



erwin Enterprise Architecture Agile V2

## **User Help Guide**

© 2021 Quest Software Inc.

## ALL RIGHTS RESERVED.

This guide contains proprietary information protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of Quest Software Inc.

The information in this document is provided in connection with Quest Software products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Quest Software products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, QUEST SOFTWARE ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL QUEST SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF QUEST SOFTWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Quest Software makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Quest Software does not make any commitment to update the information contained in this document.

If you have any questions regarding your potential use of this material, contact:

Quest Software Inc.

Attn: LEGAL Dept

4 Polaris Way

Aliso Viejo, CA 92656

Refer to our Web site (<https://www.quest.com>) for regional and international office information.

## Patents

Quest Software is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at <https://www.quest.com/legal>.

## Trademarks

Quest, the Quest logo, and Join the Innovation are trademarks and registered trademarks of Quest Software Inc. For a complete list of Quest marks, visit <https://www.quest.com/legal/trademark-information.aspx>. All other trademarks and registered trademarks are property of their respective owners.

## Legend



**CAUTION:** A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.



**IMPORTANT, NOTE, TIP, MOBILE, or VIDEO:** An information icon indicates supporting information.

# Table of Contents

<b>Capabilities.....</b>	<b>1</b>
Business Process Modeling Notation (BPMN) .....	1
Login.....	17
Navigation .....	18
Objects.....	21
Smart queries .....	28
Publications .....	35
Notifications .....	39
View Manager .....	40
Workspaces .....	54
Roll-ups .....	58
Diagram/View/Publication Object representations .....	61
Campaigns .....	66
Campaign .....	66
Gamification .....	69
Dashboard.....	70
Kanban Boards.....	72
Analysis of Objects.....	79
Comparisons.....	79
Charting - Using and Creating charts .....	82
Pivot tables .....	84
Relationship Analysis .....	87
Roadmaps .....	93
Road Mapping .....	93
Agile Enterprise Architecture and Roadmaps.....	97
Customization.....	99
Import.....	99
Scorecards.....	104

Customizable metamodel .....	108
Visio Import.....	115
Settings.....	118
Account profile .....	118
Tags.....	119
Licensing .....	120
Metamodel .....	121
What is a meta model?.....	121
Innovation management metamodel.....	123
Enterprise architecture metamodel .....	124
Enterprise architecture notation .....	144
ArchiMate Concepts .....	146
Tool Change Overlays.....	147
In-Module Search.....	150
Global Search .....	151
Object Deletion .....	152
Activity log.....	153
Task management.....	154
Mentions.....	157
<b>System .....</b>	<b>158</b>
Domains .....	158
Platforms.....	159
Installation .....	160
User Roles.....	161
User Management.....	164
Change the assigned Administrator .....	164
<b>User Creation (for Administrators)</b> .....	165
Change/Assign a license to an existing platform user (for Administrators).....	167
User Removal (for Administrators) .....	168
Requesting More Licenses .....	169
Support Access .....	170
Billing Options .....	171
Terms and Conditions .....	172
<b>Agile EA Tutorials .....</b>	<b>173</b>

10-minute evaluation guide for Agile EA .....	173
Tutorial for Agile Enterprise Architecture.....	178
Importing Data .....	181
Managing Objects .....	182
Pivot Tables.....	185
Charts .....	188
Application Components and Analysis.....	191
Scorecards.....	192
Locations.....	194
Diagrams .....	196
Views and Visualizations .....	203
Table Visualization.....	205
Kanban Visualization .....	207
Roadmap Visualization.....	210
Smart Queries .....	212
Charts Revisited .....	214
Pivot Table revisited .....	215
Publications .....	217
Customization Tutorial .....	220
Creating Object Types.....	221
Creating Relationships between Object Types.....	223
Adding Attributes to Object Types.....	225
Create instances of Object Types.....	227
<b>Migration .....</b>	<b>234</b>
Open Group File Format (OGFF) to SPP - online exchange .....	234
Input, Mapping and Output .....	235
Run import of data into SPP .....	238
Re-Generate SPP Structure File (for customized SPP metamodels) .....	239
Add Customizations to the Mapping File (for customized source tool or SPP metamodels).....	240
System Architect to erwin EA Data Migration.....	242
2 Steps for importing models - non-customized System Architect metamodels.....	243
3 Steps for importing models - Customized System Architect metamodels.....	248
System Architect Diagram Import.....	260
Export from System Architect.....	261

Import into erwin EA agile .....	262
System Architect Macros.....	265
How to use the macros .....	266
Meta Model Report Macro .....	268
XML Diagram Export Macro.....	269
Migrating model data from erwin DM to erwin EA Agile.....	270
<b>Developer API .....</b>	<b>272</b>
SPP API .....	272
API Example.....	275
API Limits .....	276
Reporting with JS Reports .....	277
EA agile CSV Import/Export API (with Postman) .....	291
MMF API .....	294
<b>About Us .....</b>	<b>295</b>

---

# Capabilities

## Business Process Modeling Notation (BPMN)

BPMN allows Agile EA to provide a notation that is usable and understandable by all business users, to communicate real-world process representations, to different interested stakeholders. The end results are business process models that allow a business to

- Document and understand their internal business processes in a consistent, graphical manner
- Be more agile and efficient, with increased visibility into its underlying processes
- Understand how different business organizations interact and collaborate within internal procedures
- Communicate the processes in a technical and non-technical manner to a wide audience

The processes can also be connected into the overall enterprise architecture, giving increased visibility of business context and understanding, as inputs into decision making and architectural change.

BPMN depicts these four element types for Business Process Diagrams:

1. Flow objects: events, tasks and gateways
2. Connecting objects: sequence flow, message flow, data association
3. Swim lanes: pool or lane
4. Artifacts: data object and data storage

## Flow Objects Supported

Name	Options	Images			Description
Event	Trigger	Start	Intermediate	End	
	None	 Receive claim request	 Receive claim request	 Claim processed	An Event is something that “happens” during the course of a Process. These Events affect the flow of the model and usually have a cause (trigger) or an impact (result). Events are circles with open centers to allow internal markers to differentiate different triggers or results. There are three types of Events, based on when they affect the flow: Start, Intermediate, and End.
	Message	 Receive claim request	 Receive claim request	 Claim processed	A Message arrives from a Participant and triggers the start of the Process.
	Timer	 Receive claim request	 Receive claim request		A specific time-date or a specific cycle (e.g., every Monday at 9am) can be set that will trigger the start of the Process.
	Conditional	 Receive claim request	 Receive claim request		This type of event is triggered when a condition such as “S&P 500 changes by more than 10% since opening”, or “Temperature above 300C” become true.
	Signal	 Receive claim request	 Receive claim request	 Claim processed	A Signal arrives that has been broadcast from another Process and triggers the start of the Process.
	Multiple	 Receive claim request	 Receive claim request	 Claim processed	This means that there are multiple ways of triggering the Process
	Parallel Multiple	 Receive claim request	 Receive claim request		This means that there are multiple triggers REQUIRED

					before the Process can be instantiated.
	<b>Escalation</b>	 Receive claim request	 Receive claim request	 Claim processed	This type of End indicates that an Escalation should be triggered.
	<b>Error</b>	 Receive claim request	 Receive claim request	 Claim processed	This type of End indicates that a named Error should be generated. All currently active threads in the particular Sub-Process are terminated as a result.
	<b>Compensation</b>	 Receive claim request	 Receive claim request	 Claim processed	This type of End indicates that compensation is necessary. If an Activity is identified, and it was successfully completed, then that Activity will be compensated.
	<b>Link</b>		 Receive claim request		The Link Intermediate Events are only valid in normal flow, i.e., they should NOT be used on the boundary of an Activity. A Link is a mechanism for connecting two sections of a Process. Link Events can be used to create looping situations or to avoid long Sequence Flow lines.
	<b>Cancel</b>		 Receive claim request	 Claim processed	This type of End is used within a Transaction Sub-Process. It will indicate that the Transaction should be cancelled and will trigger a Cancel Intermediate Event attached to the Sub-Process boundary.
	<b>Terminate</b>			 Claim processed	This type of End indicates that all Activities in the Process should be immediately ended.

