

Microsoft Excel Extensions for Enterprise Architect

Microsoft Excel Extensions for Enterprise Architect	1
Disclaimer	2
Dependencies	2
Overview	2
Installation.....	4
Verifying the installation	4
Base templates	7
Template1	7
Templates 2 and 3	9
Template 4	10
Exporting model data into Excel.....	11
Re-importing model data from Excel	16
Refreshing Excel data from EA	18
Support and contact information.....	21

Disclaimer

Version 1.0.x of the *Excel Extensions* add-in has been successfully deployed with EA version 9.1 or later.

There is no guarantee that versions prior to EA 9.1 will work properly. No effort will be made to support earlier releases of Enterprise Architect.

If any problems are encountered with the installation or operation of this software please [contact us](#) through any of the channels listed at the bottom of this document.

Dependencies

The add-in depends on the following components being installed on the system:

- Interop.EA.dll (part of the standard Sparx installation files).
- Microsoft .Net Framework 4.0 client profile (if missing, the installer will detect this and offer to automatically download and install the Framework).

Overview

The 1.0.x version of the *Excel Extensions* augments the built-in CSV import/export capabilities of Enterprise Architect as follows:

- Direct update of Microsoft Excel (2010 or later) files.
 - This removes the need to convert to and from CSV files.
- Template based export to Excel.
 - A sample of Excel template files is provided as part of the add-in, to be used as a starting point for exporting User data.
 - The End User can create additional templates files as needed and expose them to the add-in.
 - The columns to be exported are defined (in row 0) of each template worksheet. Multiple worksheets can be exported into.
 - All standard EA element properties can be exported, as well as any Tagged Value properties.
 - EA related columns can be mixed with User defined columns (these will not be affected by the export process, except where formula-based cells are recalculated after the export).
 - Specific starting and ending rows can be defined into which the exported elements are inserted. All other rows (outside the specified range) are then unaffected by the export.
 - This specifically allows formulas to be defined in the worksheet which can operate against the row values exported from EA!
 - The elements selected for export can be based on:
 - A single Package.
 - A Package hierarchy.

- Selected elements in the Project Browser or diagram.
- Elements resulting from a defined search.
 - If a SQL-based search definition is selected, the CLASSGUID and CLASSTYPE fields must be returned (as [documented](#) by Sparx) in the result set.
- Re-import of previously exported Excel files.
 - As long as the element GUID column is present in the Excel worksheet, the other EA related column values will be imported into EA, updating any current values.
 - Rows without a GUID value are ignored.
 - If a GUID cannot be resolved in the repository where the re-import is occurring the related row values are ignored (i.e. this re-import facility cannot be used to create new elements! Sparx already provides a creation feature through the standard CSV import).
 - Values imported into standard EA properties with defined (enumerated) value sets, such as Priority, Complexity, Status, etc. will be validated for compliance, and incorrect values rejected.
 - Tagged values specified in the worksheet will be uploaded into the repository.
 - Caveat: no attempt is made to validate the values loaded from Excel against any data constraints expressed by a Profile. It is the End User's responsibility to ensure the integrity of the supplied Tagged Value data.
 - Tags not previously existing in the model can be automatically added to the model.
- Refresh of previously exported Excel files.
 - As long as the element GUID column is present in the Excel worksheet, the other EA related column values will be updated with the current repository data.
 - Rows without a GUID value are ignored.
 - An option can be set to preserve the formulas of any cell belonging to a Tagged Value Column. This allows Excel to recalculate the values of these cells based on the formulas, perhaps for later upload into EA.

Installation

Exit any running instances of Enterprise Architect, then launch the “setup.exe” program provided by Cephass Consulting and follow the on-screen instructions.

The installation will attempt to update the Windows registry, so the User needs to ensure that s/he has sufficient privileges to run the setup program.

The recommended install path is to place the DLL and any supporting files in an “Addins” folder in the Sparx Systems installation directory, e.g.

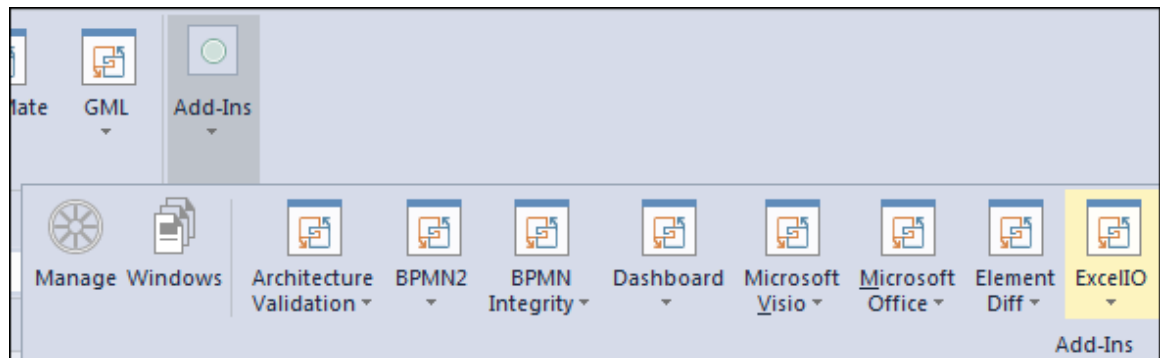
C:\Program Files (x86)\Sparx Systems\Addins.

Note that older versions of the software are automatically removed and replaced.

Should the installation fail for any reason other than insufficient User privileges, please take appropriate screenshots and email the data to the [support](#) address listed at the bottom of this document.

Verifying the installation

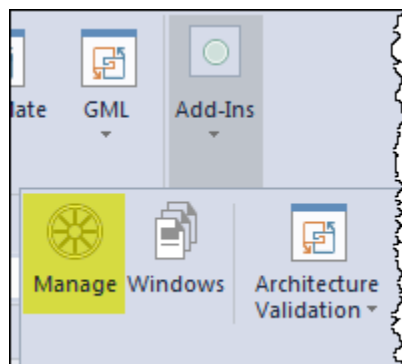
Bring up Enterprise Architect and verify that there is a “ExcellIO” item in the SPECIALIZE ribbon (EA 14.x):



