

FINANCIAL MODELING: INDUSTRY-SPECIFIC APPLICATIONS, CASE STUDIES AND NEWS

User Group Meeting in Austin

We hosted our 2007 user group meeting September 18-19 following the SOA Valuation Actuary Symposium in Austin, Texas. General sessions on Tuesday afternoon were followed by a reception that evening. The meeting ended after a series of breakout sessions on Wednesday morning.

We began with presentations on new and upcoming MoSes version 6 releases, including a demonstration. Our 2007 releases, version 6.0 in July and version 6.1 in December, provide significantly improved efficiency, usability and stability compared to previous versions. We encourage MoSes users to consider upgrading to the latest version. Early in 2008, we will introduce another version of MoSes that will support Compute Cluster Server, a high-performance grid processing solution from Microsoft. For more information on this, see our article on page 2.

We also introduced RiskAgility™ in Austin. This new financial modeling and risk management framework is our long-term vision for software, including financial modeling.

Because the insurance industry faces challenges that require increased power, scalability and control, other financial modeling software providers are wrapping their existing systems in newer technology shells to extend the life cycle of their products. At Towers Perrin, we are providing similar enhancements with the introduction of MoSes support for Microsoft high-performance computing early in 2008.

With RiskAgility, however, we're going well beyond that by offering a clear technology path that will carry you into the future on a reliable, highly scalable system that offers the power, speed and control that your company needs. More information on this is available in our article on page 8.

Our general sessions concluded with a presentation of our support for principles-based approaches (PBA) in our software products, including TAS 9.50, MoSes (including TAS-MoSes) and RiskAgility™.

Breakout sessions covered a variety of topics for TAS 9.5, MoSes (including TAS-MoSes) and general actuarial subjects of interest, including "What's New in TAS-MoSes," "Modeling Equity-Indexed Products" and "High-Performance Computing," to name a few.

FEEDBACK

- "The live example of porting from [MoSes] v5.2 to v6 was very informative. It's also nice to see the workings and navigation of v6 in general."
- "I like your approach of building RiskAgility from the ground up instead of wrapping a platform around MoSes."
- "Good, practical information/ideas on a topic [TAS-MoSes] that applies to a task I do regularly. Concrete info."
- "Breakout sessions were particularly valuable. Only wish I could have gone to more than three of them...excellent meeting!"

We encourage you to attend next year's user group meeting. Stay tuned for our announcement in early 2008. ■

IN THIS ISSUE

User Group Meeting in Austin	1
New Grid Solution for MoSes Version 6	2
Improved Debugging Features in MoSes Version 6	3
A New Approach to Building Nested Projections in MoSes	5
Recent User Q&As	7
Financial Modeling Software for Today and Tomorrow	8

Microsoft Compute Cluster Server 2003 is an integrated, out-of-the-box solution for high-performance computing.

New Grid Solution for MoSes Version 6

Towers Perrin has teamed with Microsoft to enable our clients to use the Windows Compute Cluster Server 2003 (CCS) with MoSes Version 6. We expect this option to be available sometime in the first quarter of 2008.

Microsoft CCS 2003 is an integrated, out-of-the-box solution for high-performance computing. It consists of Windows Server

INSURANCE INDUSTRY TOP PRIORITIES

- Run more complex financial calculations faster.
- Improve customer satisfaction with faster response times.
- Improve cost management through operational efficiencies.
- Foster efficient collaboration among actuaries, managers and the compliance team.
- Protect intellectual property within financial models while sharing information.

2003 — Compute Cluster edition and the Compute Cluster Pack. Windows Server 2003 is secure, reliable and tested server software that includes support for high-performance hardware with high-speed interconnects. The Compute Cluster Pack includes a job scheduler, resource manager, cluster management and message passing. Together, these capabilities address the insurance industry's top priorities by making grid cluster operation as easy and secure as a single system while leveraging customer investment in Windows software administration and tools.

