

Dashboard Extension for Enterprise Architect

Dashboard Extension for Enterprise Architect	1
Disclaimer	2
Dependencies	2
Overview	2
Limitations of the free version of the extension	3
Example Dashboard Grid.....	4
Installation.....	5
Verifying the installation	5
Viewing and interacting with the dashboard	8
Changing the Package Hierarchy for an Element Type.....	10
Exporting the dashboard to Excel.....	11
XML Configuration File	12
Support and contact information.....	14

Disclaimer

Version 1.0.x of the *Dashboard* extension has been successfully tested for deployment with Enterprise Architect versions 9.3 up to 12.1.

This deployment, as well as the guidelines included herein, may or may not be applicable to any later version of the tool as released by the vendor, Sparx Systems. If required, updates to this software will be made available to support future versions of Enterprise Architect.

There is no guarantee that versions prior to EA 9.3 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 *Dashboard* extension provides the ability to display a project status dashboard through a grid like interface embedded within the Enterprise Architect diagramming area (see the [example](#) below).

- This interface is non-modal so all other EA windows can be used while the extension is active.
- Current project status values are computed for the following element types:
 - Requirements.
 - Use Cases.
 - Classes.
 - Components
- Values are reported for the following element related properties, with each report section broken down by stereotype:
 - Element status (Proposed, Validated, Approved, Implemented, etc.).
 - Realization (i.e. whether an element is realized by means of a particular association in the model. For example, whether a Requirement is realized by one or more Use Cases).
 - The following properties of this grid section can be customized by the User by modifying an [external XML configuration file](#):
 - The type of connector (e.g. “Realisation”) that represents a logical realization.

Dashboard Extension User Guide

- The direction of the connector (source-to-target or target-to-source).
- The types of element that are considered to realize another one when placed in a relationship with it. For example, “Use Case” to realize a Requirement, or “Class or Component” to realize a Use Case.
 - Testing items, broken down by test type (e.g. unit, system, scenario, etc.) and by test result (pass, fail, etc.).
 - Maintenance items, broken down by type (e.g. defect, issue, etc.) and by status (new, verified, complete).
- Element sections of the grid can be shown or hidden.
 - Alternatively one or more element types (e.g. Use Case) can be disabled (i.e. prevented from being included in the grid display) by updating an external XML configuration file.
- Status sections of the grid can be shown or hidden.
 - User settings are saved to an external XML configuration file and automatically reapplied when the dashboard is reopened for the same EA repository.
- By default the project status values are reported against the entire model, but the User has the option, for each element type, to constrain the results to a selected package structure.
 - User settings are saved to an external XML configuration file and automatically reapplied when the dashboard is reopened for the same EA repository.
- The entire grid can be [exported to Excel](#).

Limitations of the free version of the extension

- Only the four element types listed (Requirement, Use Case, Class and Component) are included in the dashboard. To add additional element types (e.g. UML or BPMN Activities) please [contact us](#).
- To determine whether an element is realized or not, only a single connector type (e.g. Dependency) can be specified.
- Only package structures, and not individual packages, can be [selected as element filters](#).
- The extension has been extensively tested with Microsoft Access and SQL Server. If you encounter any difficulties in executing the underlying SQL statements on your database, please [contact us](#) with the specifics of your database setup and any error information reported by the add-in.

Example Dashboard Grid

	Requirement Type	Category	Count	UseCase Type	Category	Count	Class Type	Category	Count	Component Type	Category	Count
► Path	Project 1.Require...	Functional		Default			Project 1.Analysis ...	Analysis Class M...		Project 1.Impleme...	Component Model	
Status	Abstract	Validated	1	Default	Approved	1	Default	Proposed	11	Default	Proposed	3
	Functional	Approved	3		Proposed	11		Validated	1	System	Proposed	1
		Proposed	1	testcase	Proposed	1	enumeration	Proposed	3	Total		4
		Validated	1	Total		13	Total		15			
	Total		6									
Realized	Abstract		1	Default		2	Default		6	Default		2
	Functional		4	Total	Functional	2	enumeration		2	System		1
	Total		5				Total		8	Total		3
Unrealized	Functional		1	Default		10	Default		6	Default		1
	Total		1	testcase		1	enumeration		1	Total		1
				Total		11	Total		7			
Testing				Default	Unit - Fail	3						
					Acceptance - Fail	1						
					Acceptance - No...	20						
					Acceptance - Pass	2						
				testcase	Acceptance - No...	5						
				Total		31						
Maintenance				Default	Defect - New	3						
					Issue - New	3						
				Total		6						

Show Type:
 Requirement
 UseCase
 Show Section:
 Testing
 Maintenance

Installation

Exit any running instances of Enterprise Architect, then launch the “setup.exe” program provided by Cephaz 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 the “Addins” folder in the Sparx Systems installation directory, e.g.

