

# The Realities of Downsizing -- Part II: Moving a SAS<sup>®</sup> Application from MVS<sup>®</sup> to UNIX<sup>®</sup>

Tracy A. Cermack, American Honda Motor Company, Inc.

Kimberly J. LeBouton, K.J.L. Computing

## Abstract

Downsizing in the current business world is deemed necessary for companies to remain competitive. Companies are downsizing mainframe applications to smaller platforms in the hope of saving money and gaining productivity for both users and developers. *But is this goal being achieved?*

This paper explores the realities of moving an entire user department (SEI), and their SAS applications, from MVS to UNIX at a major manufacturing company. It also provides an update on the more recent adventures and experiences in the implementation of this project, which was first presented at SUGI 22<sup>1</sup>.

## Key Issues

This paper will discuss the following issues related to the conversion project and their impact on the project:

- Moving 20 gigs of data across platforms on a weekly basis: reviewing SAS/CONNECT<sup>®</sup>, FTP, and CLIO/S<sup>®</sup> downloading processes
  - ◊ The Future Production Environment – Tivoli and Platinum Autosys
- IS Management Considerations
  - ◊ Reduction in mainframe CPU utilization?
  - ◊ Taking responsibility for a new OS – Oh! Those Politics!
  - ◊ IS staffing shortages
- Applications & Programmer Considerations

◊ The GUIness of It All – RAD and FRAME Applications

- User Considerations
- Those Nasty Cost Considerations
- More! Bigger! Faster!
- Suggestions

## Downloading the Data ... Yikes!

Whether it's related to transferring information via the Internet, refreshing a data warehouse, or transferring data between two different computer platforms, the necessary technology has simply not caught up with the need to move data quickly.

The biggest single issue for this project was transferring over twenty gigs of data across platforms on a weekly basis. More than sixty percent of the consultant's<sup>1</sup> time was spent on this issue alone. The twenty gigs of data contained 356 data sets, primarily SAS data sets. Additionally, the required mainframe flat files and DB2 files were converted to SAS files on the RS6000 platform during the download process itself. This proved to be a nice side benefit for the users and programmers.

## Weekly Transfer of Data

Three different approaches were tried before the best transfer method was determined. (See LeBouton<sup>1</sup> for a complete discussion.) The first approach involved SAS/CONNECT.

The connection for SAS/CONNECT originated on the RS6000. The transfer speed using SAS/CONNECT was too slow to work within the stated constraints. There were also connection problems. Programs were created to restart the connection if it was interrupted, but the mainframe account would get locked up for hours. It was concluded at that time that SAS/CONNECT would not be able to handle the demands of such a large, frequent transfer.

The second approach was to try using the FILENAME FTP transfer method that became available with version 6.11 of base SAS. Unfortunately, there were problems with this method. Subsequent installation of version 6.12 resolved these issues, but by that time other transfer methods had been established (as discussed below).

The final approach was to create production jobs using PROC CPORT, and then FTP the file down to the RS6000. A SAS program was developed on the RS6000 to grab the transported file and turn it into a SAS data set with PROC CIMPORT. It took over three hours

of CPU time to CPORT the data and over 8 hours of I/O processing to transfer the data down to the RS6000.

### CLIO/S -- The Success Story

To cut transfer rates in half, hopes were placed on CLIO/S -- an IBM® data transfer product. In the earlier report<sup>1</sup>, it was stated that CLIO/S did not perform as hoped. Subsequently, with fine-tuning and appropriate installation corrections, CLIO/S proved to be the champion its advertising claimed ... even better, actually.

The product was purported to improve transfer rates by 50% on large data sets. In this application, that figure was far exceeded (see Figure 1). The "large" data sets (approximately 1 GB) showed a transfer rate three times faster than without CLIO/S; "medium" data sets (approximately 54 MB) transferred five times faster; and small data sets (approximately 25 KB) transferred in half the time. Big Blue is not dead!