MoSes v6 will be able to distribute financial projections using a CCS capability called "parametric sweeps," which runs the same program across clusters using different input and output files, as follows:

- MoSes will split the input file by iteration or in-force file group into multiple input files, where the file name consists of the same stem name with incrementing extension numbers.
- The CCS Job Manager is set up to create separate jobs for a particular MoSes model and the set of input and output files specified by the stem names.
- CCS distributes these jobs across the cluster and monitors them as they run and produce separate output files.

- Once all of the distributed jobs are finished, MoSes will combine the output files into one result set for analysis and reporting.

By splitting the work into smaller jobs and running them in parallel, MoSes projections will finish in much less time.

While MoSes supports a number of different grid solutions, Microsoft CCS is the preferred solution for MoSes v6. Furthermore, we are engineering RiskAgility, our new financial modeling solution, with CCS integration. This means that CCS can provide a transition path to the future: Your investments to use CCS with MoSes v6 will provide value with RiskAgility as well.

For more information about MoSes v6 and Microsoft CCS, please contact your TSS account manager or send an e-mail to software.solutions@towersperrin.com. ■

The Microsoft Compiler brings several benefits, such as more robust syntax checking, single-pass error detection, access to the Microsoft Debugger and the Compile Error Viewer.

Improved Debugging Features in MoSes Version 6

MoSes version 6 includes some new features to help modelers debug their applications. Here's what they are and how they can help you.

THE NEW MICROSOFT COMPILER

In MoSes version 5.1, we introduced the option to be able to use a Microsoft C++ compiler instead of the Borland C++ compiler used in all previous MoSes releases. In v6, the Borland compiler is no longer used, and we now distribute the Microsoft 2005 compiler free with the MoSes system.

From a debugging perspective, the Microsoft compiler brings several benefits, such as more robust syntax checking, single-pass error detection, access to the Microsoft Debugger and the Compile Error Viewer.

More Robust Syntax Checking

Microsoft compilers perform more strict syntax checking than the Borland compiler and will detect many coding errors that the Borland compiler would have overlooked, including the following:

- uninitialized local variables
- comparisons between incompatible types
- accidental assignments in “if” statements, for example if “(x = y)” instead of “if (x == y).”

Single-Pass Error Detection

During compilation, all compile errors are listed from all submodels. The Borland compiler displayed only the errors in the first model. With the Microsoft compiler, you can now resolve all errors in a single pass, rather than having to fix and recompile in stages, as was the case with Borland.

Access to the Microsoft Debugger

Using a Microsoft compiler allows us to use a Microsoft Debugger, which is described below.

Compile Error Viewer

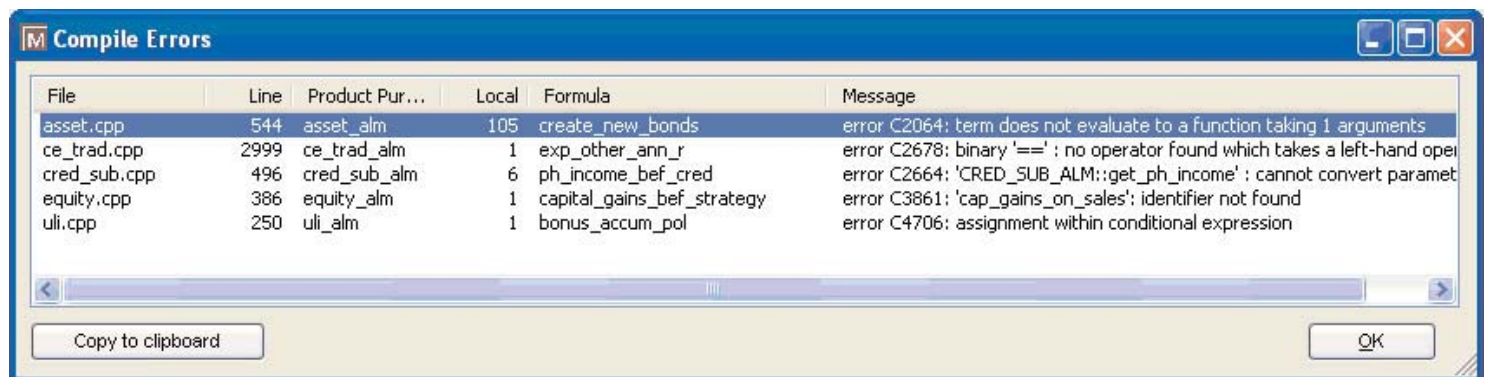
The MoSes Compile Error Viewer has been updated to make diagnosis easier if your code won't compile. For example, we've enhanced the viewer to show the local line numbers within MoSes formulas where compile errors exist (see *Figure 1*). These line numbers have been incorporated into an enhanced formula editing panel.