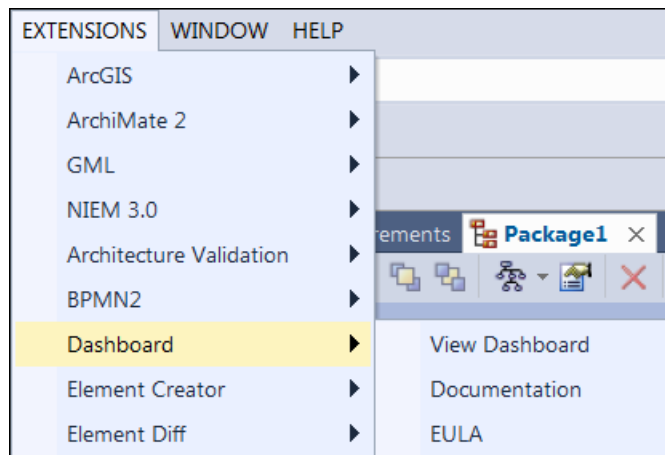
C:\Program Files (x86)\Sparx Systems\Addins.

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.

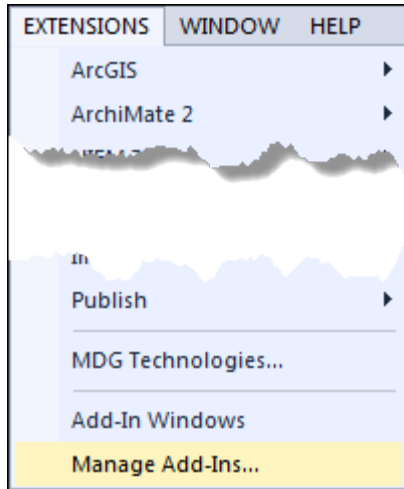
Verifying the installation

Bring up Enterprise Architect and verify that there is a “Dashboard” menu item under the Extensions menu:

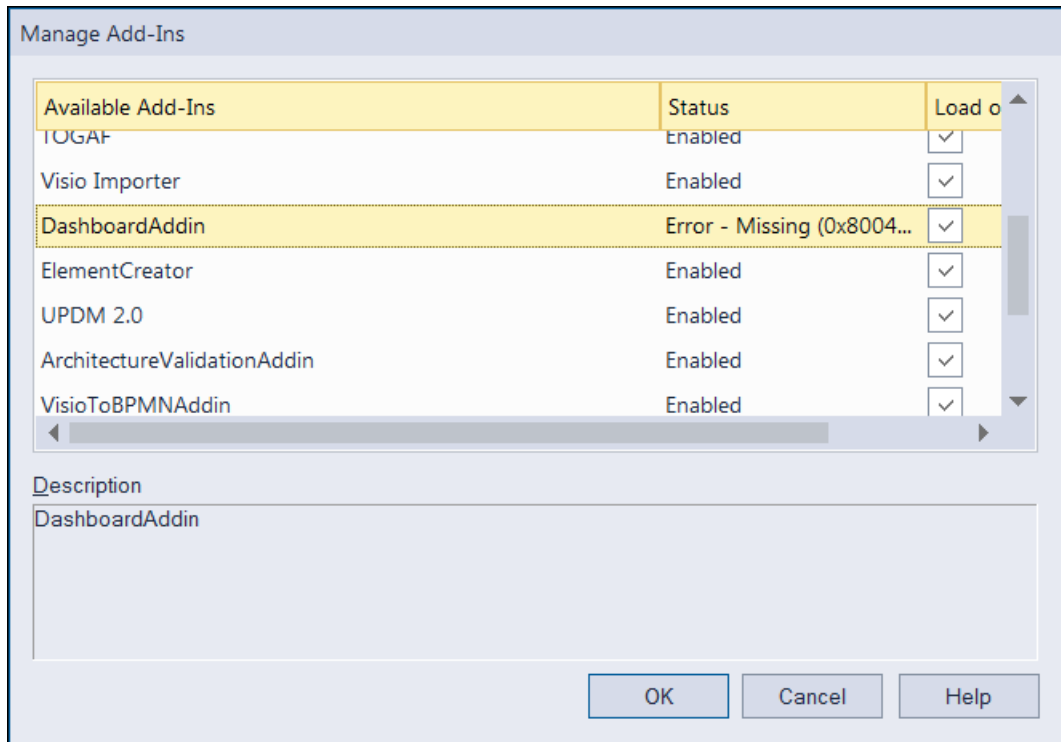


Note that the other extensions shown in the above screenshot may or may not be present, depending on your Enterprise Architect version and configuration.

Should the menu entry not be present, select the “Manage Add-Ins” menu option:



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



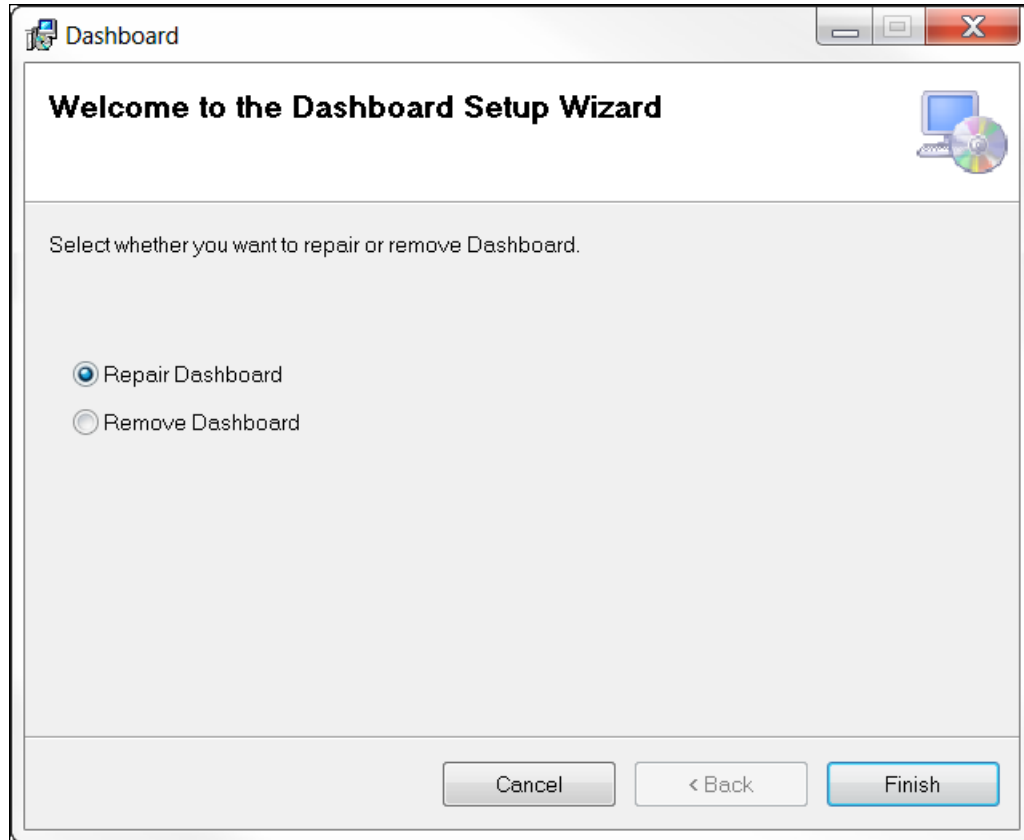
If an error status is shown, as in the example above, this typically means that either:

- 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 “Dashboard” entry itself is not found then the extension 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:



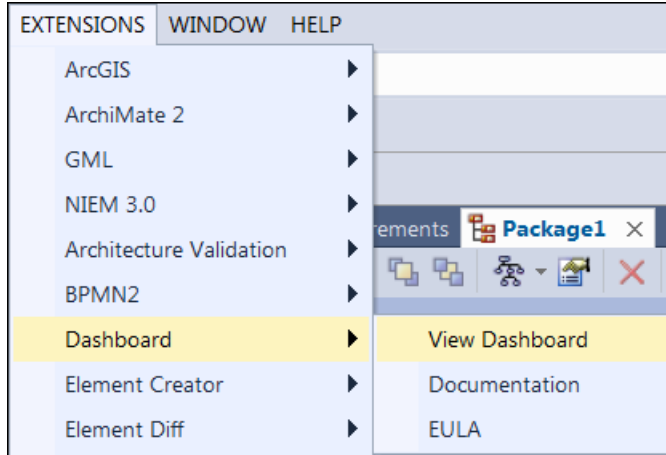
If, after the repair procedure, the *Dashboard* extension is still not loaded correctly in Enterprise Architect, remove the program through the Windows control panel and start the installation process over.

At the completion of a successful installation the following files are installed in the selected directory:

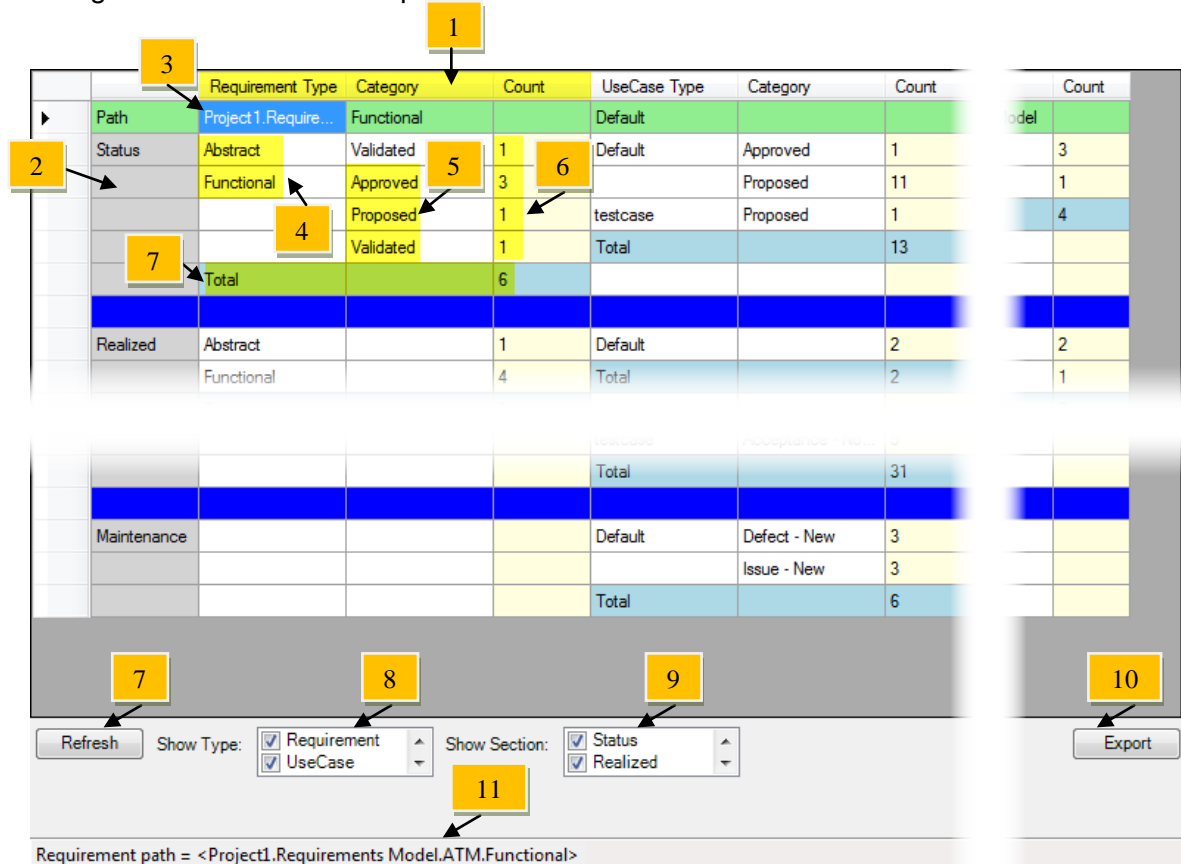
- Cephass_Software_EULA.pdf
- Dashboard.pdf (this file)
- Interop.EA.dll
- DashboardAddin.dll