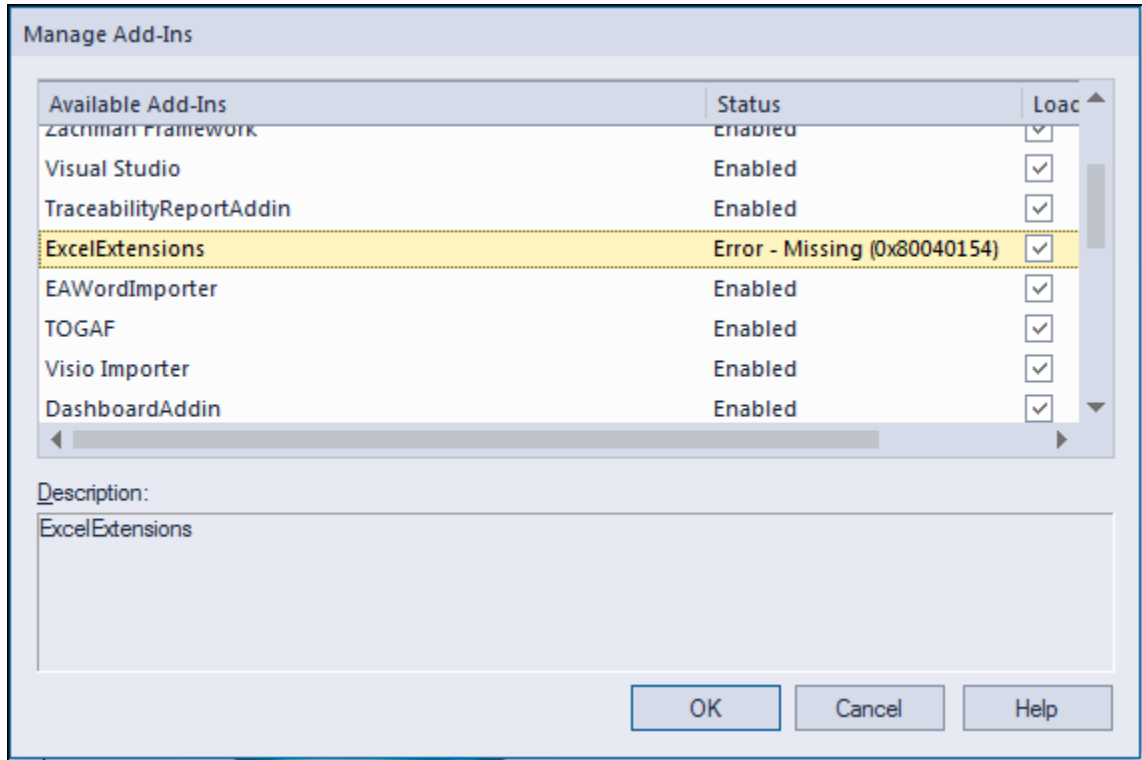
Note that the other extensions shown in the above screenshot may or may not be present as well, depending on your Enterprise Architect version and configuration.

The content of the “ExcellIO” sub-menu will vary depending on the selected context in EA (Package, element, diagram,...).

Should the menu entry not be present, select the “Manage” ribbon item:



And confirm that the “ExcelExtensions” add-in is loaded and enabled:



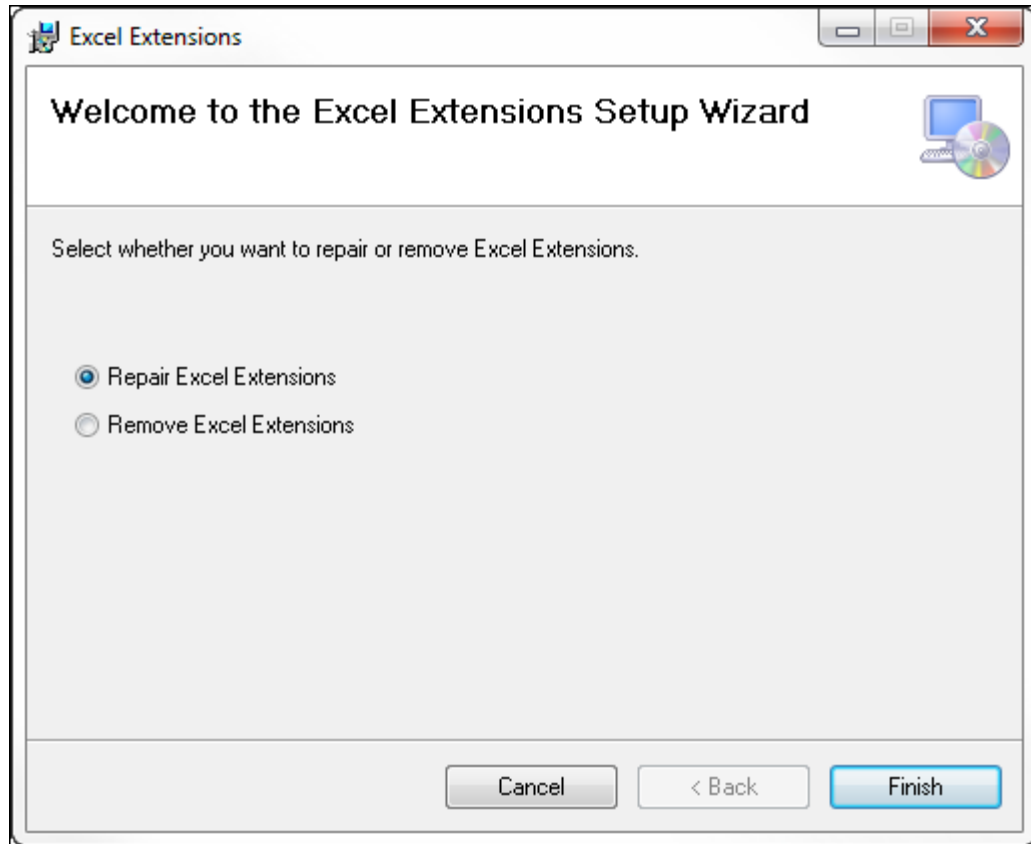
If an error status is shown, as in the example above, this typically means that either:

- The DLL installation process failed and EA cannot locate its reference, either in the Windows registry or in the file system.
- The installation did succeed but the DLL file was later moved or deleted.

If the “ExcelExtensions” entry itself is not found then the add-in installation did not complete successfully.

To fix an incorrect installation:

- Exit out of all instances of Enterprise Architect.
- Launch the setup process again. The installer will automatically provide a repair option:



If, after the repair procedure, the *ExcelExtensions* add-in is still not loaded correctly in Enterprise Architect, remove the program through the Windows control panel and start the installation process over.