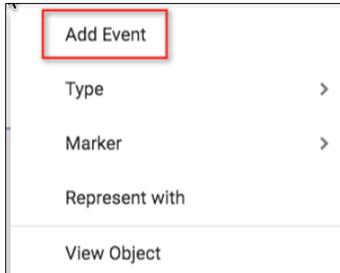
### Boundary Events

Intermediate events can be used to channel flow onto the exception path from the boundary of a task or sub-process. Boundary events can be interrupting, meaning the task or sub-process will be interrupted if the

---

boundary event receives a signal. BPMN also offers the use of non-interrupting boundary events that allow the sub-process to complete normally but initiate a parallel exception path.

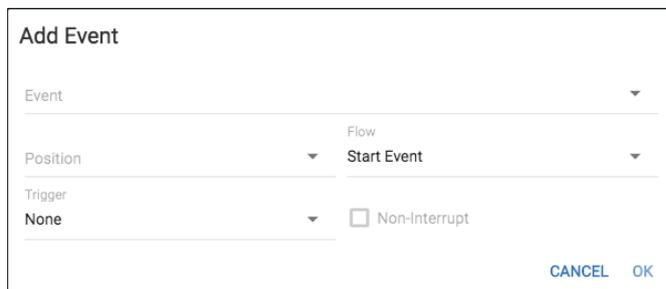
To add a boundary event, **right click a task** and select **Add Event**



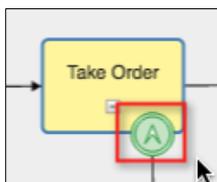
Select an existing event or create a new event in the **Event** field. Select

- The **position** you wish it to appear on the task
- The **Flow** of the event
- The **Trigger** on the event
- If it is **interrupting or not** - checking the box makes it non-interrupting

Press **OK** once you are done.



The boundary event can then be seen on the task



You can edit, view the details or remove the boundary event, by **right clicking a task** and selecting one of the option from the context menu.

---

Add Event

- Edit Event >
- View Event >
- Remove Event >

Type >

Marker >

Represent with

**Edit Event**

Event

Do Not Proceed Further ✕ ▾

---

Position Flow

Bottom Right ▾ Intermediate Event ▾

---

Trigger

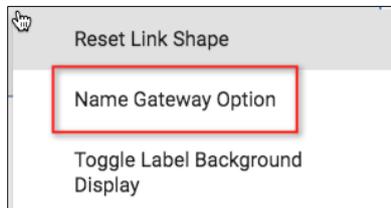
Escalation ▾  Non-Interrupt

[CANCEL](#) [OK](#)

## Gateway Objects

Name	Options	Image	Description
Gateway			
	<b>Exclusive</b>	 Resource available?	A diverging Exclusive Gateway is used to create alternative paths within a Process flow. This is basically the “diversion point in the road” for a Process. For a given instance of the Process, only one of the paths can be taken.
	<b>Inclusive</b>	 Resource available?	Used to create alternative but also parallel paths within a Process flow. Unlike the Exclusive Gateway, all condition Expressions are evaluated. The evaluation of one condition does not exclude the evaluation of other conditions. Since each path is considered to be independent, all combinations of the paths MAY be taken, from zero to all. However, it should be designed so that at least one path is taken.
	<b>Parallel</b>	 Resource available?	A Parallel Gateway is used to combine parallel flows and to create parallel flows. Not evaluating any condition or event, but used to represent two concurrent tasks in a business flow.
	<b>Complex</b>	 Resource Available?	Used for the most complex flows in the business process, often when you need multiple gateways to describe the business flow.
	<b>Event-Based</b>	 Resource available?	Similar to an exclusive gateway, the event-based gateway involves one path in the flow. The alternative paths that follow the gateway are based on the events that occur, rather than data seen in the process.
	<b>None</b>	 Resource available?	A Gateway is used to control the convergence and divergence of Sequence Flows in a Process. It determines the possible branching, merging, forking and joining of paths. Markers on the gateway will indicate the type of behavior control. The term “Gateway” implies that there is a gating mechanism that either allows or disallows passage through the Gateway.
	<b>Parallel Event-based</b>	 Resource Available?	This gateway is similar to a parallel gateway, allowing multiple processes to happen at the same time. Unlike the parallel gateway, the processes are not dependent on events, multiple events can trigger multiple processes.

You can add labels to the gateway lines, by right clicking the line and selecting to **Name Gateway Option**



**Name Gateway Option**

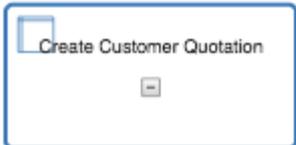
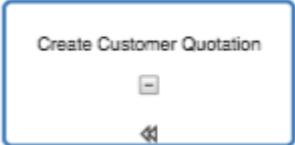
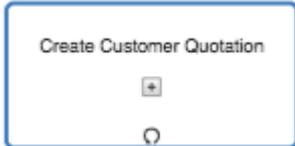
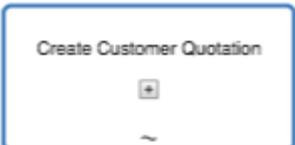
Gateway Option Name

No

[CANCEL](#) [OK](#)

### Task Objects and Sub processes

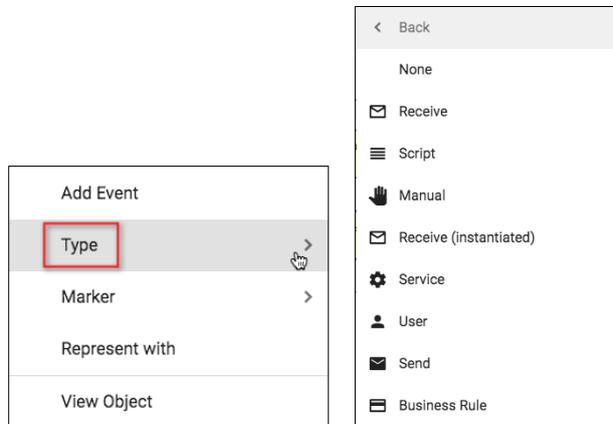
Name	Options	Image	Description
<b>Task</b>	<b>Type</b>		
	<b>None</b>		A Task is an atomic Activity within a Process flow. A Task is used when the work in the Process cannot be broken down to a finer level of detail. Generally, an end-user and/or applications are used to perform the Task when it is executed.
	<b>Receive</b>		A Receive Task is a simple Task that is designed to wait for a Message to arrive from an external Participant (relative to the Process). Once the Message has been received, the Task is completed.
	<b>Script</b>		A Script Task is executed by a business process engine. The modeler or implementer defines a script in a language that the engine can interpret.
	<b>Manual</b>		A Manual Task is a Task that is expected to be performed without the aid of any business process execution engine or any application. An example of this could be a telephone technician installing a telephone at a customer location.
	<b>Service</b>		A Service Task is a Task that uses some sort of service, which could be a Web service or an automated application.

