AGEnn</td>
</tr>
<tr>
<td>College Year</td>
<td>CYRnn</td>
</tr>
<tr>
<td>Comm College Transfers</td>
<td>CCTnn</td>
</tr>
<tr>
<td>Degrees Granted</td>
<td>DEGnn</td>
</tr>
<tr>
<td>Enrollment by Ethnicity</td>
<td>ETHnn</td>
</tr>
<tr>
<td>Majors</td>
<td>MAJnn</td>
</tr>
<tr>
<td>New Students</td>
<td>NWSnn</td>
</tr>
<tr>
<td>Residence Status/Citizenship</td>
<td>RSCnn</td>
</tr>
<tr>
<td>Term Enrollment</td>
<td>TRMnn</td>
</tr>
</tbody>
</table>

PROTOTYPES AND MACROS

Thirteen prototypes were created to be used as templates for the various types of reports to be generated.

SAS macros were created to be placed at the beginning of each program to create the SAS/ACCESS global access descriptors and views and to pass any other global information required for a particular report type.

A sample setup is shown below.

```sas
rsubmit;
%let TABLENAM=AGCY.AGGR_CY_05;
%let USERID=ARSGXXX; /* Oracle user id */
%let PASSWORD=xxxxx;
libname RS 6000GLOB
  '/home/staff/cast/sas/views/global';
%include '/home/staff/cast/sas/source/accdesc/cyr';
libname RS 6000VIEW
  '/home/staff/cast/sas/views/ers/statrpts';
proc access dbms=ORACLE accdesc=RS
  6000GLOB.CYR.ACCESS;
create RS 6000VIEW.ETHT06.VIEW;
select xxxx xxxx;
subset xxxx xx.xx;
run;
endsubmit;
```

Oracle common table references for campus codes, discipline codes, and ethnic group codes were defined in global libraries.
Cross Platform Generation of Reports from Oracle
Using SAS® on MVS, Macintosh and AIX

Marvyn Lindsey, The California State University
Office of The Chancellor, Long Beach, CA

Requirements for the generation of the output reports originally included the creation of two output file formats for each report. The first was to be a standard non-delimited report format and the second was to be a semicolon delimited report format that could be downloaded into Excel on the Macintosh desktop.

Even though the client later indicated that only the non-delimited format would be required, the team continued with coding both report format types.

The model PROC PRINTTO to generate the non-delimited report format was:

```sas
proc printto print=REPORT new;
proc REPORT data='""UXX .... (generate main detail body of non-delimited report)
proc printto print=REPORT;
proc REPORT data=xxxx ...
  (generate subtotals and grand totals lines)
```

The model PROC PRINTTO to generate the delimited report format was:

```sas
proc printto print=EXCELRPT new
```

A four digit year value appears within the page header of most status reports and was coded via the global macro variable &SYYYYY.

A push button object was subsequently added to the production ERS Reports Menu on the Macintosh to allow the client to dynamically set this macro variable via the autoexec.sas file on AIX.

QUALITY ASSURANCE PROCEDURES

To ensure the accuracy of the reports, a Quality Assurance Procedures Document was created.

In that document, the team members were given procedures to validate the report numbers against the 1993 published reports, instructions to inform the QA person via e-mail of completed reports, information on how to access the Document Distribution Server and information on how to copy output samples to the server for review.

PROJECT DURATION

The project started at the end of May 1995 with the creation of the guidelines and other documentation and the creation of the report prototypes. The first completed reports were coded in early June 1995. All reports were completed by the end of December 1995. Four months after the start date 262 of the 334 reports (78.4%) had been completed. Six months after the start date 332 of the 334 reports (99.4%) had been completed.

In January 1996 work turned to finalizing the Macintosh front end via SAS/EIS menus and FRAME entries and learning how to run the SAS programs in batch mode on the AIX instead of running them in real-time mode as was done during testing and development.

