ABSTRACT

Market Risk management at the enterprise level at GMAC includes data collection, stress analysis and fixed income analytics in Asset Liability management (ALM) system and Value-at-Risk in another. The different systems are rewritten in a single SAS/AF application to overcome the shortcomings of the ALM system, written in MS-Access, (capacity, robustness, complexity) and the VaR system, written in SAS, (lack of interactive functionality), while expanding the functionality of both processes.

INTRODUCTION

GMAC Enterprise Risk Services' (ERS) responsibility to report Market Value-at-Risk had expanded from the Mortgage Group to North American and International Auto Finance. The total coverage represents approximately 85% of the market risk to GMAC.

ERS collects data from the businesses, depending on whether the business involves mortgages or not. Because of the difficulty in estimating profiles for mortgage loans, we send the businesses within the Mortgage Group volatility reports which they use to produce product profiles, which are uploaded into our database. As described in my NESUG 2003 paper, our VaR is calculated using a two factor model, interest rate and spread, but the greatest risk in these profiles is the prepayment risk that are implicit in the profiles.

On the automotive finance side, the businesses produced no such profiles and we had to devise a system to derive these with minimal data. An ALM system in MS-Access was created to derive the convexity and duration and perform scenario and stress analysis on these products. While the system performed well with the initial amounts of data, its operation remained awkward at best, requiring considerable hands-on manipulation of the input feeds, constant vigilence for input error and a profound knowledge of the market risk parameters. Capacity quickly became a problem and each run required thinning out of previous periods of data from a wide range of tables, a database repair/compress and often a system reboot. Analytics tied up the machine for extended periods, without any guarantee that the results wouldn’t reveal errors in input feeds.

The greatest drawback of this system is that is requires a lot of hands on knowledge of the data being received and dexterity in manipulating both Excel and MS-Access to receive the data and produce the monthly analytics. Much time was spent on writing the operating manual, but it was easy to make mistakes or run items out of order. While less involved, the same was true for the VaR process that was written in SAS.

The VaR system had evolved to the point where one file that held all variable parameters was referenced for all of the monthly jobs; this file, however didn’t specify any order that the files had to be run, nor did it make suggestions for default parameters. We decided that the ALM and VaR should be should run under one system in SAS/AF. The goal was to embed the logic of the process into the code and eliminate the need to rely on detailed instruction manuals. Until such time as all of the extensive reporting capabilities had been replaced in the new system, the MS-Access database tables would be updated and maintained from SAS.

Three new elements to deal with are a desktop shortcut to invoke the application, an autoexec.sas and the sasv9.cfg files. The shortcut initiates the SAS session and points to the catalog that contains SAS/AF program as well as the autoexec.sas and sasv9.cfg files SAS is going to use. As the SAS/AF programs are prone to have parameters specific to the machine on which they are run, it makes sense to initiate the librefs in this autoexec.sas file, instead of the code within the program, so that all such changes can be made within one file.

Designing an interactive system requires some planning, but also offers a lot of control. Some very simple functionality is available through some very basic program logic. In this system, we often run VaR recalculating past periods. It’s easy to make a mistake and code invalid dates or dates that are in the wrong order (the end date
precedes the start date). Here we can use pick lists to prevent making mistakes.

Note that the End Date is grayed out, i.e., we cannot enter anything in the control, until a Start Date is chosen.

Once the Start Date is chosen, Start Date is grayed out and the pick list for the End Date appears which is programmed to appear with only corresponding valid values. Note that we control not only the validity of the dates, but also the navigation.

In fact, it is not only possible to gray out components, but also hide them until they are needed. This greatly helps keep the screen uncluttered and easy to follow. Components such as the ListBox control inherit the .gray() and .hide() methods from the widget class.

It pays to spend some time to design the system. Retrofitting an MS-Access or SAS process run in batch doesn't exploit the enhanced functionality SAS/AF can provide, but it is possible. As this approach provides much of the functionality we want, we start here.

The existing ALM system involved considerable interaction with Excel. Because of the flexibility within SAS and Excel, I used dde to read from and write to Excel.

On the first business day of the month we calculate the volatilities of the risk factors and distribute the resultant Excel spreadsheet to the Mortgage Group. The code behind the control push button reads the End Date and generates the Excel spreadsheet with the code to format the spreadsheet imbedded in the program.

Although there are alternative components that would work, I decided to use the listview control to keep track of the feeds that were present and successful read. This component is not included in the default list, but available under "Resources," the folder that appears just above Components. These components are to be used at your own risk, as they may not be fully tested or may work only in given host environments.

The temptation is to code the data steps to read the feeds in a single submit block, there are advantages to splitting up the code such that each file is read separately in its own submit block. Feedback from Base SAS cannot occur until the end of the submit block passes control back to SAS/AF. You can also selectively direct the log to the frame. To help keep the frame uncluttered, it is possible to include the control in a new frame or unhide a section of an existing frame.