We've also added a “Copy to clipboard” button, which allows you to copy the list of errors and paste them into another program such as Excel or WordPad. This gives you an easier way to manage a large list of compile errors, rather than repeatedly opening and closing the Compile Error Viewer.

MICROSOFT DEBUGGER

The Microsoft 2005 Express C++ Debugger is available free of charge from the Microsoft Web site. Instructions on how to download and install the Debugger are included on your *(continued on next page)*

FIGURE 1



Improved Debugging Features in MoSes Version 6

(continued from page 3)

MoSes Version 6 CD. The Microsoft Debugger provides powerful functionality to help you manage development of even the most complex applications.

User-Friendly Interface

The Microsoft Debugger provides a user-friendly environment for debugging models. The Debugger displays each module of code as a separate tab and provides a large number of debugging tools made available through dockable windows that can be manipulated to your own layout preferences. This is a vast improvement over earlier MoSes debuggers.

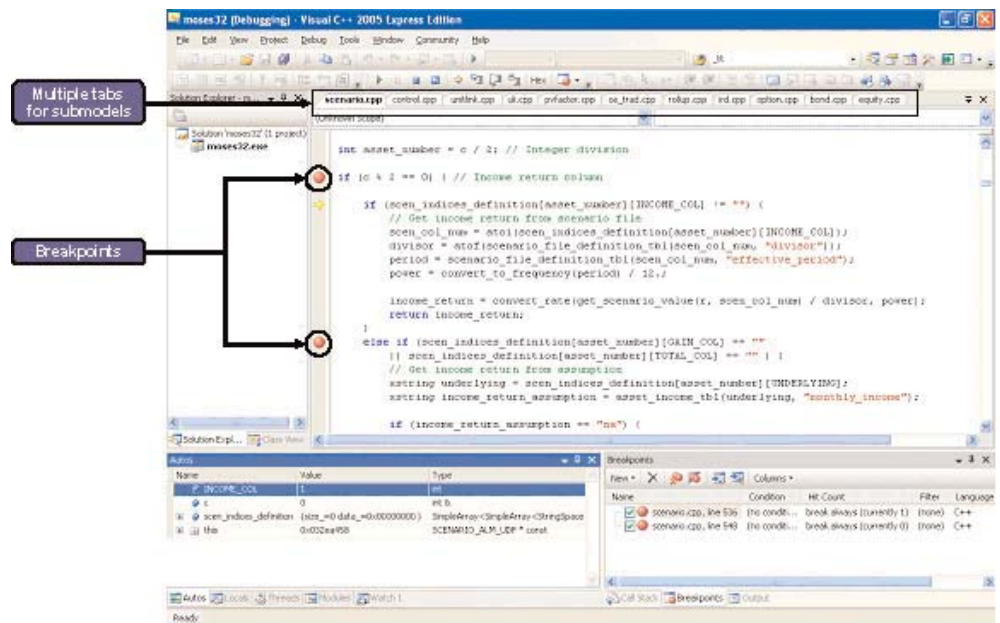
Using Breakpoints

A breakpoint is a temporary marker that will suspend your MoSes projection at the location where it is set. A breakpoint can be set at any point in the code, and any number of breakpoints can be set within an application.

At each breakpoint, you can inspect the values of formulas and variables that make up a calculation. This can be done by hovering over them with your mouse pointer or by using the paneled windows.

Alternatively, you can choose to step through your code line by line. This allows you to examine the precise path taken by an application and help you quickly find the cause of any unexpected behavior (see *Figure 2*).

FIGURE 2



You can also set conditional breakpoints to suspend your projection based on criteria that you set. Examples include setting breakpoints at particular time periods or when the value of a variable is negative or exceeds a certain value.