TECHNICAL ISSUES RESOLVED

During the project several technical issues were identified and resolved by the team. These issues included how to handle the wide page width of some of the reports, how to replicate the column headings in some of the reports, how to run batch jobs on the AIX system and how to improve the run time performance of the batch jobs.

PAGE WIDTH AND COLUMN HEADINGS

Reports that required a page width of 133 columns caused problems until the programmers learned to co-ordinate the pagesizes of both their desktop and remote SAS sessions.
Cross Platform Generation of Reports from Oracle Using SAS® on MVS, Macintosh and AIX

Marvyn Lindsey, The California State University
Office of The Chancellor, Long Beach, CA

The column headings of some reports were difficult to replicate in PROC REPORT until the team learned how to span column labels with SPAN and learned how to use an auto fill character over a spanned label.

**BATCH JOBS ON AIX**

During development and testing, individual report programs were tested standalone. Because the RS/6000 is extremely fast, the programmers did not suffer too greatly from their workstation Macintosh or AIX connection being tied up while their test job would run. However, when runs of complete categories of reports were tested it was obvious that we needed to run the job streams in a batch mode on AIX so that the programmers workstation could be used to do other work while their batch job stream was running.

The SAS Companion for UNIX provided the information we needed to invoke SAS in a batch mode. Below is sample SCL code developed on the Macintosh to invoke the batch submission of a report program on AIX via a push button object in a FRAME entry from the Macintosh:

**SUBOBJ:**
/* Submit Report Request to AIX */

```
if rlink('cas')=0
   then _msg_='Please signon to cas first to run reports.';
else do;
   submit continue;
   rsubmit cas;
   x cd /usr/local/sasprod;
   x sas /sas/source/ers/statrpts/cct99.src -print $HOME -log $HOME & ;
   endsubmit;
   endsubmit;
   _msg_='The reports have been submitted.';
   end;
return;
```

**PERFORMANCE ISSUES**

Further testing of complete reporting categories revealed that the elapsed time (wall clock time) to run some of the categories was extremely high, especially for an RS/6000.

For example, the New Students report category produced 128 reports and took 235 minutes to complete.

The main problem was that the macros to create the access descriptor and view were being run before each report even though most of the reports used the same access descriptor or view. After re-working the job flow to avoid this problem, the entire New Students report category completed in 83 minutes. The elapsed time for the Majors report category, which generates 22 reports, went from 110 minutes to 33 minutes.

Further work will be done in the future to improve the performance even more. Testing needs to be done to compare the performance differences of using SAS datasets alone versus using SAS/ACCESS for Oracle with SAS datasets (the current method) versus using SQL Pass Through to eliminate the use of SAS/ACCESS for Oracle.

**FINAL IMPLEMENTATION**

Final implementation of the project included the creation of a User's Guide and completion and distribution of a GUI front end on the Macintosh to the client.

The User's Guide includes procedures to run the system, pictures of the various menu screens and frames, a list of the report categories and report names, and the names and locations of the output files.
Cross Platform Generation of Reports from Oracle
Using SAS® on MVS, Macintosh and AIX

Marvyn Lindsey, The California State University
Office of The Chancellor, Long Beach, CA

The GUI front end on the Macintosh includes SAS/EIS Menus which allow the user to set the academic year to be run, to select the report category to be run, and to view the log files from the batch runs on AIX.

ACKNOWLEDGEMENTS

SAS, SAS/ACCESS, SAS/CONNECT, SAS/EIS, FRAME and SAS System for the Macintosh are registered trademarks of SAS Institute Inc. in the U.S. and other countries. ® indicates USA registration.

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

AUTHOR CONTACT INFORMATION

Marvyn M. Lindsey
The California State University
Office of the Chancellor
P.O. Box 3842
Seal Beach, CA 90740-7842

Telephone: 310-985-9604
Fax: 310-985-9400
Internet: marv@calstate.edu