	<b>User</b>		A User Task is a typical “workflow” Task where a human performer performs the Task with the assistance of a software application and is scheduled through a task list manager of some sort.
	<b>Send</b>		A Send Task is a simple Task that is designed to send a Message to an external Participant (relative to the Process). Once the Message has been sent, the Task is completed.
	<b>Business Rule</b>		A Business Rule Task provides a mechanism for the Process to provide input to a Business Rules Engine and to get the output of calculations that the Business Rules Engine might provide.
<b>Sub-process</b>	<b>Markers</b>		
	<b>Compensation</b>		Compensation is a specialized version of a task object that only happens when another specific task occurs previously to it.
	<b>Multiple instances</b>		Allows for creation of a desired number of task instances. The instances MAY execute in parallel or MAY be sequential.
	<b>Loop</b>		Tasks MAY be repeated sequentially, essentially behaving like a loop. e.g. 'edit draft'.
	<b>Ad-Hoc</b>		An ad-hoc task is one which has no sequence: the order in which their constituent tasks are performed is unknown or unspecified.

---

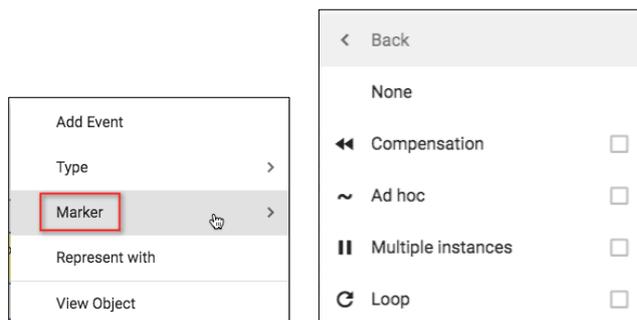
## Set Task Type

Right click a task and select **Type** from the context menu. You can then select the type.



## Set Task Marker

Right click a task and select **Marker** from the context menu. You can then select the marker(s) to be applied.



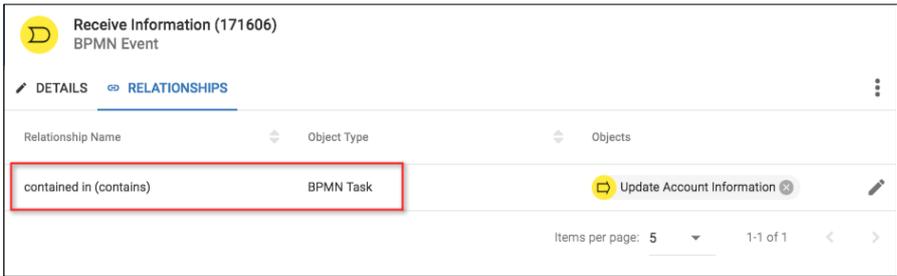
## Sub-process

BPMN provides us with the sub-process. This can be shown using either the expanding/collapsing view of a task, or by creating a URL attribute in the task object and using the URL of the child diagram that explains the process in more detail.

**Sub-Process (Collapsed)** - The details of the Sub-Process are not visible in the Diagram. A “plus” sign in the lower-center of the shape indicates that the Activity is a Sub-Process and has a lowerlevel of detail.

**Sub-Process (Expanded)** - The boundary of the Sub-Process is expanded and the details (a Process) are visible within its boundary. Note that Sequence Flows should cross the boundary of a Sub-Process.

Events, tasks and gateways can be dragged into tasks to create a sub-process. Once saved, the objects in the sub-process will have a new 'contained in' relationship added to them and the sub-process will have a 'contains' relationship to the objects that describe it in more detail.

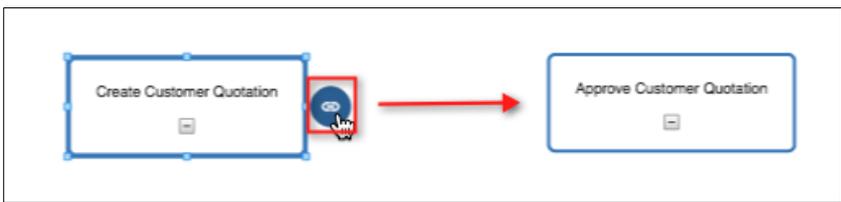


**Connecting Objects**

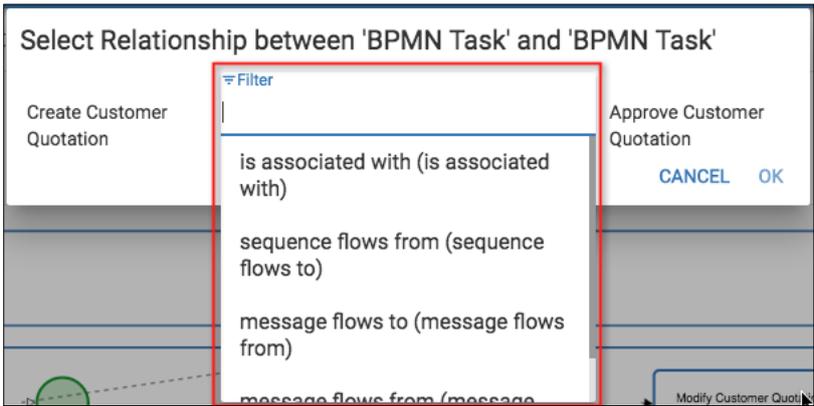
Name	Images	Description
Sequence		A Sequence Flow is used to show the order that Activities will be performed in a Process.
Message		A Message Flow is used to show the flow of Messages between two Participants that are prepared to send and receive them.
Data Association		An Association is used to link information and Artifacts with BPMN graphical elements.

**Creating relationships between objects on the BPMN diagram**

To create a relationship between objects, select the source object, click the link adornment button and drag toward the target object for the relationship (a pink box will appear around the target).

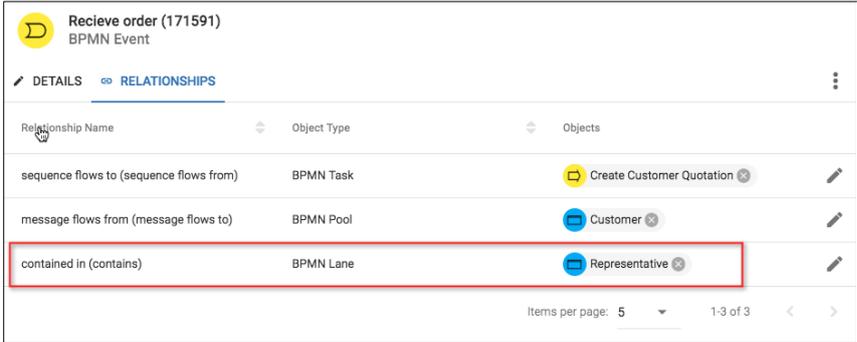


Once the relationship has been accepted, a dialog box will pop up asking you to select the type of relationship you would like to use.