Capturing MoSes Exceptions

Rather than suspending your run at a preset breakpoint, the debugger can also stop a projection at a point where a MoSes exception occurs. You can use this to identify the cause of nonfatal errors (such as model skips caused by divide-by-zero errors or access violations) or fatal run-time errors.

OTHER DEBUGGING FEATURES

In addition to the features covered above, some further enhancements in v6 include the following:

- a completely redeveloped Dependency View in MoSes v6.1
- more useful error messages for dealing with run-time errors, such as out-of-bounds errors when using SmartArrays
- a more tightly integrated Snip test, which allows the user to check for correct syntax at the individual formula level.

For more information on the many debugging features, see the “Model Debugging” section of the MoSes Online Help. ■

Cloning provides an easy way to use the same formulas under a different set of assumptions.

A New Approach to Building Nested Projections in MoSes

MoSes has long had a built-in capability to run nested projections using a clone sub-model feature. Recently, we have developed and tested another approach to nested projections that uses less computer memory. Read on to learn the basic workings, pros and cons of both approaches.

WHY USE NESTED PROJECTIONS?

A nested projection is a projection within a projection. The primary projection, typically called the experience projection, is stopped at certain points in time so that a secondary nested projection can be run. The nested projection typically uses different methodology or assumptions from the experience projection. When the nested projection finishes, results are returned to the experience projection, which then continues on its way (see *Figure 1*).

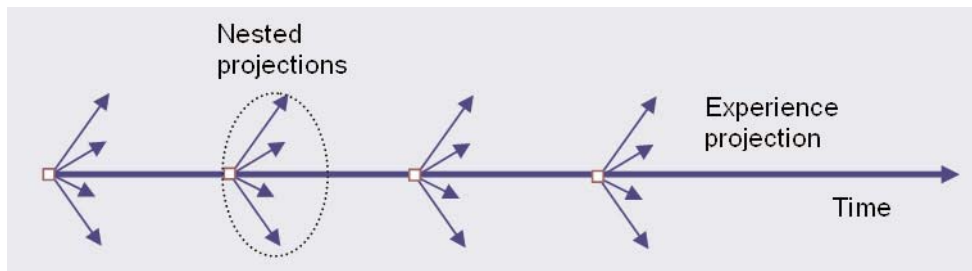
This approach is used in a wide variety of financial modeling applications (see *Figure 2*).

FIGURE 2: APPLICATIONS THAT USE NESTED PROJECTIONS

- Calculating pricing for variable annuities subject to RBC C-3 Phase II
- Calculating CARVM reserves for deferred annuities
- Calculating market value and net yields for fixed-income assets
- Performing dynamic GAAP unlocking
- Calculating SOP 03-1 GMDB reserves
- Projecting principles-based reserves and capital

Consider, for example, a model that prices variable annuities subject to RBC C-3 Phase II capital requirements. The pricing actuary may want to project capital at each future point in time in the experience projection. This requires nested calculations using a range of C-3 Phase II-compliant scenarios before continuing to the next time step in the experience projection. In this example, scenarios used in the experience projection are different from those used in the nested projection. Otherwise, the experience and nested projection calculations are similar.

FIGURE 1: TIMELINE WITH NESTED PROJECTIONS



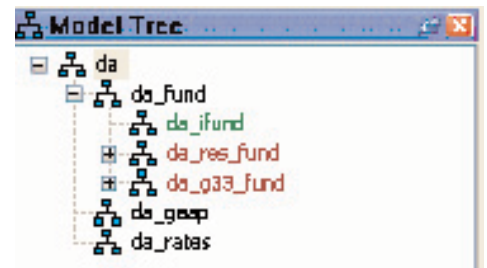
CLONING APPROACH TO NESTED PROJECTIONS

Writing computer programs to perform both experience and nested projections could involve a lot of duplicated programming logic. Often, the code for these projection types is similar, with the nested projection using the same calculations with different assumptions and methodology options from the experience projection.