The resultant spreadsheet resembles what was formatted by hand before.

```
put ['SELECT("R6C1,R7C8,R15C8","R15C8")'];
put ['VSCROLL(56,TRUE)'];
put ['SELECT("R6C1,R7C8,R15C8,R59C1,R76C1,R84C1,R83C1,R84C1","R84C1")'];
put ['FONT.PROPERTIES("Arial","Bold Italic",10,FALSE,FALSE,FALSE,FALSE,FALSE,1,0,FALSE)'];
put ['VSCROLL(38,TRUE)'];
put ['SELECT("R45C1")'];
put ['FONT.PROPERTIES("Arial","Bold Italic",10,FALSE,FALSE,FALSE,FALSE,FALSE,1,0,FALSE)'];
put ['SELECT("R43C8")'];
put ['VSCROLL(1,TRUE)'];
put ['SELECT("R5C1:R5C6,R5C8:R5C10","R5C8")'];
put ['PATTERNS(1,0,15,TRUE)'];
put ['SELECT("R5C4:R5C6")'];
```
libname alm "l:\ALM_.mdb" ;
proc sql ;
DELETE * from alm.Risk_Factor_Extract rfe
where rfe.valuation_date < '31jan2006:00:00:00'dt
and rfe.risk_factor_type='Interest Rate' ;
quit ;

proc sql feedback ;
connect to access as db (path="l:\ALM.mdb") ;
execute (delete rfe.* from risk_factor_extract rfe
where rfe.valuation_date < #1/31/2006#
and rfe.risk_factor_type = 'Interest Rate') by db ;
disconnect from db ;
quit ;

WHAT IS VALUE-AT-RISK?
Value-at-Risk is the maximum amount of money that may
be lost on a portfolio on average over a given period of
time, with a given confidence interval under normal
circumstances. For traders this period of time, or holding
period, can be a day or less. For our model the holding
period is one month. When we say that our VaR is -$47m
for a given company at the 95th confidence interval, we
mean that, under normal markets conditions, we expect to
lose more than $47m, 5% of the time. Note that this does
not consider catastrophic events.

VALUE-AT-RISK at GMAC
While there are various ways of calculating Value-at-Risk,
we use a two factor, interest rate and spread, model. To
help explain the mechanics of the model, I’ve illustrated an
eample involving three risk factors, three products,
running 10 simulations in Appendix A.

SELLING THE PROCESS
Although we can backtest this monthly process with
synthetic bonds, it is difficult at best to evaluate the
methodology of a one month holding period on a
performance basis. Transparency has always been the
aim in dealing with upper management. Appendix A is a
copy of a poster I keep in my office, as this 10-simulation
type index in all aspects of the VaR. Lately I’ve adopted a
more hands-on approach, creating examples in Excel to
pass on. This affords greater flexibility in the number of
scenarios and in the number of simulations.

Attempts at incorporating all aspects of a Monte Carlo
process can be cumbersome to run in a live demo and
probably involve Excel add-ons that may not be installed
or licensed. SAS/AF can greatly speed up the processing
of the examples in Excel. We have any example of
manipulating the data in Excel, which is read into SAS,
processed and written to multiple work sheets in Excel.
The formatting code is transferred from SAS and makes
the process transparent. Here the power of SAS and dde
are clearly demonstrated, as SAS is able to populate a
number of scenarios, a varying number of simulations and
write them to separate worksheets within a spreadsheet.
While the heavy lifting is done in SAS, the results, to be
distributed in Excel, are there for future reference.
CONCLUSION

Our situation is hardly unique and despite the shortcomings, development in MS-Access continues unabated. Fortunately, SAS/AF is alive and well and ready to be a viable solution. It offers the full push button functionality of others systems, while being able to deliver the power of SAS.

REFERENCES

SAS/AF

SAS®, SAS/IML®, and SAS/AF® are registered trademarks of SAS Institute Inc. in the USA and other countries.


SAS/AF® Software: Application Development I & II (manuals for Instructor-Based Training).


VaR

Gradante, CJ. The Pitfalls of Using Value-at-Risk to Measure Hedge Fund Risk at http://www.hennesseegroup.com/Featured_Articles/pitfalls.html

Gloria Mundi website: http://www.GloriaMundi.org


ACKNOWLEDGMENTS

Special thanks to Ann Rockett and Lynn Curley and others of the SAS Technical Support team, who are tireless in responding to questions. SAS’s greatest asset is in technical support and nowhere is it this contribution greater than in dealing with SAS/AF.

CONTACT INFORMATION

George Rezek
GMAC Enterprise Risk Services
8400 Normandale Lake Blvd
Bloomington, MN 55437
Work phone: (952) 857-6042
Fax: (952) 857-6688
e-mail: george.rezek@gmacers.com