**Create Relationships by dragging and dropping**

Within the BPMN diagram, dragging and dropping an object onto another object, e.g. an event into a lane, will create a 'contained in' relationship in the model.



The object cannot then be removed, until you have right clicked and select 'Remove from group', which will remove the relationship from the model.

This also extends to a sub-tasks representation.

**Swimlane objects**

Name	Images	Description
Pool		A Pool is the graphical representation of a Participant in a Collaboration. It also acts as a “swimlane” and a graphical container for partitioning a set of Activities from other Pools. A Pool MAY have internal details, in the form of the Process that will be executed. Or a Pool MAY have no internal details, i.e., it can be a "black box."
Lane	Populated within Pools	A Lane is a sub-partition within a Pool and will extend the entire length of the Pool, either vertically or horizontally. Lanes are used to organize and categorize Activities.

## How to draw Pools/Lanes

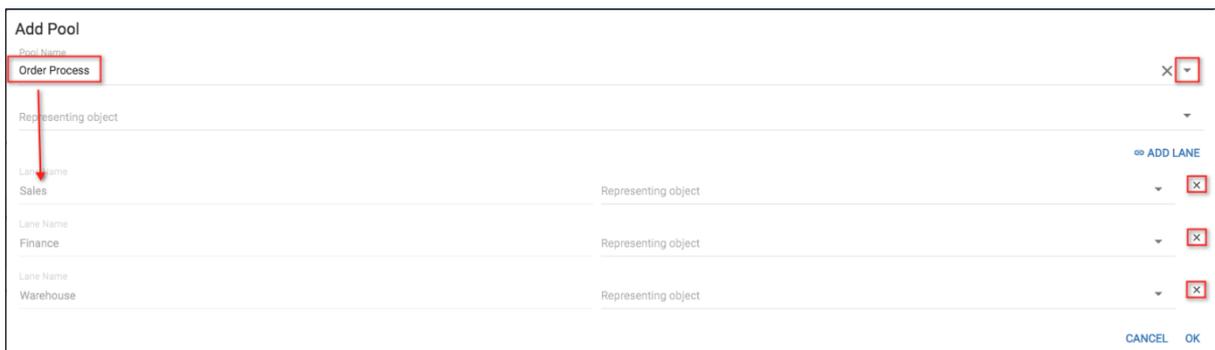
To add a pool, click and drag a Pool object from the palette onto the diagram.

If you add a new pool, a dialog will appear, where you can

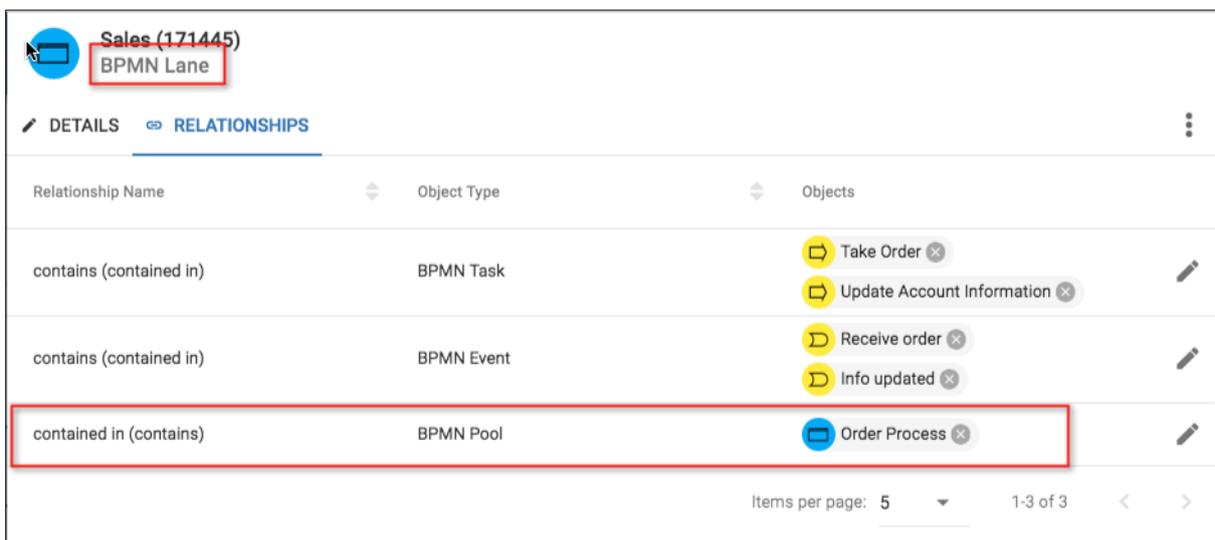
- Name the pool, creating a new pool, or re-use an existing pool by using the drop down list
- Select an ArchiMate model object that this pool represents in this process flow (explained further later in this section)
- Add Lanes to the pool, either new or existing



**i** If you re-use an existing pool, the lanes that also exist and have a relationship to that pool will be pre-populated into the lanes section. You can either choose to use these or remove them using the 'x' in their row.



The Pool to Lane relationship can be viewed in the relationships tab from either object.



Relationship Name	Object Type	Objects
contains (contained in)	BPMN Task	Take Order, Update Account Information
contains (contained in)	BPMN Event	Receive order, Info updated
contained in (contains)	BPMN Pool	Order Process

---

To Add or Remove Lanes from a pool on the diagram

- Right click the pool and choose **Configure Pool**.
- The dialog will open that will allow you to add or remove lanes from the pool.

---

## Artifacts

Name	Images	Description
<b>Data Object</b>		Data Objects provide information about what Activities require to be performed and/or what they produce, Data Objects can represent a singular object or a collection of objects.
<b>Data Store</b>		A DataStore provides a mechanism for Activities to retrieve or update stored information that will persist beyond the scope of the Process. The same DataStore can be visualized, through a Data Store Reference, in one or more places in the Process.

## Layouts

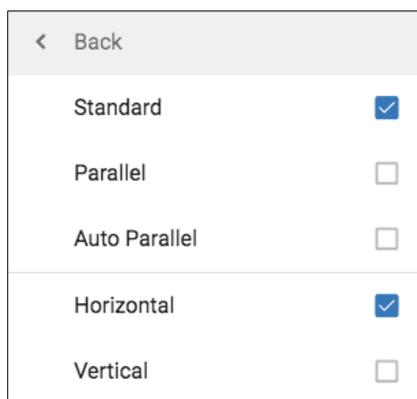
There are several layouts supported for use with BPMN - these can be found under the **View > Layout** menu;

**Standard** - This allows a user to layout the diagram as they wish and show connections from top, bottom, left and right of all objects

**Parallel** - This limits the connections from tasks, events and gateways to be from right or left of the objects

**Auto Parallel** - This is a mode that will lay out the diagram for users. Whilst the checkbox is checked, any object added to the diagram will be added in the most logical place - it will move accordingly as connections are made to it and from it. You will need to move into Standard or Parallel options to be able to move the shapes to where you want them.