At a minimum the following files should have been installed in the selected installation directory:

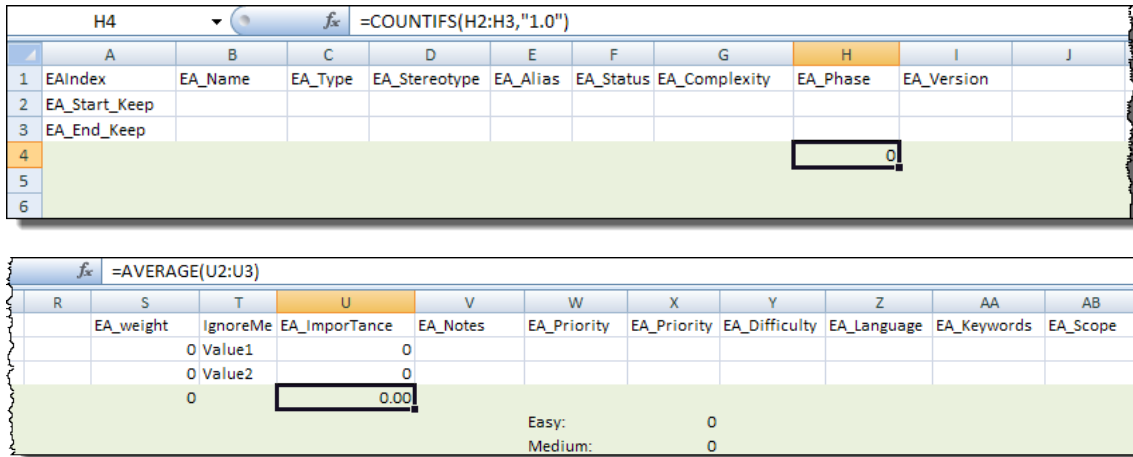
- Cephass_Software_EULA.pdf
- Excel Extensions.pdf (this file)
- Interop.EA.dll
- ExcelExtensions.dll

Base templates

Four Excel template files (named “Template<n>” where n is 1 to 4) are provided as part of the installation, in a sub-directory named “Excel Templates” of the installation folder (typically “C:\ Program Files (x86)\Sparx Systems\Addins\Excel Templates”).

The intent behind these templates is to use them as starting points for building your own set of templates. Once they are no longer useful simply delete them from the installation folder.

Template1



	A	B	C	D	E	F	G	H	I	J
1	EAIndex	EA_Name	EA_Type	EA_Stereotype	EA_Alias	EA_Status	EA_Complexity	EA_Phase	EA_Version	
2	EA_Start_Keep									
3	EA_End_Keep									
4								0		
5										
6										

	R	S	T	U	V	W	X	Y	Z	AA	AB
	EA_weight	IgnoreMe	EA_Importance	EA_Notes	EA_Priority	EA_Priority	EA_Difficulty	EA_Language	EA_Keywords	EA_Scope	
		0 Value1		0							
		0 Value2		0							
		0		0.00							
					Easy:		0				
					Medium:		0				

The first template demonstrates the following features:

- The presence of an index column (named “EAIndex”) containing the starting and ending row markers in between which the EA data needs to be inserted.
 - The index column is meaningful only to the template and will not be present in the generated Excel file.
 - There are 2 types of marker pairs:
 - EA_Start and EA_End: the marker rows themselves will be replaced with data loaded from EA.
 - EA_Start_Keep and EA_End_Keep: these are used in this first template to indicate that the marker rows themselves contain data to be preserved and that the rows loaded from EA need to be inserted in between these markers, without overriding them.
- Mixing of EA related columns (both standard properties and Tagged Values) with non EA data, including blank columns.
 - To distinguish the EA related columns their name must be preceded with the prefix “EA_”, including the Tagged Value property names!
 - If the template contains only EA related columns then the prefix can be omitted for all columns (i.e. do not mix defining EA columns both with and without the prefix).

- The complete list of standard EA property names supported for export are: Name, Notes, Phase, Version, Priority, Stereotype, Alias, Status, Complexity, Difficulty, Language, Keywords, Scope, Type and GUID.
- EA columns can be repeated in the sheet.
- The rows marked with a green background contain a number of sample formulas used to illustrate how these formulas can be defined inside the marker rows, their cell values to be updated automatically based on the data loaded from EA.
- Exports can be of any element type.
 - Some standard EA properties are relevant only to certain metatypes (e.g. Difficulty is relevant for Requirements but not for Classes), so it should be expected that certain cell values will contain nulls.
 - Likewise, if a Tagged Value name that is specified in the template cannot be found for an exported element a null will be exported.

An export of a sample data set using sheet 1 of Template1 is included for illustration:

G9 fx =COUNTIFS(G2:G8,"1.0")									
	A	B	C	D	E	F	G	H	I
1	EA_Name	EA_Type	EA_Stereotype	EA_Alias	EA_Status	EA_Complexity	EA_Phase	EA_Version	
2									
3	Class1	Class		CLASS1	Approved	Easy	2.0	1.1	
4	Class5	Class		CLASS5	Approved	Difficult	1.0	1.1	
5	Class2	Class		CLASS2	Approved	Easy	1.0	1.1	
6	Class3	Class		CLASS3	Approved	Medium	1.0	1.1	
7	Class4	Class		CLASS4	Approved	Easy	1.0	1.1	
8									
9							4		
10									
11									

fx =AVERAGE(T2:T8)												
R	S	T	U	V	W	X	Y	Z	AA	AB		
EA_weight	ignoreMe	EA_Importance	EA_Notes	EA_Priority	EA_Priority	EA_Difficulty	EA_Language	EA_Keywords	EA_Scope	EA_Guid		
0	Value1	0										
1	Value2	4	<>Documentation </>for Class 1.				Java	Key1	Public	{3AE55D2-559C-48ad-BDC3-AD361BABA4F6}		
5	Value2	2					<none>	Key5	Public	{68D19830-9EA5-46ba-A653-3FDA5D1F863D}		
2	Value2	8	Class 2 documentation				Java	Key2	Public	{6488BA54-4CD2-443f-BB90-23E008720CF6}		
3	Value2	6	<>Documentation </>for Class 3.				Java	Key3	Public	{871DB172-A6E1-4cbb-87D5-12964DF71B6D}		
4	Value2	4					<none>	Key4	Public	{F50AE5AE-B651-4e10-97C8-E768339050D9}		
0	Value2	0										
15		3.43										
				Easy:	3							
				Medium:	1							

Note how:

- The index column is not present in the exported file.
- The original marker rows (2 and 3 in the template) are preserved (in rows 2 and 8 of the generated sheet).
- The cells containing formulas are updated based on the data uploaded from EA.

The second sheet of the template contains the following:

	A	B	C	D	E
1			Name	Weight	Guid
2	This text should not be affected!				

It illustrates:

- The use of EA related columns only, thus without requiring the “EA_” prefix.
- That by default (without markers) all data is inserted starting at row 2, and that any pre-existing data in the sheet is unaffected by the export.

Sample output in sheet 2:

	A	B	C	D	E
1			Name	Weight	Guid
2			Class1	1	{3AEE55D2-559C-48ad-BDC3-AD3618ABA4F6}
3			Class5	5	{68D19830-9EA5-46ba-A653-3FDA5D1FB63D}
4			Class2	2	{6488BA54-4CD2-443f-BB90-23E008720CF6}
5			Class3	3	{871DB172-A6E1-4cbb-87D5-12964DF71B6D}
6			Class4	4	{F50AE5AE-B651-4e10-97C8-E768339050D9}
7	This text should not be affected!				