**Figure 1. DOWNLOAD TIMES: CLIO/S vs. FTP**

| member              | obs<br>length | # obs in each file |            |              | total bytes in each file |                   |                    |
|---------------------|---------------|--------------------|------------|--------------|--------------------------|-------------------|--------------------|
|                     |               | small              | medium     | large        | small                    | medium            | large              |
| A1                  | 44            | 12                 | 3,002      | 39,576       | 528                      | 132,088           | 1,741,344          |
| A2                  | 36            | 11                 | 2,443      | 27,375       | 396                      | 87,948            | 985,500            |
| B1                  | 32            | 11                 | 0          | 11           | 352                      | 0                 | 352                |
| C1                  | 410           | 50                 | 111,795    | 2,054,288    | 20,500                   | 45,835,950        | 842,258,080        |
| E1                  | 19            | 106                | 126,778    | 1,724,436    | 2,014                    | 2,408,782         | 32,764,284         |
| L1                  | 26            | 11                 | 25,660     | 505,641      | 286                      | 667,160           | 13,146,666         |
| P1                  | 37            | 16                 | 125,185    | 1,976,330    | 592                      | 4,631,845         | 73,124,210         |
| T1                  | 19            | 11                 | 0          | 11           | 209                      | 0                 | 209                |
| <u>T2</u>           | <u>49</u>     | <u>11</u>          | <u>271</u> | <u>5,859</u> | <u>539</u>               | <u>13,279</u>     | <u>287,091</u>     |
| All                 | 672           | 239                | 395,134    | 6,333,527    | <u>25,416</u>            | <u>53,777,052</u> | <u>964,307,736</u> |
| CPU download times: |               |                    |            | FTP:         | 0.2/minute               | 2.5/minute        | 0.97/minute        |
|                     |               |                    |            | CLIO/S:      | 0.1/minute               | 0.5/minute        | 0.34/minute        |
| Approx. change:     |               |                    |            |              | 1/2 the time             | 1/5 the time      | 1/3 the time       |



## The Future Production Environment

Two upgrades still underway for the RS6000/UNIX environment are the addition of Tivoli and Platinum Autosys. These additions are being utilized in order to provide a more mainframe-like production environment in which automated alerts are provided for problems encountered during production processes.

Currently, UNIX "production" processes are run by traditional cron jobs, as is SAS/Share. When SAS/Share has gone down, no one is alerted until a user complains that they can't get into a menu (because of a libname problem with the SAS/Share Server). Obviously, this is not the best situation.

Tivoli will be used to monitor SAS/Share and is intended to provide an automated alert at any time the process goes down. In addition, hopes are placed on Platinum Autosys which, similar to ZEKE & ZEBB (or UCC7/CA7) in mainframe-land, schedules and monitors "production" jobs during transfer of data, etc., and provides a programmer alert if any jobs abort. This will be the company's pilot study for the product.

## **IS Management Considerations**

### Reductions in Mainframe CPU Utilization?

During the initial stages of this project, it was determined that this particular end-user group (SEI) was utilizing fifty percent of the MIPS on the end-user mainframe side, which accounted for approximately 8 to 10 percent of the entire 3090 mainframe capacity. The IS department was excited at the prospect of gaining back that processing time.

Unfortunately, but expectedly, this processing time was quickly utilized by other end-users, bringing the mainframe back to its previous use levels within one month of SEI moving to the UNIX environment.

In addition, there were the necessary increases in mainframe CPU utilization required for the

download process itself. Even though these processes are carried out during non-peak hours, it does place an additional burden on the mainframe's production processes.

Any eager anticipation of mainframe CPU returns were certainly not realized.

### Assigning Responsibility for a New OS -- Oh! Those Politics!

Since the RS6000/UNIX environment was new at the company, assigning responsibility for the caretaking of a new OS and hardware was a major political hot potato! Meeting after meeting was held with no one stepping up to accept responsibility for these new tasks. The next step is for the user group representative to go directly to the VP of IS and discuss the serious lack of support in the situation.

The consultant was and still is retained on site to handle these tasks:

- ◇ Production Downloads
- ◇ File Type Transfers (DB2 to SAS, etc.)
- ◇ Upgrades to Production Jobs
- ◇ Normal Maintenance Issues and User Needs

Additionally, consultants remain at the helm of the RS6000 itself, controlling and maintaining all processes related to the hardware, its environment, and the UNIX OS.

Obviously, it can cause serious anxieties for users when they are required to move their entire computing dependency to an environment which has no answerable management support and in which no permanent corporate employee has training, knowledge, or responsibility. These issues have yet to be resolved and still cause palpable consternation within the user group.