The diagram and swim lanes can be modelled in horizontal or vertical fashion.



You may want to change the routing of a relationship that has been drawn in Auto Parallel mode, once you come out of that mode. To do this right click the relationship, and select **Reset Link Shape** option - this will make the link direct again.

---

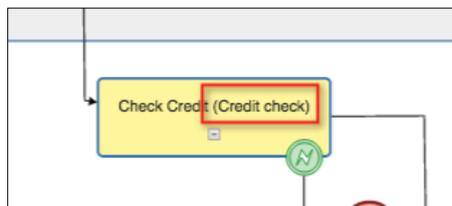
## Represent ArchiMate objects with the BPMN process flows

We still want to use the BPMN process flows created by our team in the context of our enterprise architecture. To do this, we can choose to allow the BPMN objects created to 'represent' existing objects from our model - this gives us context as to the wider effects that a process change might have and ensures we don't work in a silo, just because this process is the only thing in our remit.

When we add a Pool object to a BPMN diagram, we will be offered the 'Representing object' field, within the configuration dialog. For other objects, the 'Represent with' option can be found in the right click context menu.



Once an object is selected, it will be displayed in brackets in the name of the BPMN objects on the diagram. The example below, the BPMN process 'Check Credit' is now representing an ArchiMate business process 'Credit check'. A common use for this will be to represent BPMN Lane objects as business roles/actors that exist within the model.



A 'represented by/represents' relationship can be seen for this within the Relationships tab of both objects.

## Diagram Kinds

Using a diagram kind, you can create a diagram type that uses the traits of another diagram type, but you can control the object types associated with it.

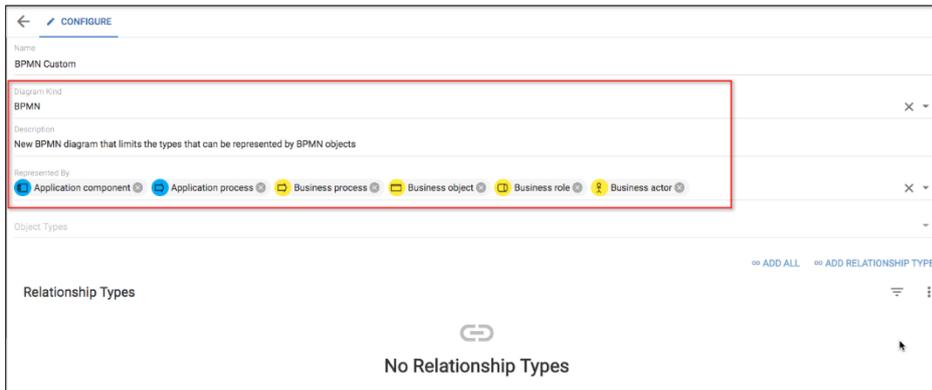
In this context, if we created a new diagram type, of 'kind' BPMN, we can tailor what model object types can be represented by BPMN Objects. As default any existing model object can be selected to be represented.

To do this, go to **User menu > System > Diagram Types**

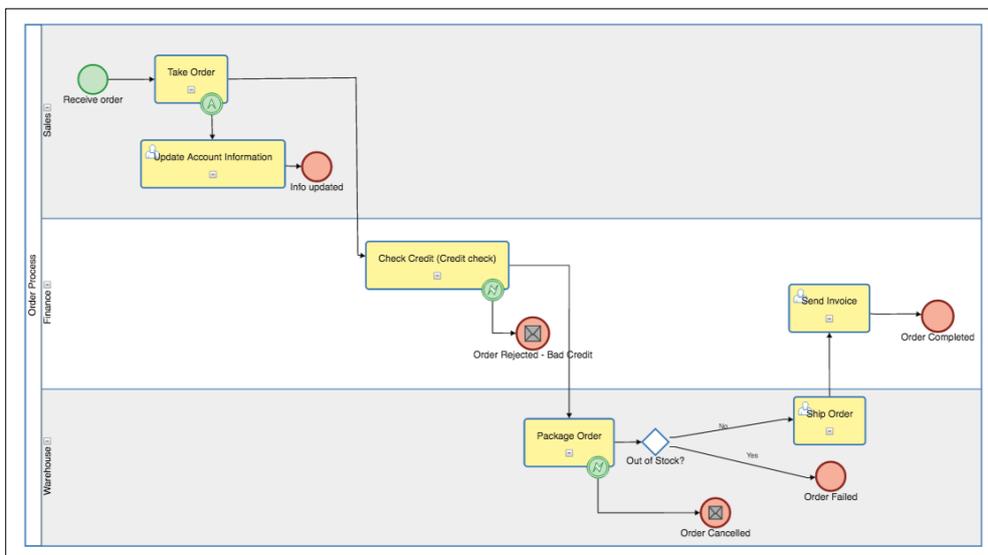
- Add a new diagram type (using the + button)
- Name the type
- Select the kind to be BPMN

- In the Represented By field, select the existing model object types you would like to allow to be selected. You may create different versions for different teams, to control the objects they can select within the model.
- - As you have selected the BPMN kind, the Object Types option will be greyed out.

Once this is done, you can begin creating your new custom BPMN diagrams.



Example BPMN process flow

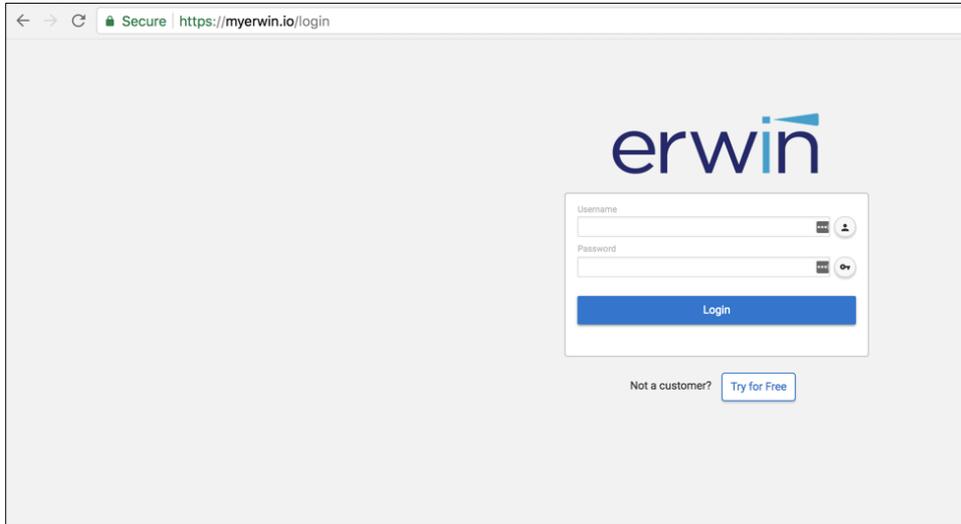


---

# Login

**Role Applicable to:** [Client Administrator](#), [User](#), [Contributor](#), [Reviewer](#)

You can login with your user name and password.