Templates 2 and 3

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	EAIndex	EA_Name	EA_Type	EA_Stereotype	EA_Alias	EA_Status	EA_Complexity	EA_Phase	EA_Version					
2	EA_Start												Comment1	Comment2
3														
4														
5	EA_End													
6							0							
7														
8														

Template 2 (sheet 1, shown above) as well as Template3 have a similar layout to Template 1 except that they use the other type of row index markers (“EA_Start” and “EA_End”), where the content of the specified range of rows (2 to 5 in these templates) is overwritten with data from EA.

Sample export data using Template 2:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	EA_Name	EA_Type	EA_Stereotype	EA_Alias	EA_Status	EA_Complexity	EA_Phase	EA_Version					
2	Class1	Class		CLASS1	Approved	Easy	2.0	1.1					
3	Class5	Class		CLASS5	Approved	Difficult	1.0	1.1				Comment1	Comment2
4	Class2	Class		CLASS2	Approved	Easy	1.0	1.1					
5	Class3	Class		CLASS3	Approved	Medium	1.0	1.1					
6	Class4	Class		CLASS4	Approved	Easy	1.0	1.1					
7							4						
8													
9													

Note how:

- The marker rows are updated with exported data, but the cell values of columns not related to EA are preserved (highlighted in yellow in the above example)!
- Once the end marker is reached, all further rows are inserted, not only preserving any pre-existing data in the sheet but expanding the range of any formulas applied to these marker rows!

Template 4

F2		fx =D2 * E2					
	A	B	C	D	E	F	G
1	EAIndex	EA_Name	EA_Guid	EA_Weight	EA_Importance	Criticality	EA_Criticality
2	EA_Start			0	0	0	0
3				0	0	0	0
4				0	0	0	0
5	EA_End			0	0	0	0
6					0	0.00	
7							
8							

Template 4 illustrates this further by using a non EA related column (“Criticality”) containing cells with formulas applicable to EA related Tagged Value columns (D and E in this case).

The resulting export of sample data yields:

E2		fx =C2 * D2				
	A	B	C	D	E	F
1	EA_Name	EA_Guid	EA_Weight	EA_Importance	Criticality	EA_Criticality
2	Class1	{3AEE55D2-559C-48ad-BDC3-AD3618ABA4F6}	1	4	4	4
3	Class5	{68D19830-9EA5-46ba-A653-3FDA5D1FB63D}	5	2	10	10
4	Class2	{6488BA54-4CD2-443f-8B90-23E008720CF6}	2	8	16	16
5	Class3	{871DB172-A6E1-4cbb-87D5-12964DF71B6D}	3	6	18	18
6	Class4	{F50AE5AE-B651-4e10-97C8-E768339050D9}	4	4	16	16
7				15	4.80	
8						
9						

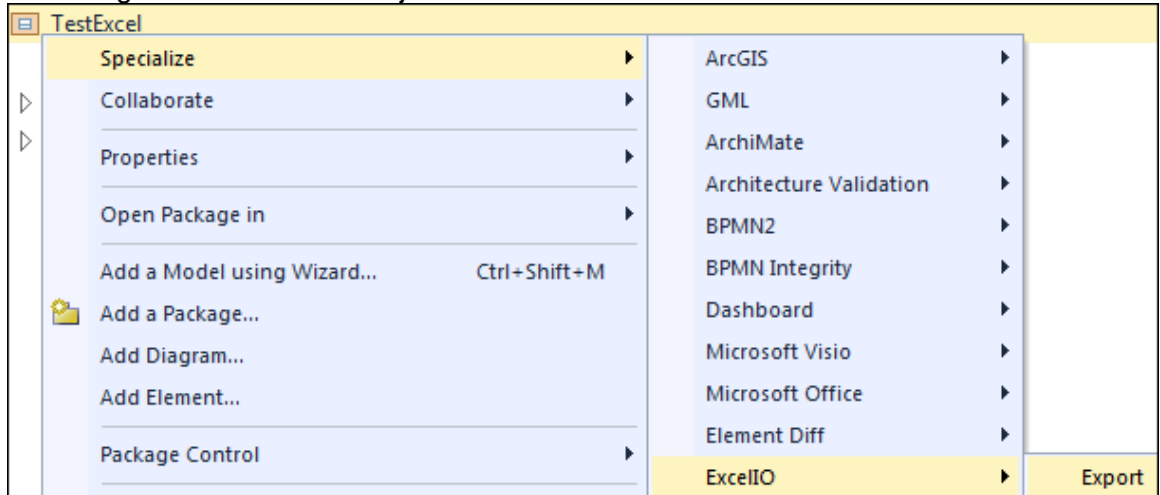
Exporting model data into Excel

The export process takes a selected set of EA elements and writes them out to a User selected Excel file based on a pre-existing template.

All worksheets in the template workbook are searched for EA related columns and, if any are found, processed for export.

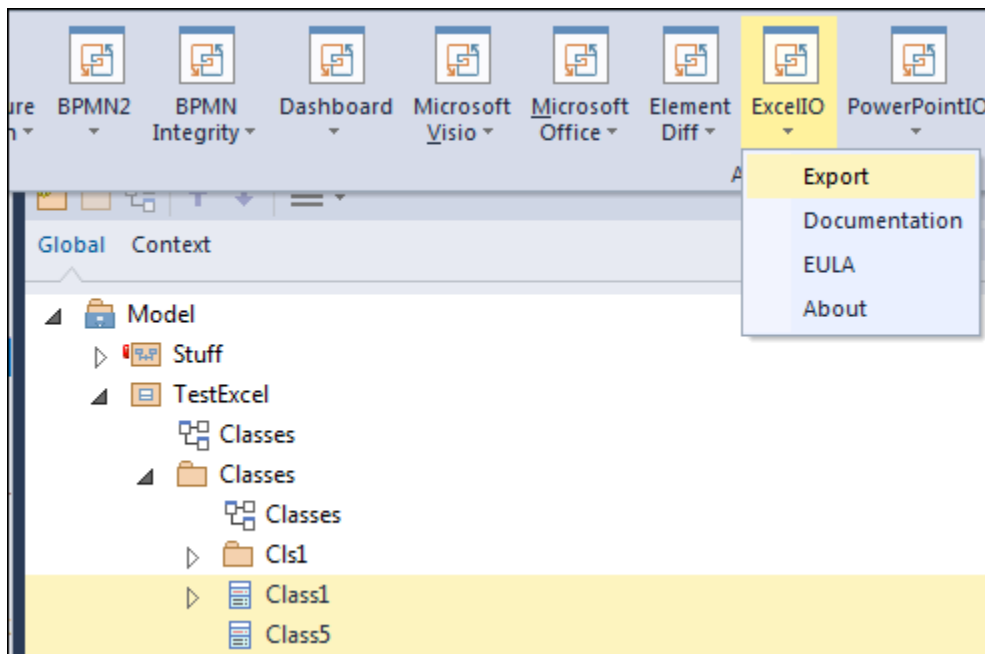
Currently the following contexts can be used as a basis for export:

- A Package selected in the Project Browser – use the context menu:

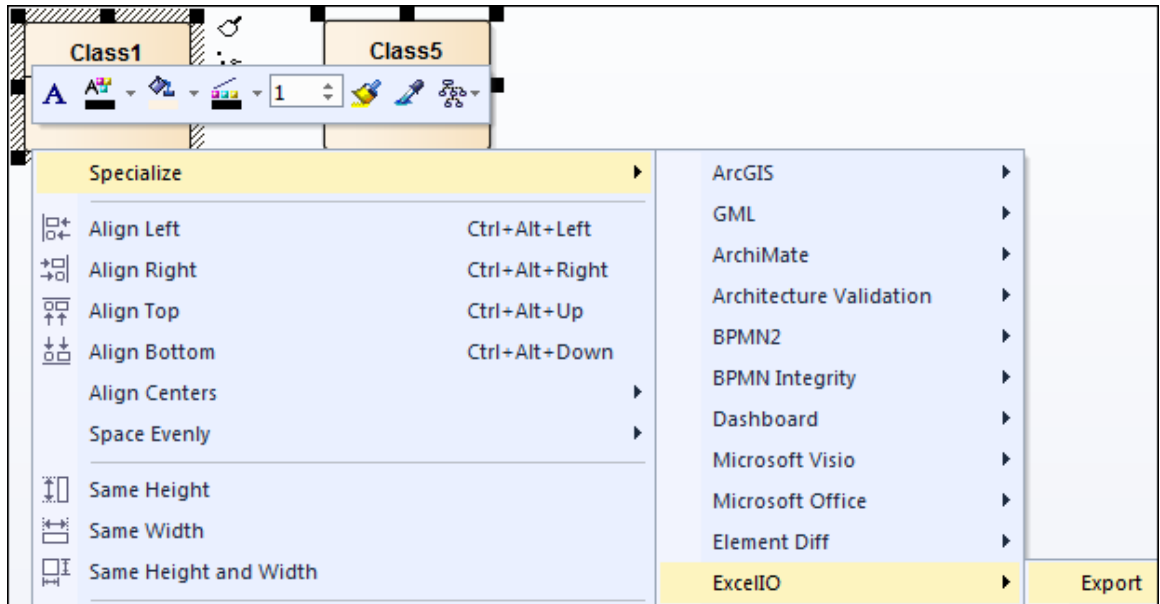


An option will be provided to include not just the selected Package, but the entire Package hierarchy underneath it.

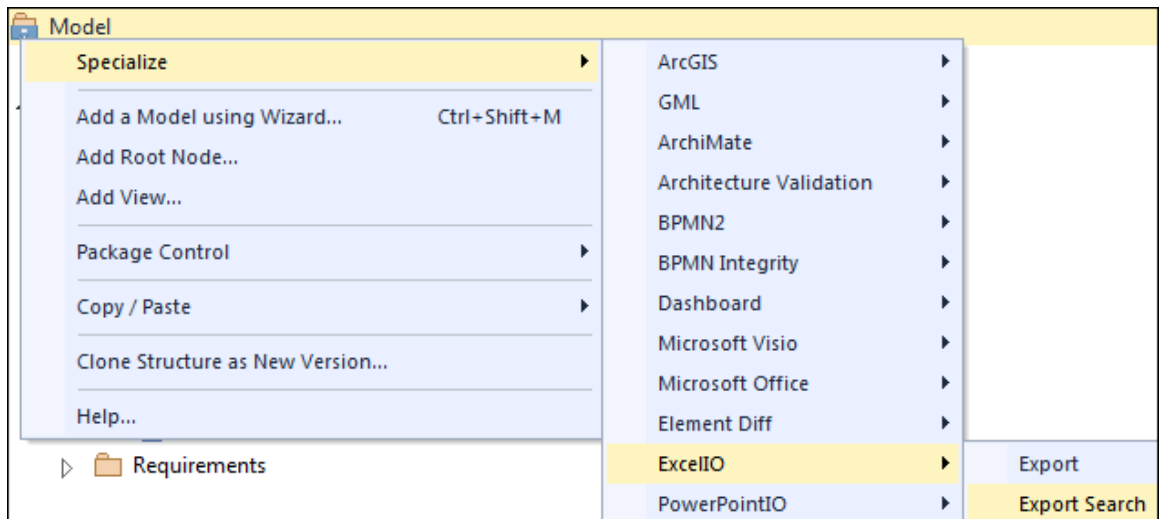
- One or more elements selected in the Project Browser - use the SPECIALIZE ribbon:



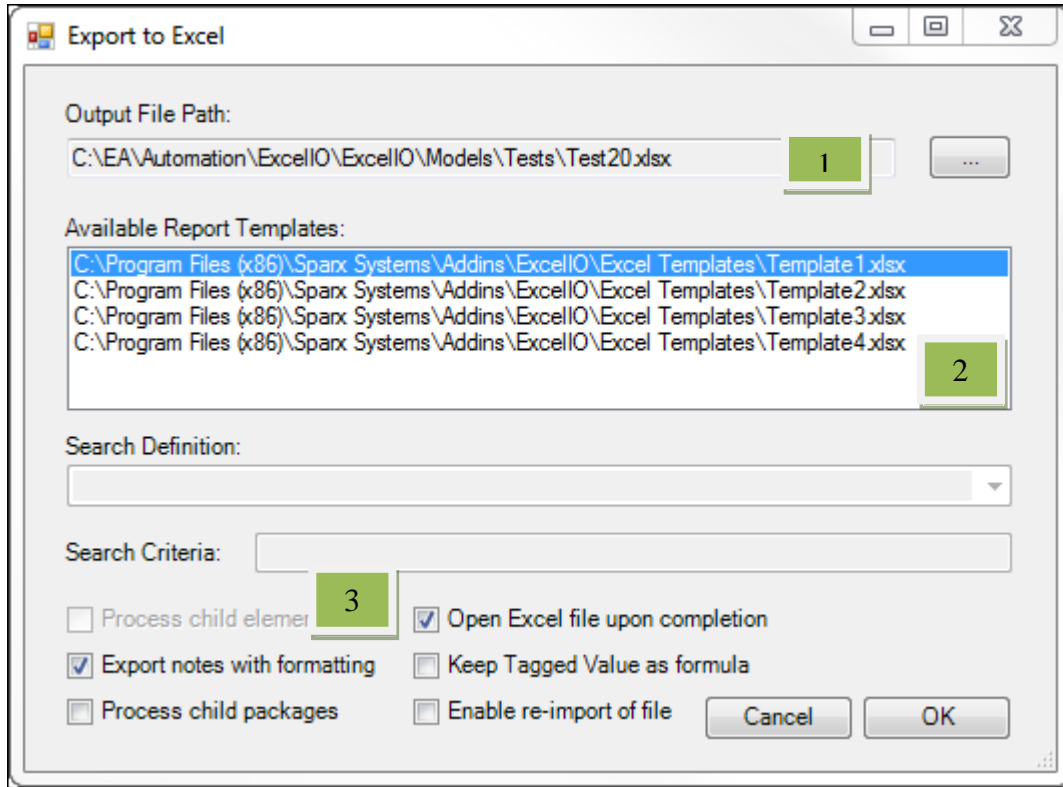
- One or more elements selected in a diagram – use the context menu:



- As the result of a search. The context for accessing the menu item is not important, so select any Package in the Project Browser to access it:



For a Package context (other than search) the following dialog box will be presented:



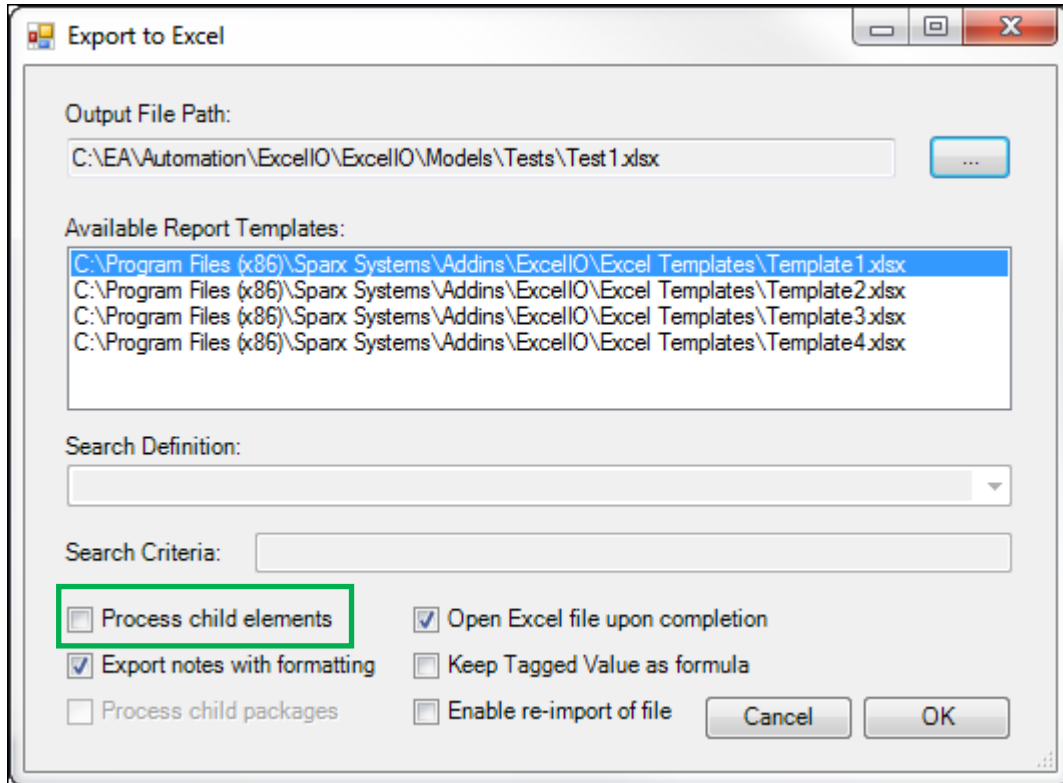
User actions:

1. Select a path for the output Excel file to be created. Existing files can be overwritten.
2. Select one of the Excel templates.
3. Checkmark any of the following options:

Option	Description
Open Excel file upon completion	Automatically opens the Excel file after export.
Export notes with formatting	Preserves the internal EA formatting of the Notes fields in the exported cells. This can be useful if the intent is to use the export Excel file as a basis for re-import later.
Process child Packages	Recursively export all elements from all child Packages under the selected Package.
Keep Tagged Value as formula	For any cells of a Tagged Value column containing a formula, preserve the formula rather than loading the value from EA. This option is only meaningful for templates containing an "EAIndex" column, where template rows are preserved using the start

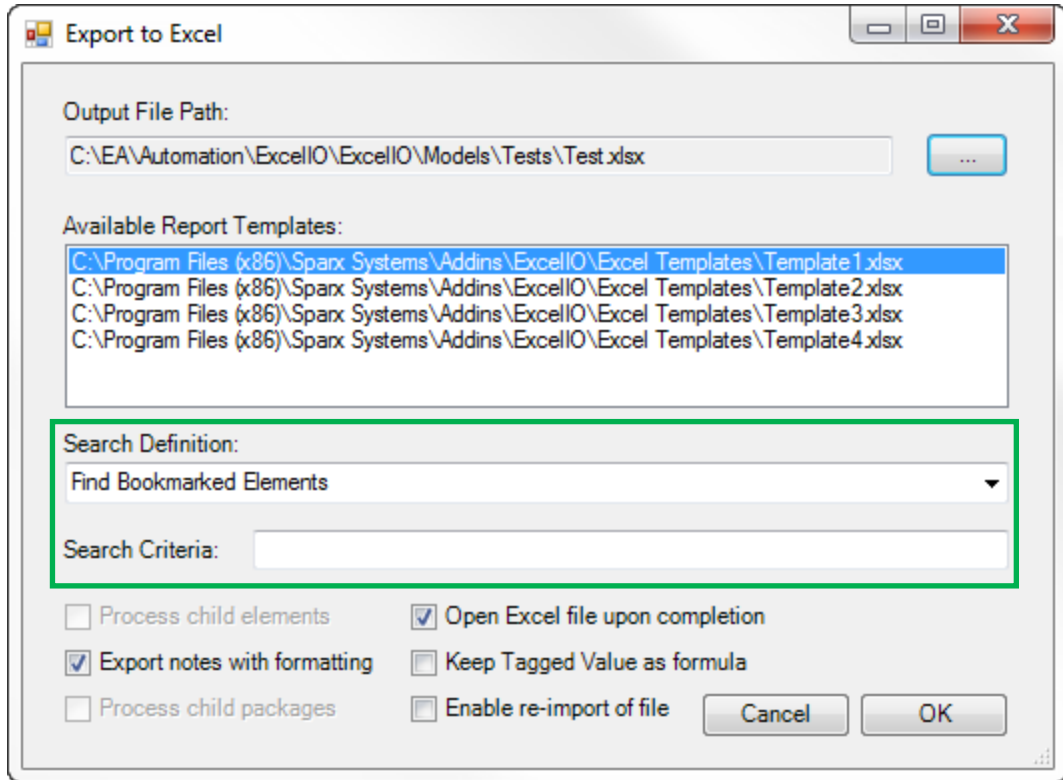
	and end indexes.
Enable re-import of file	If the template does not already contain a GUID column one is added automatically so that the export can be used later for re-import or refresh purposes.

For an export based on selected elements (from the Browser or from a diagram) the dialog box presented is:



The only difference with the previous interface is the option to recursively process child elements.