### IS Staffing Shortages

Another glitch in the implementation of this project was IS's staff shortages. IS is notoriously understaffed (particularly when the users are trying to get something accomplished!) ... or at least that is what we

are told. There was a severe shortage of personnel with the qualifications, experience, and knowledge needed to perform work in the UNIX environment. This prompted the hiring of consultants to carry out this work. Even so, the non-availability of knowledgeable personnel when needed for specific tasks, was often the cause of schedule delays during this project.

## Applications and Programmer Considerations

Two quick items here: (1) mainframe dinosaur programmers do adjust, even though they may for a time feel they have lost some of their power because of not being “expert” in the new OS and its commands and quirks. (2) If there is any way you can get larger monitors for programmers/users, it is highly recommended! This author laughed at the thought of exchanging her 17” monitor for a 21” monitor ... there was some mention of “waste of money” and “total silliness,” if memory serves. HOWEVER, that 21” monitor is a true benefit in the UNIX world ... being able to view multiple windows simultaneously, and actually being able to read what’s in them, is a big time saver and a wonderful convenience. (And the boss can’t see you too easily around that big thing either!)

### The GUIness of It All – RAD and FRAME Applications

The SEI users (analysts) and the programmers were quite happy with the SAS menuing systems on TSO. Sure, there were additions to the systems which the analysts requested, and those were added as time allowed. But most of the complaints, of course, were about the slowness of the mainframe system itself ... users and programmers alike were frustrated by this continuing, unresolved problem.

This being the case, the hidden benefits of SAS in the UNIX environment became another reason for delight. Menuing systems soon became point-and-click rather than fill-in-the-blanks, and new applications which were intended to ease the users’ transition to the

new OS proved to be doorways to ever more popular functionality.

Separate menuing systems, which were related but nonintegrated, were brought together via umbrella menus. While this was something that the programmers had wanted to do for years, the UNIX platform made the task easy and fun, and provided a learning experience at the same time.

One of SEI’s programmers developed a SAS FRAME system so that the users could view the jobs they currently had in the background waiting to run at later times, much like an ISPF listing of the “queue” in TSO. This allowed the users to see what times were underutilized for the submission of background jobs. It also allowed the users to view, change, and delete jobs which they had previously submitted to run later. This application proved to be very popular with the users.

In addition, SEI wasn’t completely happy with the graphics catalog functionality in SAS. Too much ‘back and forth’ to view graphs and try to figure out which graphs were the required ones. Programmers developed their own graphics viewing application in SAS FRAME ... allowing the users to view the listing of graphs, and view graphs on the same screen at the same time, print with a function key (and without changing any other options), and sort, organize, and delete graphs ... all from the same screen, and without knowledge of any additional SAS, UNIX, or other commands. This application saves time for the users, and they are extremely happy with the new functionality gained with this approach.

Naturally, the programmers had a field day creating new front-ends for the existing menuing systems. So many new avenues were presented for Rapid Applications Development that the programmers were only limited by decisions regarding which direction to travel first. New fun phrases were heard around the office, including “Whoa ... they’re gonna love this one!” and “Bill Gates has nothing on this!”

All in all, the programming staff was, and still is, getting a lot more done than before ... and having a ball doing it.

## User Considerations

User adaptability and training issues are kept to a minimum if the users are familiar with a PC environment. The users adapted to UNIX easily because of this fact and because SEI's programmers developed several systems which made UNIX knowledge unnecessary to the users (see Applications Considerations for discussion). The users involved in this project remain extremely enthusiastic about the UNIX environment.

Other user considerations include (1) larger screens, for the same reasons stated in Programmer Considerations; and (2) costs (see next section).

## Those Nasty Cost Considerations

As in so many larger corporations, and shortly prior to the beginning of this project, a corporate-wide "Chargeback" system for computer usage was implemented. This is, fundamentally, an exchange of "funny money" (budgeted dollars exchanged inside the corporation only) in order to support the existence of the IS department. For those of you who have not experienced such a system, it is basically justifying your department's computer usage/time for the entire budget year ... and it's quite a delight.

During the initial phase of the project which was during Fiscal Year (FY) 96, IS maintained that SEI's UNIX costs would actually be less than their TSO costs. As shown in Figure 2, this expectation did not prove to be correct due to several factors:

1. The Chargeback system was completely overhauled during FY96, causing all MVS charges to be greatly reduced.
2. With the original, smaller, UNIX SP2 box during FY98, the charges were

approximately equal to MVS process charges during FY97.

3. The response times were not adequate for SEI users, resulting in the upgrading of their hardware to an SMP box in FY98. This significantly increased costs.

