

Interfacing SAS® Software, Excel, and the Intranet without SAS/Intrnet™ Software or SAS® Software for the Personal Computer

Peter N. Prause, The Hartford, Hartford CT
Charles Patridge, The Hartford, Hartford CT

Abstract

The Hartford's corporate planning system, called the Corporate Profit Model (CPM), is a SAS based multi-platform application that runs on a DEC Alpha VMS host and on a Windows NT Server. It makes heavy use of Base SAS and SAS/CONNECT® on these platforms. It also interfaces with a Visual Basic (VB) application that runs under Microsoft Excel. The VB portion of the system allows users to input assumptions into the model, and view output generated from the model in the form of Excel spreadsheets. These inputs and outputs are accessed via the company's intranet facility. The Alpha portion of the system performs the necessary number crunching, with the NT Server portion serving as the connection between the user and the Alpha host.

Introduction

The Hartford is a large insurance company that provides coverage for a host of individual and commercial customers. CPM was a system developed 30 years ago in Fortran that enabled the modeling of premiums, losses due to claims, and loss expenses resulting from payment of claims for the property/casualty insurance products. Other inputs for such things as general expenses, loss reserve changes, and reinsurance activity were loaded to produce a forecast of what is called Adjusted Underwriting Income. This legacy CPM was very difficult to maintain due to the "spaghetti" code that had developed over the years. It also required the users to input planning assumptions using manual, time consuming methods. It only produced hard copy reports. However, its internal business logic was still sound.

By the mid-90's, The Hartford's financial services portfolio had expanded to where it was now necessary to plan other sources of income, particularly investment income and servicing income, in more detail. To do this, a batch system, Integrated Financial Planning System (IFPS), was developed in SAS. IFPS sort of "wrapped around" CPM by using CPM as an input source, then taking in additional forecast assumptions for planning Net Income, Cash Flow, and Balance Sheets. Completing IFPS, our attention then shifted back to rewriting the legacy CPM and integrating the two planning systems.

The Problem

The new planning system (we decided to keep the name CPM) would still have to run on the Alpha where the horsepower was available to handle the large amount of data and calculations involved. But how could we provide a user friendly, GUI interface for our customers without installing specialized software or applications on their desktops creating a system administration nightmare? Expense was a consideration also. Management wanted to use existing software products and avoid additional costs involving new software products on user desktops.

Our actuarial, controller, and planning analysts all use Excel as a "natural language" for communicating quantitative and financial information. Since the company had standardized on WinTel, all users had Microsoft Office products and used Excel as their primary spreadsheet tool. An excellent corporate communications infrastructure existed, which connected all LANs via a backbone and provided intranet services.

The Solution

We decided our new CPM would be written in SAS and be executed on the Alpha. However, it would have to be invoked via a "front end" that could download input templates in the form of spreadsheet applications to user desktops. The front end application would have to reside on the Corporate Actuarial LAN server (Windows NT), the home of our newly developed Web site and our connection to both the corporate communication backbone and the intranet.

At the time, there were only two realistic language choices for front end development—Java and Visual Basic. Our choice was guided by the broad based support for Visual Basic and the lack of Java standardization at the time. Since we wanted to minimize impact on the customer environment and they all used MS Office, we settled on the Excel subset of Visual Basic for our macro language.

The Visual Basic code behind the front end application would validate and accept user input for the CPM run. It would then create customized input files and parameters reflecting the modeling assumptions desired. After triggering a run of CPM on the Alpha, output would be returned to the users in the form of front end spreadsheets.

Exhibit 1 is a diagram of the overall processing flow in the new CPM.

Obstacles encountered and resolved

Here are some of the critical questions that had to be dealt with in building the new CPM.

1. How can a batch SAS job on an NT Server be invoked from an Excel desktop screen?

Using the Excel toolbar, an icon was created on a spreadsheet that, when clicked, would invoke a SAS batch job on the network connected NT Server. This was done using the following steps. Pick these menu choices from the Excel toolbar: Insert, Object, Create New Package. In the resulting Object Packager Window, select Edit and then the Command Line choice. In the Command Line box, type in the network path pointing to the SAS.EXE program, with the SYSIN option and the CONFIG option specified. For example,

```
\\CORPACT1\SAS\SAS.EXE
```

```
-SYSIN  \\CORPACT1\SAS\CPMPARM.SAS
-CONFIG \\CORPACT1\SAS\CPMCONFIG.SAS
```

This is a normal way to invoke SAS, and is documented in the manual *SAS Companion for the Microsoft Windows Environment, Version 6, Second Edition* available from SAS Institute.