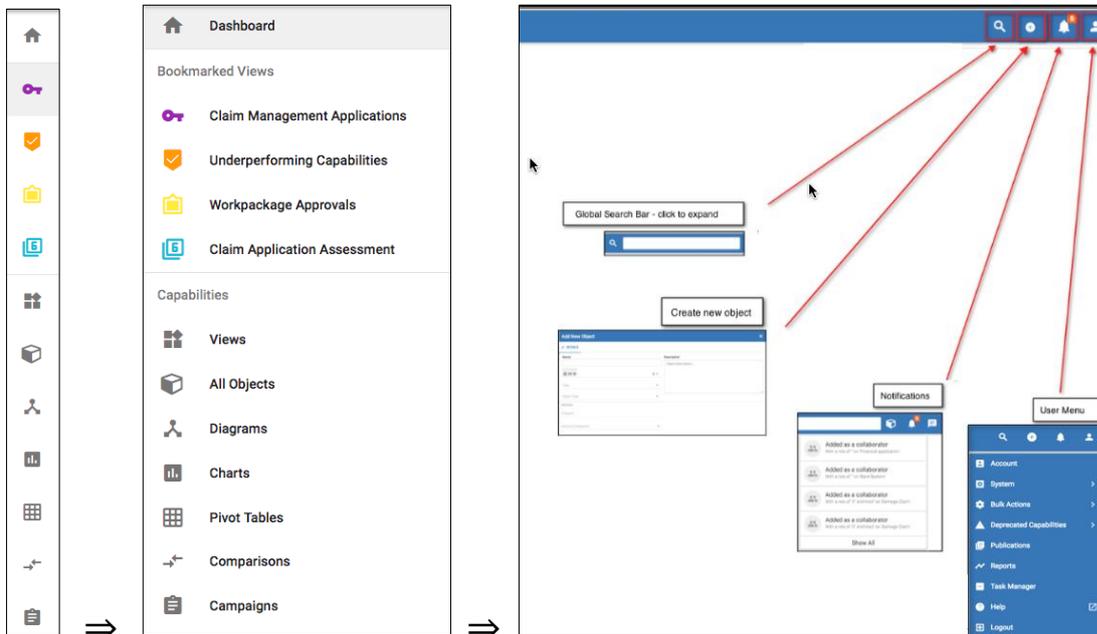


If you have forgotten either of these, you can click on the button at the end of each field to receive the information via email.

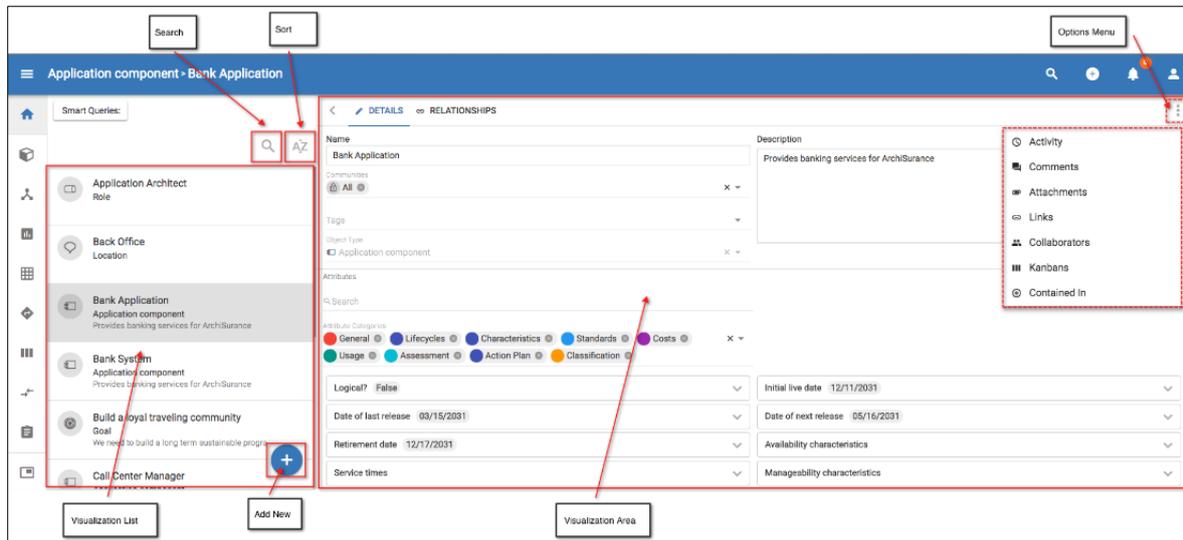
**i** | If you want to add an icon for the platform to your desktop, drag and drop the icon next to the URL to your home screen. On an iPad or Android, follow the local instructions for your browser.

# Navigation

## On the Dashboard

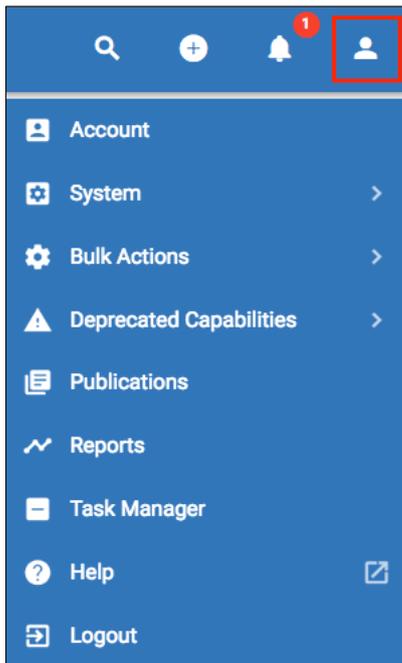


## When a menu item is selected ('All Objects' example)



## User Menu Options

Select the User Menu to allow access to



- Account (Includes Communities, System Alerts, Tags, User Profile, Change Password, Create Users, Assign licenses and Enable Support Access)
- System (Role restrictions apply)
  - Object Types
  - Attribute Categories
  - Diagram Types
  - Scorecards
  - Workspaces
- Bulk Actions
  - Object Export
  - Object Import
  - Object Delete
  - Diagram Import
  - Erwin Import
- Deprecated Capabilities
  - Roadmaps
  - Kanbans
  - Kanban Types
- Publications
- Reports
- Task Manager
- Help
- Logout

---

## Navigation within a view

Each view will have an **Action Menu**

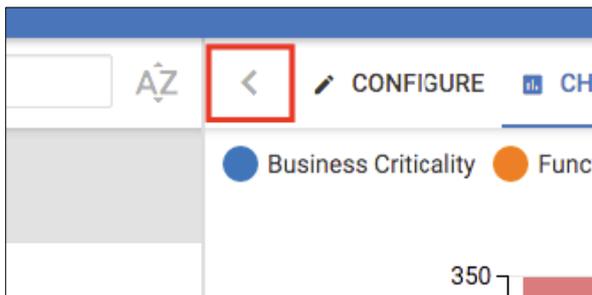


This will contain various actions relevant to the view you're in, for example **Toggle Totals** for pivot tables can be found in there.

This will also be the publication button available in every view from this menu. 

Smart Queries will also be found there for each view. 

You can also minimise the list on every page to give yourself more room to work with, to do this click the arrow to the left of the tabs.