Finally, for a search based export the interface is:

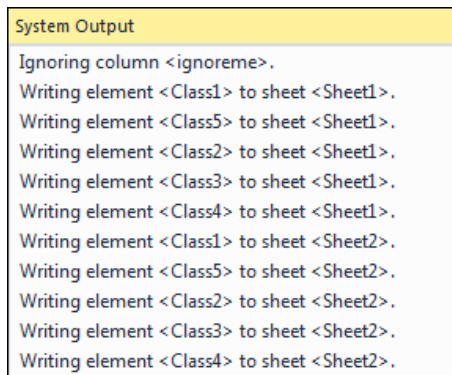


The drop-down presents a relevant subset of the standard EA search definitions plus any custom searches defined by the User. Optionally a filter value can be specified in the “Search Criteria” textbox, exactly as in the standard EA search interface.

Limitations:

- Search definitions that are loaded by an MDG Technology are not included.
- It is currently not possible to run a standard EA search, then select rows inside the result window and launch the export: as of version 14.x EA does not make the User selection/s in the results window visible to the Automation APIs ☹.

The program automatically creates an “Excel I/O” tab in EA’s Output Window, ensures that the tab is visible to the User, and logs the export results to it:



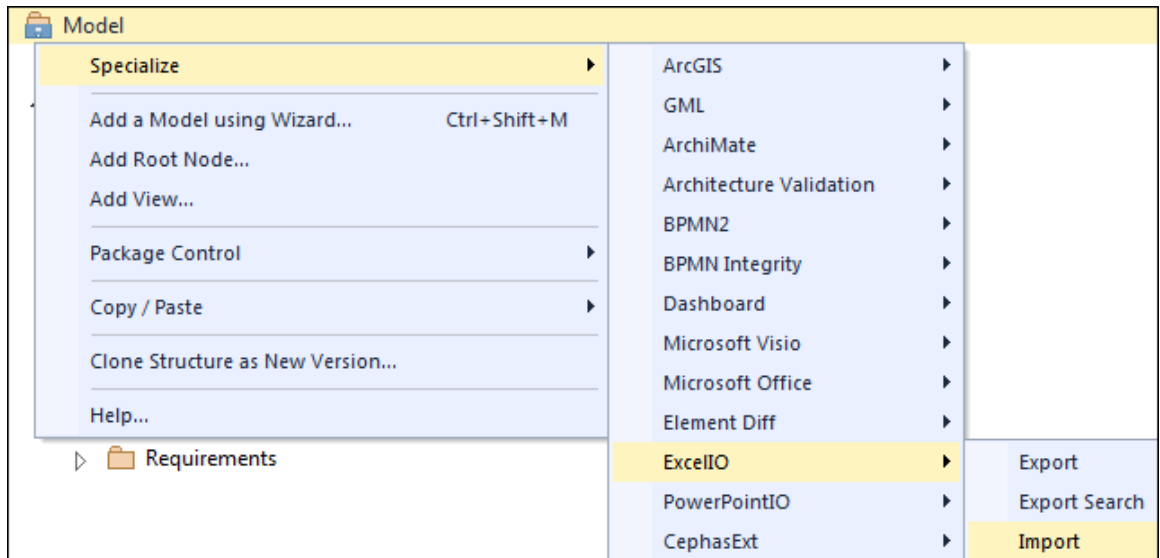
Re-importing model data from Excel

After an exported Excel file has been manually updated, the User may want to upload the values back into EA. The only pre-conditions for the updates to succeed are that:

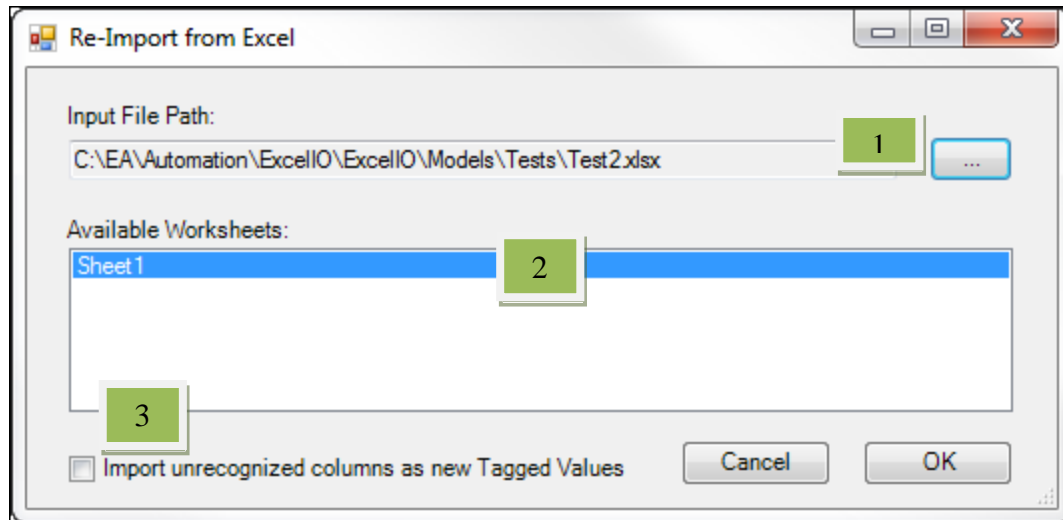
- A GUID column be present in the worksheet.
- The GUID values can be resolved in the repository where the re-import occurs.
- The element to be updated is not currently locked through EA Security or by Version Control.

The program does not try to isolate only those values that have changed: all cells of EA related columns (with non-null values) will be uploaded into the repository. To avoid extraneous and/or unwanted updates it is recommended to delete all irrelevant EA columns from the sheet/s.

Re-imports can affect elements located anywhere in the repository, so the context for accessing the menu item is not significant. Therefore select any Package in the Project Browser to access the context menu item (or use the top level “Extensions” menu):



The following dialog box will be presented:

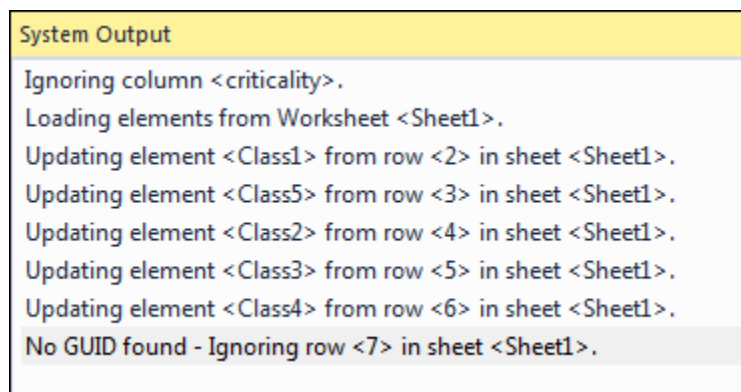


User actions:

1. Set a path to the Excel file to be uploaded. The file will be opened in read-only mode.
2. Click the worksheet/s to upload.
3. Optionally, checkmark the “Import unrecognized columns as new Tagged Values”. If set, for each unrecognized Tagged Value column a new Tagged Value is added (if not already defined!) to every row/element imported, but only if the cell value is not null.