What is noteworthy here is not to expect any great dollar savings ... at least not on paper. The true savings are in the faster response times garnered from the new environment, and the concomitant increase in throughput and user satisfaction.

**Figure 2. SEI's Chargeback Budget**

| Year    | MVS       | UNIX      | Total     |
|---------|-----------|-----------|-----------|
| FY96*#  | \$435,688 |           | \$435,688 |
| FY97*   | \$227,689 |           | \$227,689 |
| FY98**  | \$28,800  | \$194,580 | \$223,380 |
| FY99*** | \$28,800  | \$278,580 | \$307,380 |

# Prior to overhauling the Chargeback system.

\* Prior to UNIX implementation.

\*\* In FY98 and beyond, MVS usage is primarily for CICS table lookups -- minimal TSO usage.

\*\*\* In FY99, the UNIX budget includes a larger (SMP) UNIX box.

## More! Bigger! Faster!

As mentioned in the COSTS section above, the SEI department did require and request a larger UNIX box. This was due to response times not being as fast as the users anticipated, expected, or required. (Forget the fact, please, that the UNIX response times were already an improvement of more than 100% over the TSO response times!) As happened with implementation of SEI's SAS menuing applications, once the users were given a taste of what might be possible, their desire for "more! bigger! faster!" was whetted ... and these people are not afraid to ask for more!

SEI includes three full time SAS programmers who design and implement SEI's SAS systems and provide ad hoc programming support, and a dozen analysts who utilize specifically-

designed SAS menuing systems continuously throughout primary work hours. Users were instructed and encouraged to utilize the off hours (i.e., evening and nighttime hours) by submitting more “background” jobs during those hours. A SAS system was implemented so that the users could select the time they wanted their jobs to run in the background environment, and review what jobs were currently in the “queue” for the off-hours.

While this approach was successful, there were still response difficulties during primary work hours. Additionally, the analysts often need an answer RIGHT NOW, and therefore cannot wait for jobs to run overnight, etc. It was for these reasons that the larger SMP box was requested for the FY99 budget cycle.

## Conclusion and Suggestions

- Be prepared for delays!
- Getting a prominent figure in IS to take an interest in, or responsibility for, your project will be a big step up.
- Having the needed technical expertise in place ahead of time will avoid numerous problems down the line.
- Transfer of data between platforms is one of the biggest issues, along with data set management.
- If you have a solid data transfer process already in use in the company, you're way ahead of the game!
- Don't worry about user resistance, especially if the response times they have elsewhere are slower than they'd like.
- Converting the SAS Code is the least of it!
- Don't expect cost reductions on paper ... it comes in other forms.
- Expect that your SAS programmers will want to reinvent a few things ... particularly to take advantages of GUI and FRAME applications.

Notes

1. See LeBouton, K.J. (1997), *The Realities of Downsizing: Moving a SAS® Application from MVS® to UNIX®*, Proceedings of the Twenty Second Annual SAS Users Group International Conference, Cary, NC: SAS Institute Inc., 658-667.

References

Please see the above-referenced paper for a listing of helpful references utilized during this project.

About the Authors

Tracy Cermack is SAS Systems Development manager for Service Engineering at American Honda Motor Company, Inc., and has 15 years of experience with SAS. Her areas of expertise include base SAS, SAS/AF, FRAME, SCL, SAS/FSP®, and SAS/GRAPH. Tracy is one of the founders of the

WUSS regional SAS user group, and was Co-chair of the inaugural WUSS conference in 1993. She has a BA in Psychology from California State University, Northridge, and an MS in Systems Management from USC.

Kim LeBouton is an independent consultant with 15 years experience with SAS. Her areas of expertise include base SAS, SAS/STAT, SAS/FSP®, and SAS/AF software. She has a BA in Psychology from California State University, Long Beach, and an MA in Educational Statistics from UCLA. Kim was WUSS '97 Co-chair, and is a SAS Institute Quality Partner™.

Tracy Cermack  
American Honda Motor Co., Inc.  
1919 Torrance Blvd., 500-2S-1B  
Torrance, CA 90501-2746  
(310) 783-3515  
TCermack@amerhonda.com

Kim LeBouton  
K.J.L. Computing  
3431 Yellowtail Drive  
Rossmoor, CA 90720  
(562) 594-9235  
Kim\_LeBouton@msn.com