Viewing and interacting with the dashboard

1. At any time select the “View Dashboard” item from the Extensions→Dashboard menu:



The following User interface is then presented:



Requirement Type	Category	Count	UseCase Type	Category	Count	Count
Path	Project1.Require...	Functional	Default			Model
Status	Abstract	Validated	Default	Approved	1	3
	Functional	Approved		Proposed	3	1
		Proposed	testcase	Proposed	1	4
		Validated	Total		1	
	Total	6			13	
Realized	Abstract		Default		2	2
	Functional		Total		4	1
			Total		31	
Maintenance			Default	Defect - New	3	
				Issue - New	3	
			Total		6	

Requirement path = <Project1.Requirements Model.ATM.Functional>

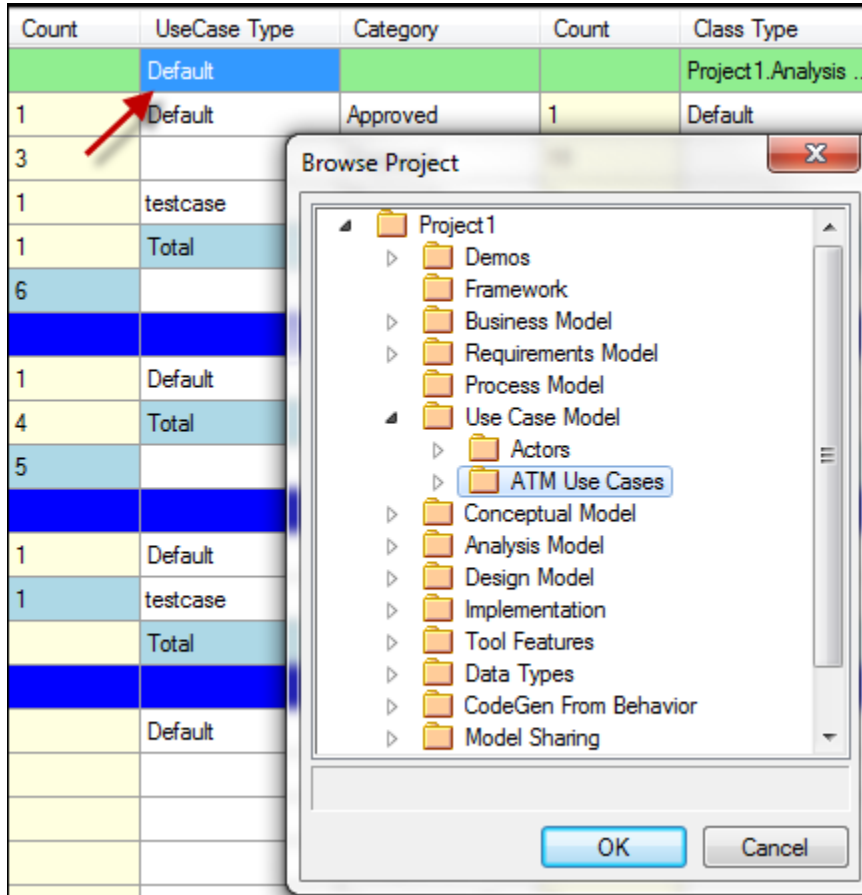
Field descriptions and User actions:

Field	Description
1.	Each element type is allocated 3 columns. The first column contains the breakdown by stereotype field. The next column may contain a secondary breakdown by category field (e.g. element status). The third column (highlighted in yellow) contains the count of elements per breakdown.
2.	The first column is reserved for the section name of the grid (Status, Realized, Unrealized, Testing and Maintenance).
3.	The first row is reserved for the package path of each element type. The string "Default" indicates the entire repository is searched for that type. The first column of each element type contains the full package path name (<u>single click</u> this cell to display the full path in the status bar – see field 11 – or <u>double click</u> the cell to select a specific package structure – see next page for details). The second column displays the top level package name of the structure and cannot be changed. A <u>single click</u> on any cell of row 0 other than the full package path will display the current location of the XML configuration file .
4.	The primary breakdown stereotype name.
5.	The secondary breakdown category name.
6.	The total count per element type for the current section. Highlighted in light blue.
7.	Click the refresh button to re-compute all of the values, for example after making changes to the model.
8.	Hide or show element types in the grid by toggling the checkmark in the dropdown list. Element types that are disabled in the XML configuration file will not be listed.
9.	Hide or show sections of the grid by toggling the checkmark in the dropdown list. Upon exiting the interface these settings are saved in an external XML configuration file and automatically applied when reopening the dashboard in the same repository.
10.	Click to export the grid to Excel . All element types and sections of the dashboard will be exported, regardless of their visibility in the grid at the time of export.
11.	The status bar provides visual feedback to any actions performed in the dashboard interface. It also allows the complete package path to be displayed (see field 3).

To exit the dashboard, simply click the "X" button in the top right hand corner of the interface.

Changing the Package Hierarchy for an Element Type

In row 0 (the top row), **double click** the first of the three columns of an element type in order to filter the status results of that type by the content of a package structure:



The first and second columns of row 0 (for that type) will be updated with, respectively, the full path name and the name of the selected package:

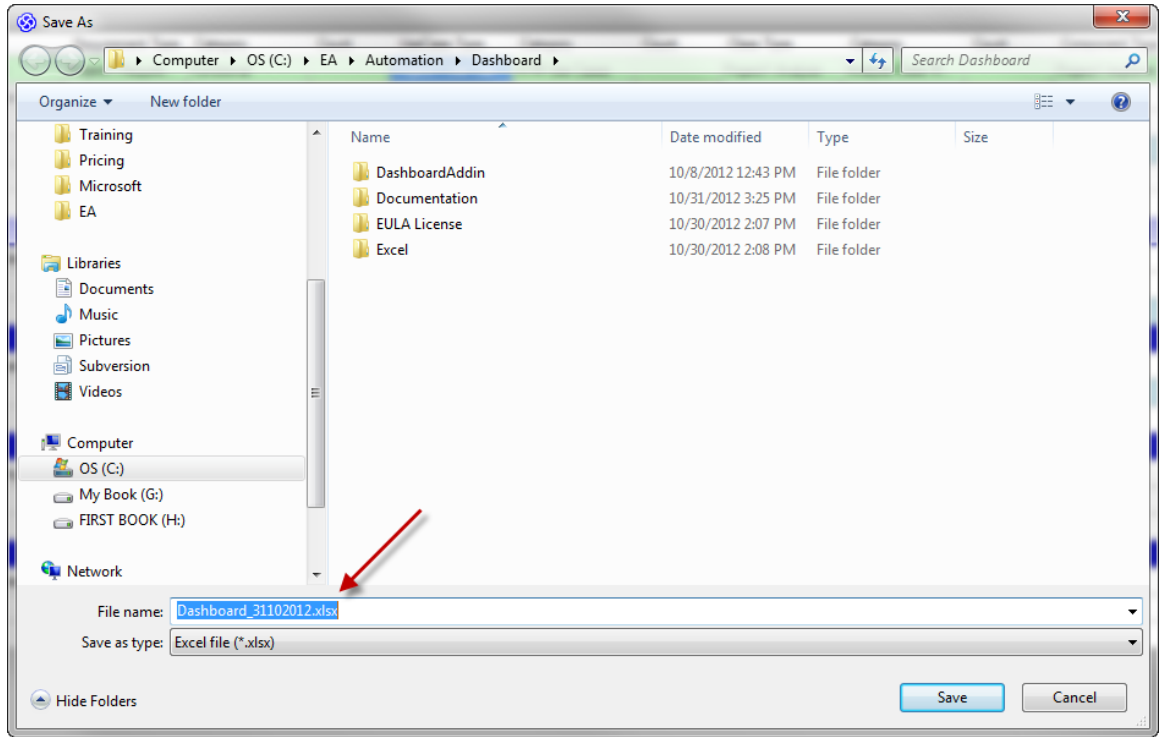
UseCase Type	Category	Count
Project 1.Use Cas...	ATM Use Cases	

To view the full package path without resizing the column, **single click** on that same cell and read the value in the interface status bar.