The program automatically:

- creates an “Excel I/O” tab in EA’s Output Window.
- ensures that the tab is visible to the User.
- logs the import results to it:

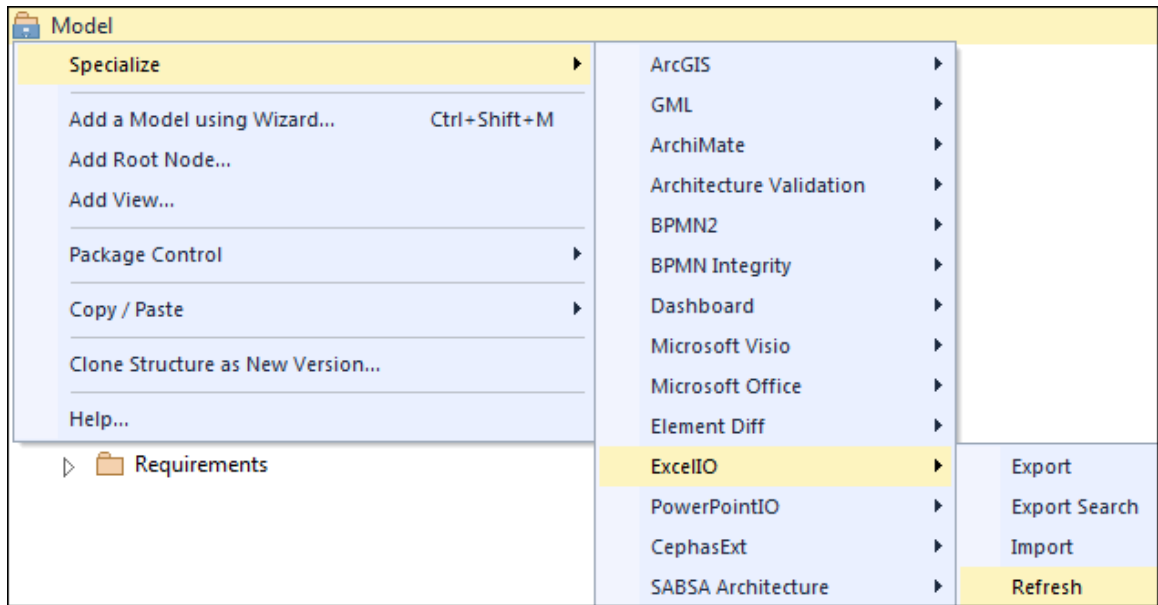


Refreshing Excel data from EA

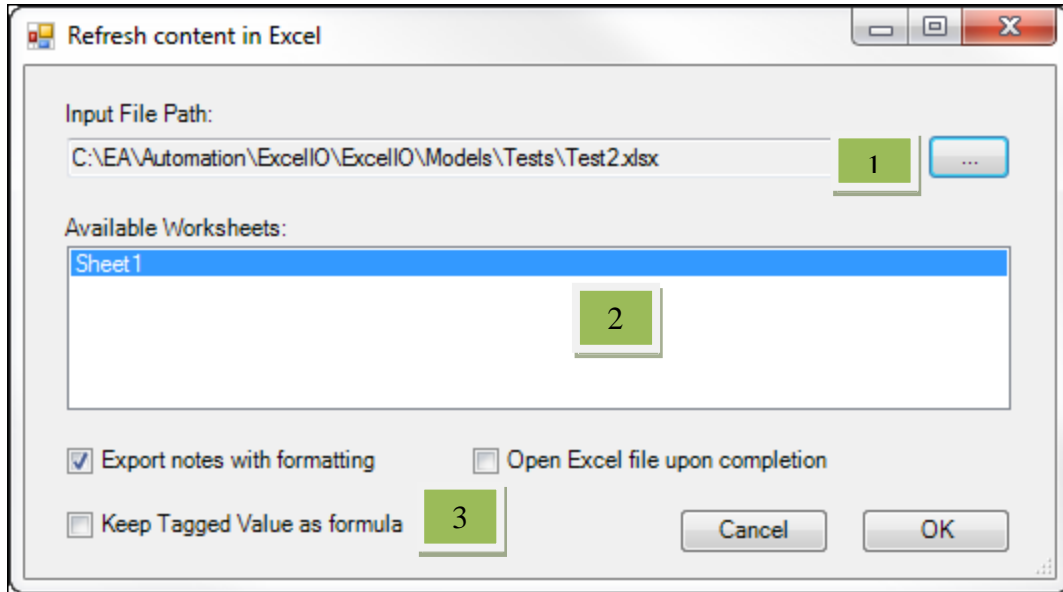
A previously exported Excel file can become 'stale' after some time, in which case the User may want to refresh the existing worksheet values from the current data in EA. The only pre-conditions for the refresh to succeed are that:

- A GUID column be present in the worksheet.
- The GUID values can be resolved in the repository where the refresh occurs.

A refresh can require access to elements from anywhere in the repository, so the context for accessing the menu item is not significant. Therefore select any Package in the Project Browser to access the context menu item (or use the top level "Extensions" menu):



The following dialog box will be presented:



User actions:

1. Set a path to the Excel file to be refreshed. The file will be opened in read-write mode.
2. Click the worksheet/s to be refreshed.
3. Checkmark any of the following options:

Option	Description
Open Excel file upon completion	Automatically opens the Excel file after export.
Export notes with formatting	Preserves the internal EA formatting of the Notes fields in the exported cells. This can be useful if the intent is to use the Excel file as a basis for re-import later.
Keep Tagged Value as formula	For any cells of a Tagged Value column containing a formula, preserve the formula rather than loading the value from EA.

The program automatically:

- creates an “Excel I/O” tab in EA’s Output Window.
- ensures that the tab is visible to the User.
- logs the refresh results to it:

```
System Output
Ignoring column <criticality>.
Loading elements from Worksheet <Sheet1>.
Refreshing element <Class1> from row <2> in sheet <Sheet1>.
Refreshing element <Class5> from row <3> in sheet <Sheet1>.
Refreshing element <Class2> from row <4> in sheet <Sheet1>.
Refreshing element <Class3> from row <5> in sheet <Sheet1>.
Refreshing element <Class4> from row <6> in sheet <Sheet1>.
No GUID found - Ignoring row <7> in sheet <Sheet1>.
```

Support and contact information

Use the contact information below for any installation or runtime issues with the extension. Feature requests or suggestions for improvement are also welcome!

Contact: Frank Truyen

Email: support@enterprisemodelingsolutions.com

Phone: 714-573-7112.

Web: <https://enterprisemodelingsolutions.com>