---

# Objects

**Role Applicable to:** [Client Administrator](#), [User](#), [Contributor](#), [Reviewer](#)

The core components of the platform are known as objects. There are different types of object, including but not limited to, Idea, Goal, and Requirement.

Objects can be related to each other via a relationship. For example, a Goal may be associated with an Idea. A Goal may also be made into sub components of a Goal.

Objects represent information that you want to store on the platform.

Each object has a set of attributes. The attributes may be amended or added to by the client administrator.

Click [here](#) for a full list of ArchiMate Objects.

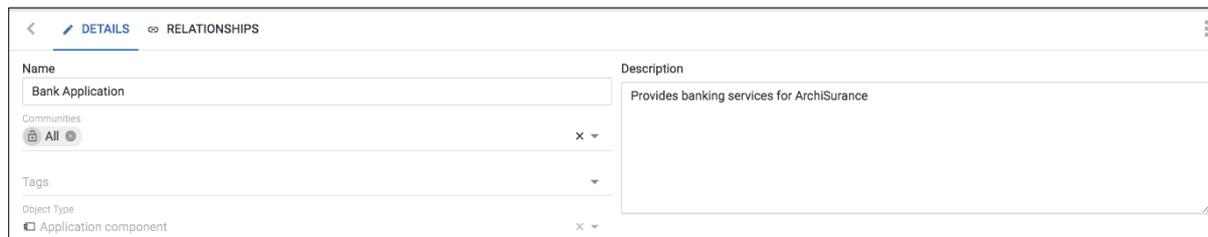
**i** The Agile Enterprise Architecture package supports the [ArchiMate](#) objects, and capability concepts as defined in [TOGAF](#)

When opening or defining an object, a set of tabs provides for further elaboration.

## Details

Each object is created with a name, description and tags.

Every object contains these details.



The name and description may be collapsed by clicking on the button with a chevron.

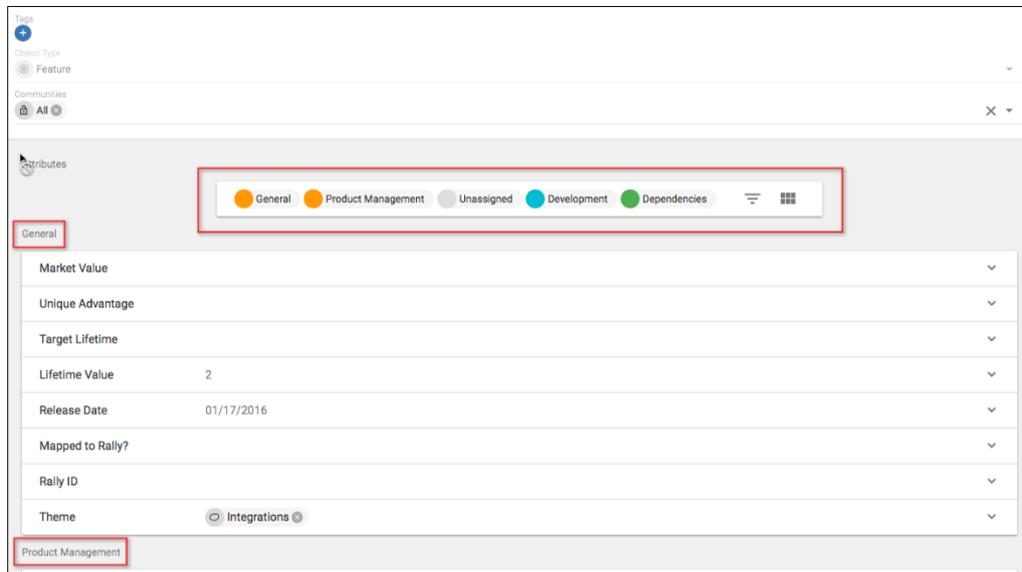
## Attributes

Further attributes show the set of additional properties that can be used to describe the object. The set of attributes differ by the type of object, and are shown beneath the description.

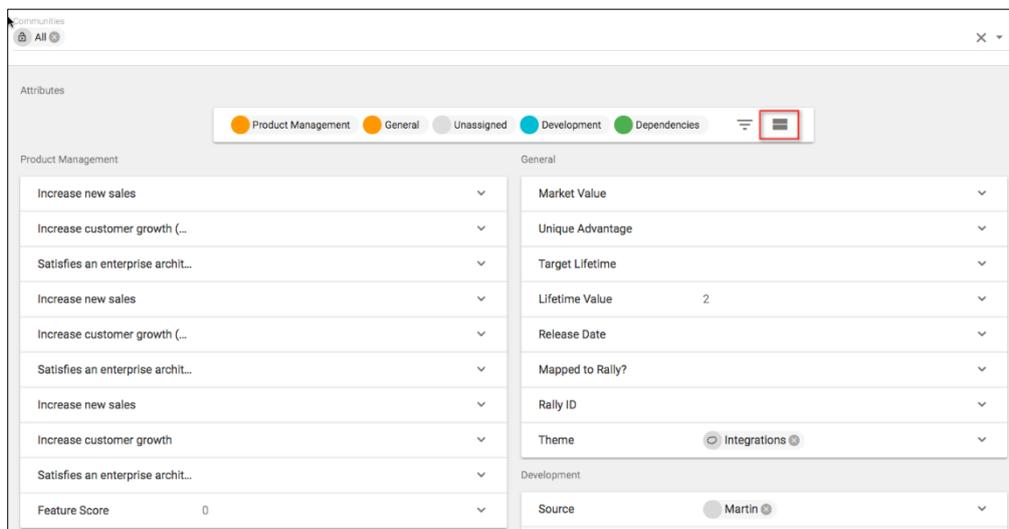
Attributes are assigned a category at the object type level.

Attribute categories are ordered in the categories bar. Categories in the bar can be selected/de-selected (shows/hides category from object view) and ordered (by dragging) them in the list. Preferences are also remember for the object type and will be applied the next time you open an instance of that type.

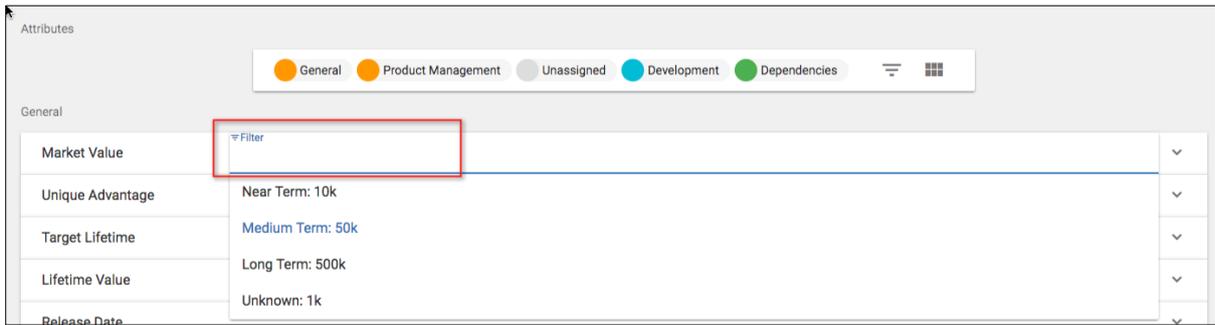
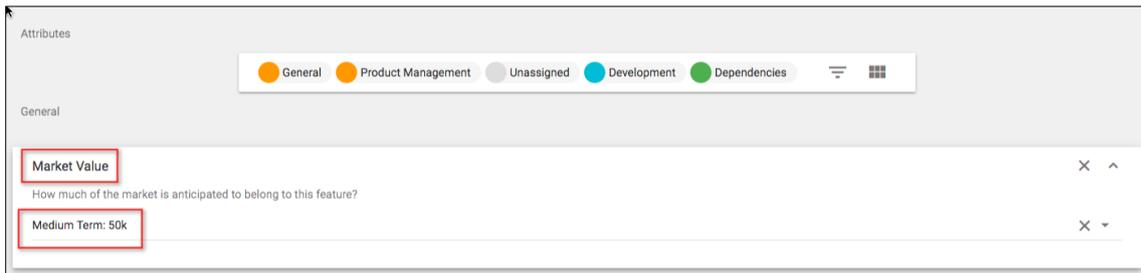
Users can also filter the list and show just the attributes needed. Any existing attribute groups have been converted to categories.