Caveat: the deeper the nested package structure, the longer it may take to compute the grid values (depending on database type and size)!

Exporting the dashboard to Excel

Click the “Export” button in the interface:



By default the current date is included in the proposed file name. Adjust the name if required and click the “Save” button.

Sample output:

	A	B	C	D	E	F	G	H
1		Requirement Type	Category	Count	UseCase Type	Category	Count	Class Type
2	Path	Project1.Requireme	Functional		Project1.Use Case M	ATM Use Cases		Project1.Analysis Mc
3	Status	Abstract	Validated	1	Default	Approved	1	Default
4		Functional	Approved	3		Proposed	5	
5			Proposed	1	Total		6	enumeration
6			Validated	1				Total
7		Total		6				
8								
9	Realized	Abstract		1	Default		2	Default
10		Functional		4	Total		2	enumeration
11		Total		5				Total
12								
13	Unrealized	Functional		1	Default		4	Default
14		Total		1	Total		4	enumeration
15								Total
16								
17	Testing				Default	Unit - Fail	2	
18						Acceptance - Not Run	16	
19					Total		18	
20								
21	Maintenance				Default	Defect - New	2	
22						Issue - New	2	
23					Total		4	

XML Configuration File

The first time the dashboard is opened for a given repository a default XML configuration file is created in the User's Application directory. The specific location may vary depending on the Windows operating in use.

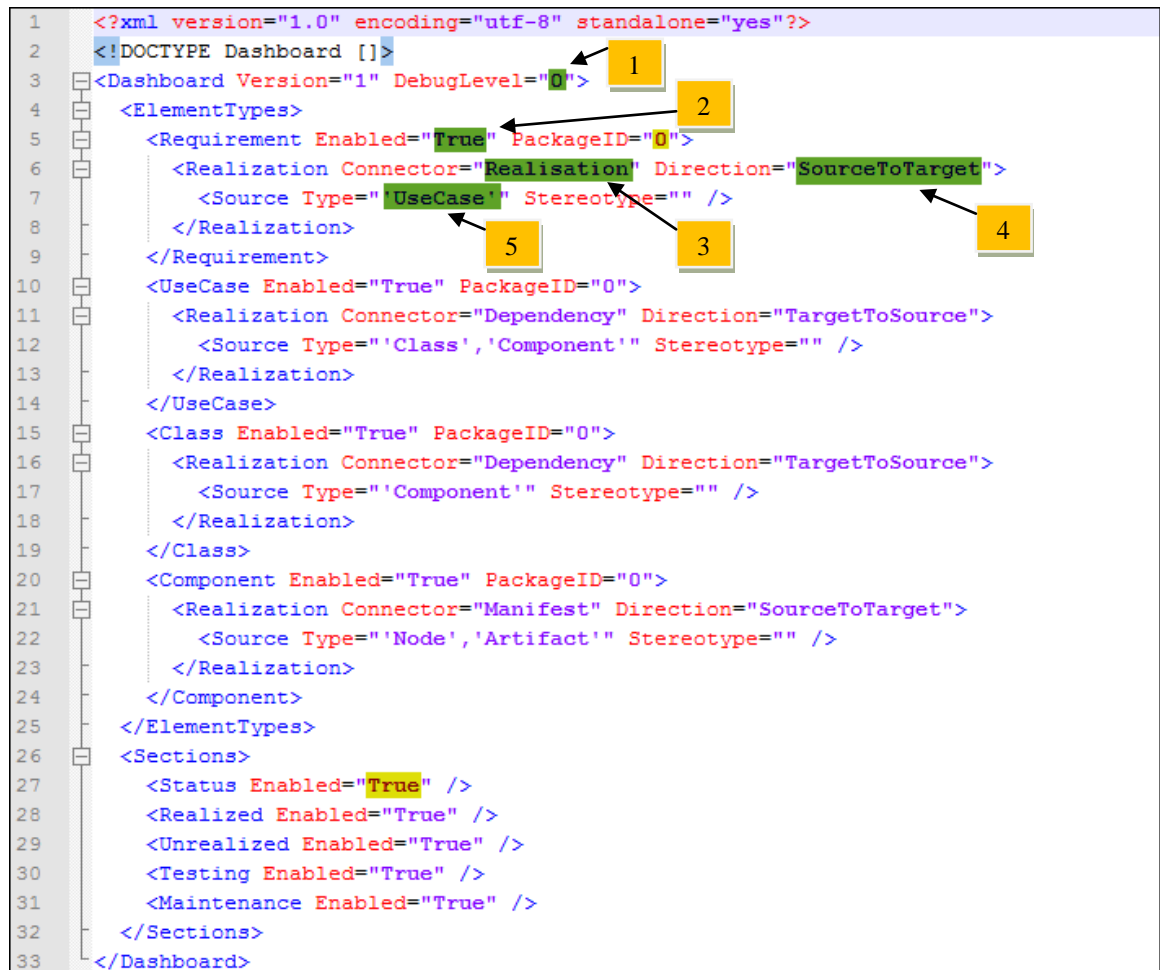
A typical file path on Windows 7 is:

C:\Users\\AppData\Roaming\Sparx Systems\Dashboards

On Windows XP:

C:\Documents and Settings\\Application Data \Sparx Systems\Dashboards

The file is keyed on the unique project GUID maintained by EA (e.g. {D2E360D0-E5C7-4314-BA2A-86D83198A739}_Dashboard.xml):



```

1  <?xml version="1.0" encoding="utf-8" standalone="yes"?>
2  <!DOCTYPE Dashboard []>
3  <Dashboard Version="1" DebugLevel="0" > 1
4  <ElementTypes>
5  <Requirement Enabled="True" PackageID="0" > 2
6  <Realization Connector="Realisation" Direction="SourceToTarget" >
7  <Source Type="UseCase" Stereotype="" /> 3
8  </Realization>
9  </Requirement>
10 <UseCase Enabled="True" PackageID="0" > 4
11 <Realization Connector="Dependency" Direction="TargetToSource" >
12 <Source Type="'Class','Component'" Stereotype="" />
13 </Realization>
14 </UseCase>
15 <Class Enabled="True" PackageID="0" >
16 <Realization Connector="Dependency" Direction="TargetToSource" >
17 <Source Type="'Component'" Stereotype="" />
18 </Realization>
19 </Class>
20 <Component Enabled="True" PackageID="0" >
21 <Realization Connector="Manifest" Direction="SourceToTarget" >
22 <Source Type="'Node','Artifact'" Stereotype="" />
23 </Realization>
24 </Component>
25 </ElementTypes>
26 <Sections>
27 <Status Enabled="True" />
28 <Realized Enabled="True" />
29 <Unrealized Enabled="True" />
30 <Testing Enabled="True" />
31 <Maintenance Enabled="True" />
32 </Sections>
33 </Dashboard>

```

The fields highlighted in yellow are automatically updated by the Dashboard extension when it is closed.

The fields highlighted in green can be changed by the User (make these updates in between dashboard launches; otherwise the changes will be overwritten by the add-in):

Dashboard Extension User Guide

1. Debug level: if problems are encountered with the add-in, set this value to 1 to see additional log information appear in EA's output window (for example, the execution times of the SQL statements).
2. Set this to "False" to prevent the element type from appearing in the dashboard grid.
3. Change the connector type that represents a realization of this type. Only a single value is accepted. Make sure the spelling matches the value expected by EA. An empty value defaults to "Realisation".
4. Set the direction to either "SourceToTarget" or "TargetToSource". The default XML configuration assumes a traditional way of marking realization for each type:

Type (Target)	Source	Direction	Connector Type
Requirement	Use Case	Source-to-target (Use Case → Requirement)	Realisation (spelled with an 's')
Use Case	Class or Component	Target-to-source (Use Case → Class/Component)	Dependency
Class	Component	Target-to-source (Class -> Component)	Dependency
Component	Node or Artifact	Source-to-target (Node/Artifact → Component)	Manifest

Change the direction as needed to match your modeling practices.

5. Set the element types that represent a realization of the target type when associated with one another using the specified connector. Enclose each value in single quotes, and separate multiple values with a comma. Alternatively set the field to empty to have **any** element that is associated through that connector type represent a realization.



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: www.enterprisemodelingsolutions.com