Mo.net
Introducing the Mo.net Excel Model Adapter

January 2024
Revision 10

Background

Despite concerns regarding the problematic convenience of spreadsheets and other end-user developed applications, such solutions continue to provide a key element of the modern actuarial technology toolkit. Spreadsheets in particular provide an almost perfect combination of flexibility, ease of use and power to solve a huge variety of business problems. Whether a spreadsheet is the most appropriate technology to solve each specific problem is an entirely different question.

Where spreadsheets do offer unmatched capability is in the prototyping of new calculations & analysis. Unfortunately, these prototype solutions often find their way into production use without necessarily having an appropriate level of control or governance applied. Since there is arguably always something more important or urgent to do, retrospective reviews or documentation of spreadsheet-based solutions are rarely, if ever performed. As a result, a tangle of unmanaged, poorly controlled, undocumented and untested spreadsheets and other end-user solutions quickly develops in most actuarial & risk functions. The number of user-developed solutions can easily run into the thousands, and become one of the most significant operational risk burdens for these functions, and the insurance company as a whole.

What is the Mo.net Excel Model Adapter?

The Mo.net Excel Model Adapter ("EMA") allows any existing Excel-based workbook to be converted into an operational Mo.net project. Once migrated, the project then benefits from the other best in class features of the Mo.net platform, such as control, performance, transparency, flexibility and enterprise integration.

The EMA converts all elements of the source workbook content, including data, functions and code modules (VBA) into corresponding & clearly segregated Mo.net components – models, inputs, parameters, functions, tables and tasks. The user has the choice to convert an entire workbook, a specific set of elements (cells, functions, ranges, etc), or only the elements required to derive a particular set of results. This allows source spreadsheet-based models containing redundant data or calculations to be simplified as part of the conversion process. With all required elements of the source workbook converted into a Mo.net project, the user can use the best-in-class interactive debugging features of the platform to satisfy themselves that the project is working as expected, before using the converted model in the operational environment, or as a basis for further development / refinement.

How Does it Work?

The EMA takes advantage of Excel and Mo.net sharing the same technology core - Microsoft's .NET Framework. This allows Mo.net to replicate the vast majority of Excel functionality and mimic the ability to recalculate results on demand without resorting to custom engineering. Furthermore, for spreadsheet models that rely on user defined functions & modules written in Visual Basic for Applications ("VBA"), these can usually be migrated to Mo.net (which uses VB.Net) with ease.

EMA provides three different migration strategies, providing the user with a choice over which is the most appropriate to use, based on the nature of the source workbook and the requirements of the target Mo.net project. Each of the strategies is described in Table 1 below.

Approach	**Description**
Interactive (Manual)	With this approach the user interactively selects each element of the source workbook – cell, range of cells, or function - and specifies the target Mo.net object type, default value and other metadata. Each element for conversion is added to a migration manifest which is then converted to a new Mo.net project by the EMA conversion engine.
Target-based	This semi-automated approach allows the user to select the results / outcomes of interest in a source workbook model and allow the EMA conversion engine to determine all precedent cells, ranges and functions. Each precedent is added to the migration manifest for review by the user before being converted to a new Mo.net project by the EMA conversion engine. Additional elements can be added to the manifest by hand if required. This strategy allows the source model to be simplified and stripped of any unnecessary content, thereby making the maintenance of the target Mo.net project more straightforward.
Complete (Brute Force)	The final approach is to convert (or attempt to convert) every element of a source model first into a Mo.net equivalent via an XML version of the workbook. This strategy brings across everything in the source workbook, whether it's of value / interest or not. As a result, any target Mo.net project is likely to contain extraneous elements which should be removed post-conversion.

Table 1 - EMA Migration Strategies

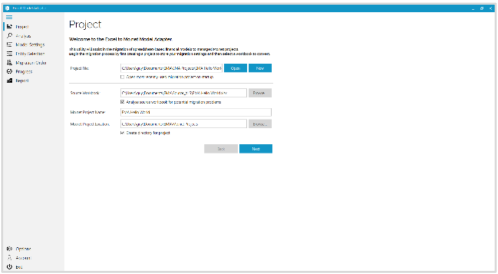
While the Excel Model Adapter will convert a huge number of spreadsheet-based models, the sheer variety & flexibility of the spreadsheet platforms means that some particularly complex source workbooks may not convert entirely. However, the EMA's migration engine will still perform the majority of the "heavy lifting" and leave the user to refine the target Mo.net solution as required.

Migration Walkthrough

To help illustrate the ease and power of the Excel Model Adapter, a walkthrough of the key migration steps is shown below.

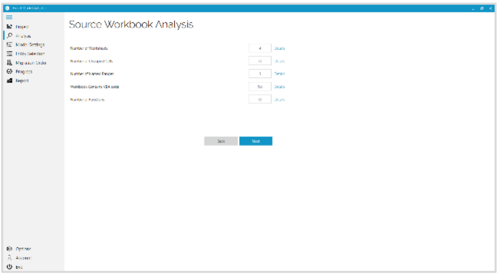
1. Select Source Workbook

The migration process starts by selecting the source workbook and the location & name of the target Mo.net project.