Next, click the Insert Icon button in the Object Packager Window, and select the icon you wish to appear on the user screen, for invoking the SAS system. Then exit the Object Packager window, and you will see the icon that, when clicked, will start up SAS. After SAS starts (as a result of the SAS.EXE portion of the command line), then the SAS program specified in the SYSIN portion will execute. You can also use the CONFIG portion to perform some automatic processing at startup time as desired.

The SAS program invoked, here called CPMPARM, executes on the NT Server and accomplishes one primary task. It reads a parameter file created by the user on the front end which tells what parameters should be used in running the CPM model. The parameter file can contain multiple records, and each record then triggers a batch run of the CPM model on the Alpha.

Exhibit 2 shows the Excel screen with the icon that the user clicks on to invoke the NT and Alpha SAS processing.

2. How can file contention between the user and the SAS NT application be avoided?

After the user invokes the SAS program from the Excel front end by clicking the icon, the program needs to access the parameter file. The possibility exists that the front end will still be updating the file and the SAS program would abort with a file in use error. The SAS program deals with this possibility by attempting to open the file with the FOPEN function. This code is in a loop so that if the return code from the function indicates that it failed, it will attempt to open again. The loop is currently set to execute 50 times if necessary. This has solved the problem for us, since the front end, if indeed updating the file, does so in a very brief batch processing window, rather than interactively which would result in keeping the file open much longer.

3. How can SAS processing on the Alpha be invoked from the NT Server?

This is easily done by using the services of SAS/CONNECT which runs on both the NT Server and the Alpha. A communications link with the Alpha is created by using the SIGNON statement in the NT Server SAS job. This causes a script file to execute to make the connection. We were able to use the script file shipped with the SAS/CONNECT software, with a customized modification to avoid the online prompt for the userid and password. For more information about SAS/CONNECT and script files, see the manual *SAS/CONNECT Software, Usage and Reference, Version 6, Second Edition* from SAS Institute.

When the CONNECT session is established, a REMOTE SUBMIT statement is issued causing the subsequent code to be shipped and executed on the Alpha. The necessary parameter file information is then PROC UPLOADED, and this is followed by a %SYSEXEC statement that submits a batch job for execution on the Alpha to run the model for that particular set of parameters. If a parameter file has multiple records for executing the model repeatedly, then multiple batch jobs would be submitted into a single threaded Alpha job queue. In this way there is no file contention between jobs. Also, the data on the Alpha is partitioned into subdirectories, one for each line of business, so if it becomes necessary to create multiple simultaneous queues, then there will still be no file contention.

4. How can users be notified of processing status, after they have clicked the icon to invoke the SAS job?

When a user clicks on the icon in the Excel screen, a tool bar icon automatically appears at the bottom of their screen with an hourglass and the beginning of the name of the batch NT SAS program. Clicking on this toolbar icon provides the user with additional information on what program is running and where the log and list files are being written. The last step in the NT SAS program is a DATA _NULL_ step with a FILE MESSAGE statement. This outputs a popup window on the users screen with customized text telling the user that all batch processing requested for CPM has been now submitted to the Alpha. The user must respond to the OK button in the popup window to continue with other desktop processing. This submit process of a variable number of Alpha jobs takes under a minute to complete.

The batch SAS jobs submitted to the Alpha each takes from 2 to 15 minutes to complete, depending on the parameters submitted. At the end of a batch job, a "done" file is created containing a date/time stamp. This done file is then read by the front end application to trigger a screen telling the user that the Alpha batch process is completed, and spreadsheet output is available for viewing. Users can query at any point to find out if the process is done.

5. How is output returned to the front end, and formatted for user viewing and printing?

When the various models (Premiums, Losses and Expenses) have completed their respective number crunching, each model will have created its own permanent SAS data sets with the appropriate calculations. With the use of several SAS macros, the contents of these SAS data sets are formatted as comma-delimited (CSV) files and shipped to a shared (Alpha and NT) directory. At that point, the front end is able to open the CSV files as Excel spreadsheets, allowing users to do as they please with the spreadsheets - copy, cut and paste, print, or save on their desktop.

Exhibit 3 shows a sample output report as it would appear in the front end application.