One way to avoid unnecessary duplication in an application is to use cloning. A clone submodel is a copy of a parent model and its related submodels. It provides an easy way to use the same formulas under a different set of assumptions.

Figure 3 shows part of the model tree for the deferred annuity model in the TAS-MoSes application. In this example, the “da_res_fund” and “da_g33_fund” submodels are clones of da_fund. These clones were created by adding submodels in MoSes with the same product/purpose as the “da_fund” parent submodel.

FIGURE 3: EXAMPLE OF A MODEL STRUCTURE WITH CLONES



(continued on next page)

Our tests show that MoSes models built with this non-clone approach can process projections with much larger numbers of data and a greater number of scenarios than with the clone approach.

A New Approach to Building Nested Projections in MoSes

(continued from page 5)

MoSes has standard features to support cloning, including the following:

- *Projection task property* to allow cloning
- *Peer submodel structure* to associate a clone submodel with its parent submodel
- *Rebase functions* to prepare clone submodels to start a new nested projection.

These built-in features support easier development of nested projections. For example, MoSes will automatically determine which clones need to be run and which calculations within the clone submodels need to be run. The primary disadvantage of using clones for nested projections is that they may require significant computer resources to process large models with large amounts of data and scenarios.

In addition, it is possible to design a model structure that is so complex that MoSes will not automatically support cloning. In such cases, modelers either must provide explicit coding to tell MoSes how to work with these complex structures in a clone environment or use a non-clone approach to modeling.

NON-CLONE APPROACH TO NESTED PROJECTIONS

Because of the resource requirements to process MoSes models that use cloning, we developed and tested an alternative approach that does not use clone submodels. This approach makes use of the same rebase functions that are used in the cloned approach. The difference in this new non-clone

approach is that both the experience and nested projections share the same model/submodel structure. Here are the steps followed in this approach:

1. Start the experience projection.
2. When it's time to run a nested projection, save the experience model/submodel state.
3. Rebase.
4. Run the nested projection and save the results.
5. Rebase.
6. Restore the saved model/submodel state from the experience projection.
7. Incorporate the results of the nested projection and continue the experience projection.

Here are a few considerations that you need to understand when using this non-clone approach.

- By default, MoSes assumes that non-clone submodels will not use rebasing. However, any submodel can be rebased by setting the "Is Rebase Submodel" to "Yes." You can use this feature to rebase the same submodel within both an experience and nested projection that you manage within your application code.
- With rebasing, you can save and restore the submodel state before and after a nested projection. When you rebase a submodel, MoSes retains column results based upon the "RebaseType" property.
- Before rebasing for a nested projection, critical columns must be calculated up to the current period. You must create and set a "flag" variable to manage calculations, methodologies and assumptions within the nested projection. You can manage this by using the values of this "flag" variable and the application "rebase_period" variable.

- Calculated arrays are not impacted by rebasing and remain populated with current values following a rebase. You must manually save the values in "SmartArrays" before a nested projection and restore these values after the nested projection. When doing this, you will need to allocate space and use "STL Copy" to copy the contents of the array and save memory. The same is true for temporary tables. We added special functions in MoSes version 6.02 for saving and restoring temporary tables, "temp_table.save()" and "temp_table.restore()."

EVALUATION OF THE NON-CLONE APPROACH

Our tests show that MoSes models built with this non-clone approach can process projections with much larger numbers of data and a greater number of scenarios than with the clone approach. In essence, you can process about as many model points and scenarios with the non-clone approach as you could if the model was not performing nested projections at all. The non-clone approach can be a big memory saver. (It has negligible impact on runtime, however, as the same calculations would be performed with either approach.)

These memory savings come at a cost of additional coding required to manage events that the MoSes cloning features handle automatically. If you have to run large projections, however, the added scalability of the non-clone approach may be well worth the additional coding required to set it up.

For more information about this new approach to nested projections with MoSes, please contact your Tillinghast consultant or send e-mail software.solutions@towersperrin.com. ■

RECENT USER Q&AS

Q) I coded a MoSes application under MoSes 5.3. I migrated it into MoSes V6 and it worked fine initially. Today I found a legacy submodel when I was using the category tree. I deleted it, and the model does not compile anymore. The data conversion task does not work either. It worked fine under both 5.2.2 and 5.3, and I did not make any changes.