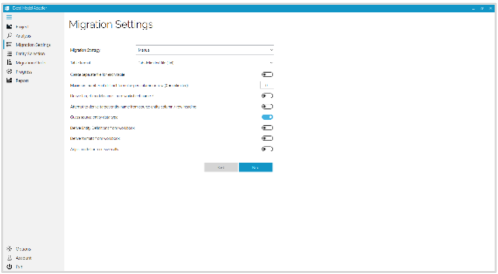
2. Analyse Workbook for Potential Migration Problems

The source workbook is now analysed to determine its key properties & dimensions – size, number of populated cells, number of functions, etc. Any workbook features that might present a migration challenge are also highlighted.



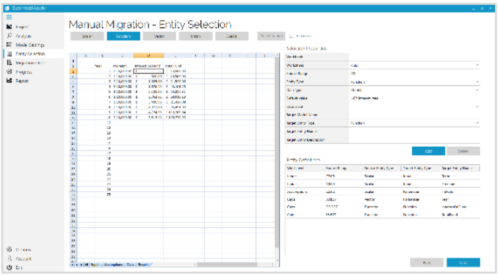
3. Select Migration Strategy & Options

Based on the workbook analysis and the user's own knowledge of the source workbook, the most appropriate migration strategy & other options are then selected.



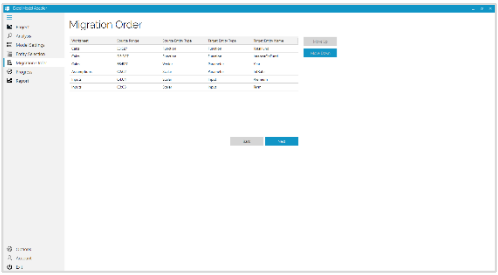
4. Select Source Entities and Define Target State Mo.net Objects

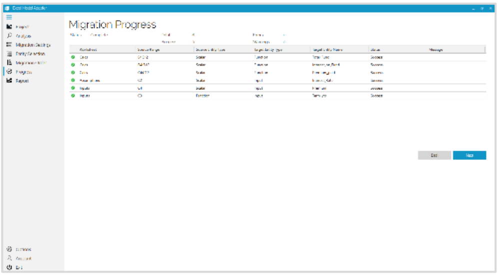
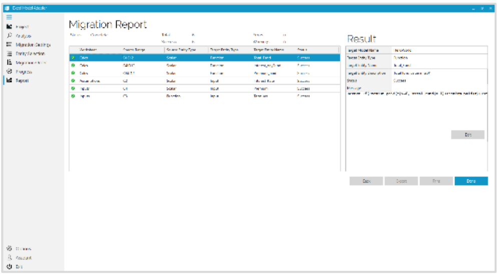
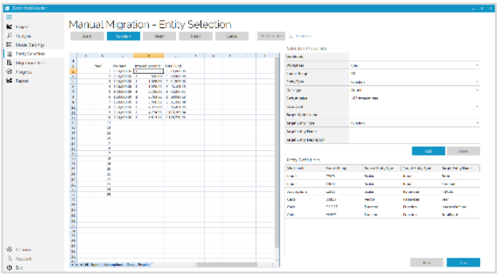
For interactive / manual migrations the user now selects each source workbook entity and defines the corresponding Mo.net target properties – type, format, default value, etc. Each entity is added to the migration manifest.



5. Confirm Migration Order

For complex workbooks & projects and to avoid potential referential integrity issues during the migration, the user can optionally choose the order in which the entities are converted.



6. Convert Workbook to Mo.net Project Once all entities have been added to the migration manifest, a trial migration can be performed. This usually takes a few seconds to complete.	
7. Review Results & Test Mo.net Project After the migration has been performed, the user should first check the migration report for any problems. The user can then open the Mo.net project to check that all entities have come across successfully, the project compiles, and that the results match the original workbook.	
8. Refine Migration Options & Entity Definitions For all but the simplest spreadsheet models, the migration project will need a few cycles of refinement to ensure all target Mo.net entities are correctly defined. This refinement process is the same regardless of original migration strategy.	

How Can the Mo.net Excel Model Adapter Help Me?

It is important to note that we are not advocating the use of the EMA simply to convert spreadsheet models to Mo.net and move inherent problems from one technology to another. What the EMA does provide however is a mechanism for placing well understood, relatively static spreadsheet-based models under more stringent control, Once migrated, the resulting Mo.net models can then be used as a basis for further model development and be consumed, as-a-service using the performance & integration potential of the Mo.net platform.

Contact Us

To find out more about the Mo.net Financial Modelling Platform, the Mo.net Excel Model Adapter, and to arrange a demonstration, please get in touch.

Software Alliance Limited
30 Stamford Street, London, SE1 9LQ
Tel: +44 (0) 20 3964 2755
www.softwarealliance.net
hello@softwarealliance.net

Author: Guy Shepherd