6. How are users given access to detail data, for further ad hoc investigation?

If a user receives unexpected results from the Corporate Profit Model, they may need to investigate underlying detail data. The ability to access data is currently provided by use of the SAS System Viewer, a free tool from SAS Institute that runs under Microsoft Windows. Using the viewer, users can access SAS data sets which in CPM represent our Premiums, Losses, and Expenses data bases. Although the access is read only, users can subset and extract data via an SQL query window. The results of the query can then be cut and pasted from the SAS Viewer window into Excel, for further manipulation and reporting by users as desired.

The SAS files with the detail data for premiums, losses, and expenses reside on the Alpha. Transparent cross-platform access to these SAS data sets from a user's desktop is possible because the SAS System Viewer makes use of the SASSFIO I/O engine. However, it is also necessary to license SAS/CONNECT and the user must be network connected directly to the Alpha. Given this setup, our users can see the Alpha SAS data sets using their Microsoft Windows Explorer, double click on them, and have the SAS System Viewer automatically invoked resulting in a spreadsheet-like display of the data. We see an approach such as this as a way to get out of the business of creating ad hoc extracts and reports.

Another type of capability was developed for the IFPS reporting detail—the underlying components of Net Income, Cash Flow, and

Balance Sheets. As with the Underwriting Income reporting, CSV files of all reports and all data element listings are created that are accessible as Excel spreadsheets. In addition, a "Create Your Own Report" capability was developed where a user may select any number of data elements and portray them for all current year and future time periods, again as an Excel spreadsheet. We are further expanding our multi-line of business reporting capabilities through Excel's extremely powerful Pivot Table functions.

7. How is documentation on the CPM system made available to the user community?

With a financial planning system such as CPM, it is important that all users have a common understanding of formulas, data items, etc. We have accomplished this by putting documentation out on the intranet for anyone to view. Using a browser (Internet Explorer or Netscape), the user can access Web pages and HTML coded documents. Included in the output facility is documentation of CPM and other downstream systems, data dictionaries (complete data dumps of production SAS data sets), text descriptions of the formulae in the various models, and some standard reports viewed using a separate Excel application invoked via HTML code.

Exhibit 4 is the Excel screen that allows any user to access the various parts of system documentation.

8. How is the system designed to reduce maintenance required to handle inevitable changes, such as new types of insurance products?

The Hartford continually strives to create products and coverages that meet our customers needs. This means that new "lines" of insurance are created, while existing lines may be split, combined, or discontinued. The legacy CPM was designed to model a maximum of 30 insurance lines, a limit that was reached many years ago. The new CPM has no restrictions on the number of lines modeled, and was designed to be flexible to meet the dynamically changing nature of new insurance products as well as new business acquisitions.

One approach used in the new system was to create table driven code and avoid the use of any hardcoded constants. For example, the current structure of our insurance lines and sublines (the components of the lines) are contained in easily maintainable tables. When a change must be made, it is made once to a "master" table that then drives on request SAS programs which create updated SAS format tables. Format tables are used liberally in many programs as the primary technique to avoid using code that would otherwise require routine program maintenance.

Another design feature was to structure the Alpha program code as a hierarchy of executable modules. There is one high level program, called the CPM Driver, which then calls (as necessary) the Premium Driver, the Loss Driver, and/or the Expense Driver. Each of these driver modules then calls multiple SAS macros that, piece by piece, perform the necessary tasks. Using a modular approach permits us to reuse the same code where ever possible and reduces our maintenance effort.

Conclusion

The CPM System is an essential tool allowing many users at the Hartford to analyze current trends and make projections into future years concerning the net income, cash flows and ROE to be potentially realized. The system was built using existing software products and without incurring extra license fees or added maintenance costs. While a total SAS solution would have been more seamless and elegant, the same end result has been

achieved without the need to install specialized software on hundreds of desktops and dozens of servers. In the future, we will be looking at the ways to increase efficiency, possibly involving the use of SAS/SHARE® and SAS Enterprise Reporter® software.

References

SAS Institute Inc. (1996) *SAS Companion for the Microsoft Windows Environment, Version 6, Second Edition*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1994) *SAS/CONNECT Software: Usage and Reference, Version 6, Second Edition*, Cary, NC: SAS Institute Inc.

Acknowledgments

We would like to thank the management of Corporate Actuarial, Planning & MIS for their support, and all the people on the CPM Reengineering Team that played a part in making this system a success.

Thanks also to fellow employee Ralph Leighton for valuable feedback and advice.

SAS, SAS/CONNECT, SAS/SHARE, and SAS Enterprise Reporter are registered trademarks or trademarks of SAS Institute, Inc. in the USA and other countries. ® indicates USA registration.