A) Please do a rebuild to force a full compile and generate. Second, the data conversion task is a bit different in MoSes V6; you now have to select “Task Inspector Panel” to see the data.

Q) I have just had my PC upgraded, and IT just installed MoSes on it. When I run the MoSes projection model, I get the following message: Fatal Error: Invalid parameter file, missing product ‘xxx’, purpose ‘yyy’.

A) This is a known problem caused by a conflict between MoSes and the Microsoft .NET framework 2.0. It has been resolved in MoSes V5.2 SP2. Meanwhile, the DotNet20Fix.exe patch that we provide

will resolve the conflict by redirecting MoSes to use the previous version of the Microsoft .NET framework.

Q) We are using MoSes 5.2.2 as our production version of MoSes and have also installed MoSes 6 for testing and conversion purposes. Is there a way to get MoSes 6 to look at the same database of user IDs that is currently being used by MoSes 5.2.2?

A) We have a FoxPro utility that can be used to copy the user settings from one machine and install to another. This will save you from having to recreate users, user groups and so forth in the new MoSes install. To copy these files, go to the MoSes Utilities directory (e.g., c:\vismoses\utilities\support) and run the usrdbf32x.exe program. This will produce a dialog box asking you to specify the source and destination MoSes directories. Select “Continue” at the bottom of the screen to update the destination settings with the source settings.

Q) My projection task now produces an output file greater than 2GB that cannot be opened in FoxPro. What can I do to get around this issue?

A) FoxPro limits model output files to 2GB. However, there are several different approaches to reducing the size of a MoSes column output file:

- In MoSes versions prior to v6, set the “Create Submodel File” projection task property to “Model Only” output to produce a separate output file for each submodel, rather than a single file containing all output. Model Only output is always produced in v6.

- Use filter sets and column output “From” and “To” settings to limit the number of columns and time periods for which results are output.

- Adjust your data/model point groupings to use fewer groups.

- There is also a facility within the Column Reorder feature in the “Projection Filter Set” view to split a file into subfiles. This is particularly useful if a single model file is heading toward the 2GB file size limit. ■



Financial Modeling Software for Today and Tomorrow

The latest releases of MoSes, versions 6 and 6.1, deliver important performance and usability enhancements. Our new quality management system and processes have helped us to deliver these releases on time and with a high level of quality and stability. And we're introducing a new version of MoSes in early 2008 that will support Compute Cluster Server, Microsoft's high-performance grid computing solution. We encourage you to take advantage of the

improved stability, quality, efficiency and ease of use that these latest versions of MoSes offer.

For TAS 9.50 users, we will provide an update in early 2008 that will include C-3 Phase III PBA capital for life insurance products.

But we're not stopping there. To meet even greater financial modeling demands in the future, we are developing the RiskAgility Financial Modeling Suite, which represents a new generation of software. While the previous generation of software systems was designed for desktop and work group computing, RiskAgility is engineered with the power, speed and control needed for enterprise financial modeling.

We introduced RiskAgility to our clients in London on November 5, 2007. The first application suite, due in the second quarter of 2008, will be for property and casualty insurers, and we will introduce life insurance applications later in 2008. ■

ABOUT TOWERS PERRIN

Towers Perrin is a global professional services firm that helps organizations improve performance through effective people, risk and financial management. The firm provides innovative solutions in the areas of human capital strategy, design and management. It also provides solutions in the areas of risk and capital management, reinsurance intermediary services and actuarial consulting. The Tillinghast practice of Towers Perrin provides consulting to the insurance industry.

Towers Perrin has offices and business partner locations in the United States, Canada, Europe, Asia, Latin America, South Africa, Australia and New Zealand. More information about Towers Perrin is available at www.towersperrin.com.

Financial Modeling Update — Americas Edition

Jan Hayes, editor, jan.hayes@towersperrin.com

©2007 Towers Perrin. All rights reserved.

TAS Tillinghast Actuarial Software™ and MoSes™ are trademarks of Towers Perrin.