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

Contact Information

Comments, suggestions and inquiries can be directed to:

Peter N. Prause
The Hartford
Corporate Actuarial
Hartford Plaza
Hartford, CT 06115
Phone: 860-547-7484
Email:
peter.prause@thehartford.com

Charles Patridge
The Hartford
Corporate Actuarial
Hartford Plaza
Hartford, CT 06115
Phone: 860-547-6644
Email:
charles.patridge@thehartford.com

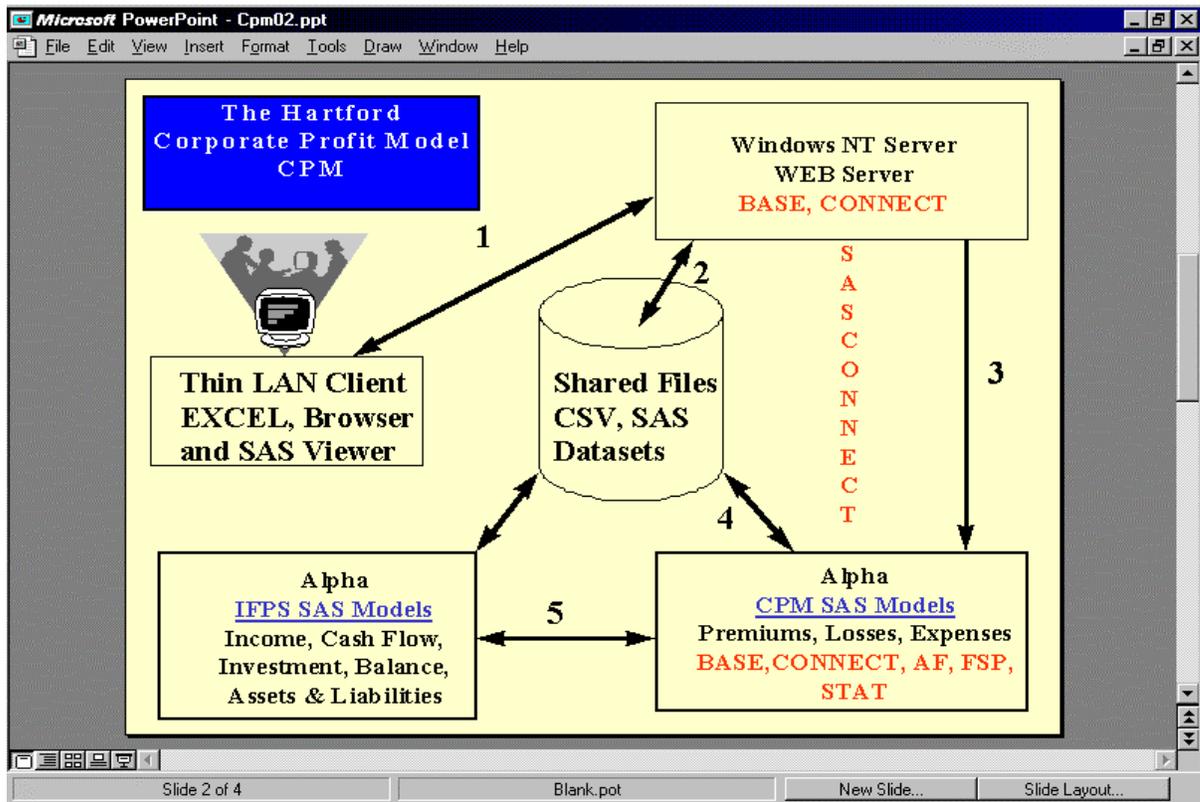


EXHIBIT 1

The spreadsheet displays a list of business lines for selection. The instructions are: "Please Place an X in the Appropriate Box for each Line of Business Requested in this Line Run".

Line of Business	Selection Box
Agency Homeowners	<input type="checkbox"/>
Agency Automobile	<input type="checkbox"/>
AARP - HO	<input type="checkbox"/>
AARP-Auto	<input type="checkbox"/>
Affinity Groups Automobile	<input type="checkbox"/>
Affinity Groups Homeowners	<input type="checkbox"/>
Insurance At Work Automobile	<input type="checkbox"/>
Insurance At Work Homeowners	<input type="checkbox"/>
Employee Insurance Automobile	<input type="checkbox"/>
Employee Insurance Homeowners	<input type="checkbox"/>
Financial Institutions Automobile	<input type="checkbox"/>
Financial Institutions Homeowners	<input type="checkbox"/>
Select Property	<input type="checkbox"/>
Select Auto	<input type="checkbox"/>
Select General Liability	<input type="checkbox"/>
Select Workers' Comp	<input type="checkbox"/>
Key Property	<input type="checkbox"/>
Key Auto	<input type="checkbox"/>
Key General Liability	<input type="checkbox"/>
Key Workers' Comp	<input type="checkbox"/>
Major Property	<input type="checkbox"/>
Major Auto	<input type="checkbox"/>
Major General Liability	<input type="checkbox"/>
Major Workers' Comp	<input type="checkbox"/>
Commercial Marine	<input type="checkbox"/>
Farm	<input type="checkbox"/>
Livestock	<input type="checkbox"/>
Crop	<input type="checkbox"/>
Crop Multi-Peril	<input type="checkbox"/>
SAID	<input type="checkbox"/>

Buttons: "Submit Line Run" and "Return to LineRun Menu".

EXHIBIT 2

Microsoft Excel - Monthuie.xls

	Jan-99	Feb-99	Mar-99	Apr-99	May-99	Jun-99	Jul-99	Aug-99	Sep-99	Oct-99	Nov-99	Dec-99
Sample Line of Insurance												
Total Written Premium	16339	15425	18596	18715	20613	24755	23467	24392	22019	24416	22802	21016
Written Premium (X98, xA/R)	18072	16558	19729	19848	21052	25368	23906	24831	22632	25202	23935	22149
Written Premium D98	-1133	-1133	-1133	-1133	-440	-613	-440	-440	-613	-786	-1133	-1133
Audits	0	0	0	0	0	0	0	0	0	0	0	0
Retros	0	0	0	0	0	0	0	0	0	0	0	0
Total Earned Premium	19641	19748	19760	19794	20571	20448	20735	20812	20710	20774	20601	20783
Earned Premium X98	20711	20818	20830	20864	20909	20969	21073	21150	21231	21478	21671	21853
Earned Premium D98	-1070	-1070	-1070	-1070	-338	-521	-338	-338	-521	-704	-1070	-1070
Total Commission \$	2393	2272	2656	2686	2908	3433	3760	3688	3089	3433	3260	3050
Direct Commission (incl D98)	2177	2074	2417	2445	2642	3114	3440	3362	2805	3118	2965	2778
Contingent Commission \$	216	198	239	241	266	319	320	326	284	315	295	272
Commission Ratio	14.13	14.73	14.28	14.35	14.11	13.87	16.02	15.12	14.03	14.06	14.30	14.51
Total Incurred Losses	12616	10889	12900	12752	12092	12617	11980	13065	13109	12845	9375	12333
Total Incurred Loss Ratio	64.23	55.14	65.28	64.43	58.78	61.70	57.78	62.77	63.30	61.83	45.51	59.34
Incurred Loss X98 XCats	9739	9828	8855	7457	7035	8045	8980	8331	7287	7952	7508	9577
Incurred Loss D98	-67	-67	-67	-67	1226	903	1226	1226	903	579	-67	-67
Incurred Loss CATS	2944	1129	4113	5363	3831	3669	1774	3508	4919	4314	1935	2823
Total Paid Losses	10900	9417	13701	14397	12981	12890	11026	13309	14001	12882	9446	11063
Total Paid Loss Ratio	55.50	47.69	69.34	72.74	63.11	63.04	53.18	63.95	67.61	62.01	45.85	53.23
Paid Losses X98	10989	9506	13790	14486	11776	12009	9821	12105	13120	12324	9535	11152
Paid Losses D98	-89	-89	-89	-89	1205	881	1205	1205	881	558	-89	-89
Paid Loss CATS	2944	1129	4113	5363	3831	3669	1774	3508	4919	4314	1935	2823
Loss Exp \$	1571	1580	1581	1584	1646	1636	1653	1665	1657	1662	1648	1661
Loss Exp Ratio	8.00	8.00	8.00	8.00	8.00	8.00	8.00	8.00	8.00	8.00	8.00	7.99
Net Income	14223.81											
Surplus	90410.43											
ROE	15.73											

EXHIBIT 3

Microsoft Excel - NEWCPM.xls [Read-Only]

Documentation
GAAP Financial Formula Listing
GAAP Financial Report Maps
Company/Business Unit/Plan Line/Subline Tables

EXHIBIT 4