Attributes by default are shown in a single column, but selecting the button furthest to the right in the category bar, will display in 2 columns (bar position 1 on the left, bar position 2 on the right, 3 on the left, 4 on the right, etc.)



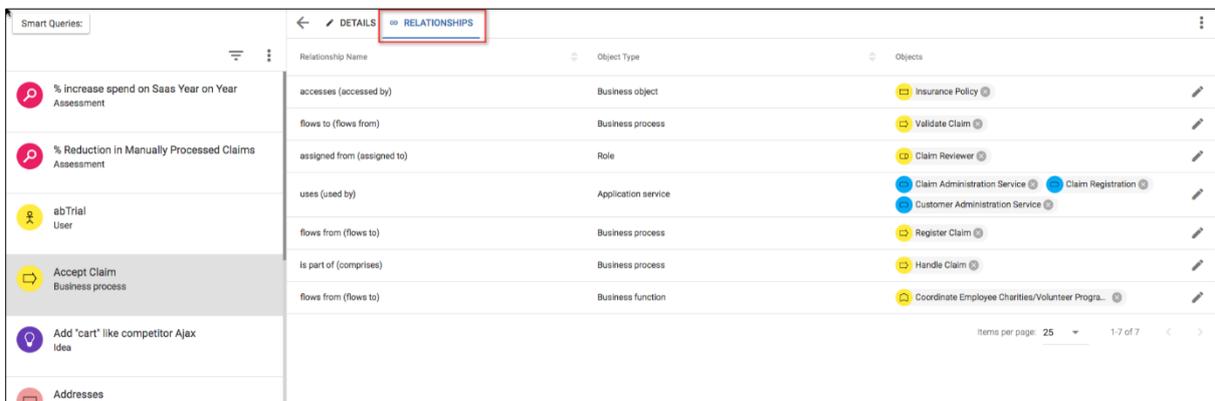
Clicking on the name of the attribute will focus onto the attribute, for you to select or type a value. Clicking on an existing value in an attribute, will open a quick edit area, where you can edit the value directly. Users can also tab between attributes.



## Relationships

Relationships allows you to connect the current object to other objects via a relationship.

For example, a Business process may flow to other Business processes, or use an Application service.



To add a new relationship, click on the Add button (bottom right hand corner). Choose a relationship type and an object type to relate with. Next choose the objects that this object is related to within the list that is returned. Use the checkboxes to select multiple objects, and click Save to create the relationship.

Relationship Type		triggers (triggered by)	
Object Type		Business process	
Business process: Accept Claim triggers			
<input type="checkbox"/>	Name	Type	
<input checked="" type="checkbox"/>	Accept Claim	Business process	
<input type="checkbox"/>	Handle Claim	Business process	
<input checked="" type="checkbox"/>	Pay Claim	Business process	
<input type="checkbox"/>	Process invalid claim	Business process	
<input type="checkbox"/>	Register Claim	Business process	

Items per page: 5 | 1-5 of 6

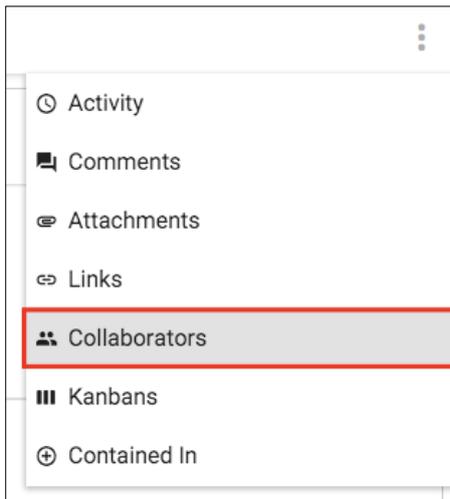
[CANCEL](#) [SAVE](#)

To modify a relationship, select the pencil at the end of the relationship type/object type row.

In the pre-populated table, you can select the checkboxes to create, or uncheck checkboxes to remove relationships, and select Save.

## Collaborators

The collaborators menu provides the capability to add other stakeholders to this object with a specific role.





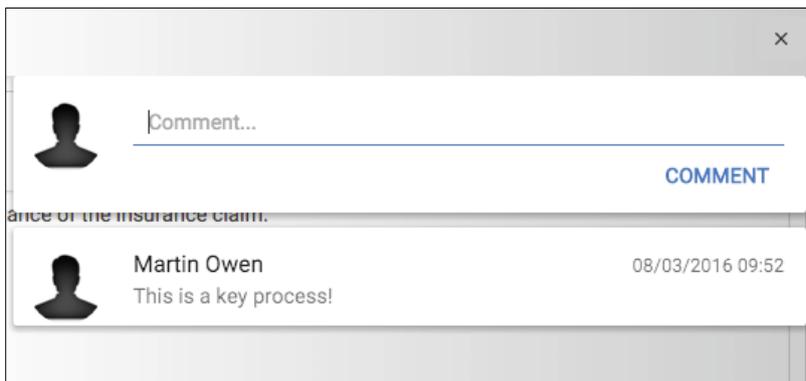
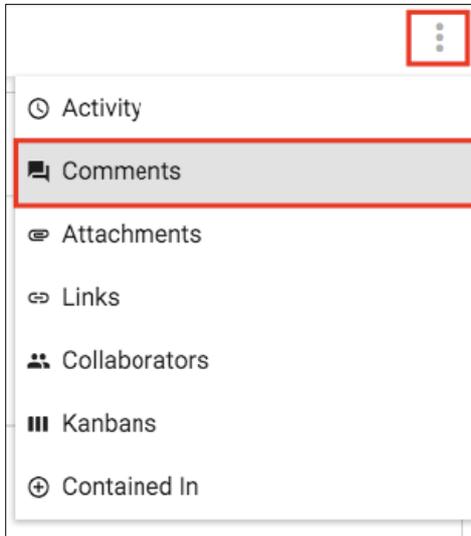
[ADD](#)

The collaborator will receive a notification email to collaborate about this object.

---

## Comments

Comments can be added for any object.

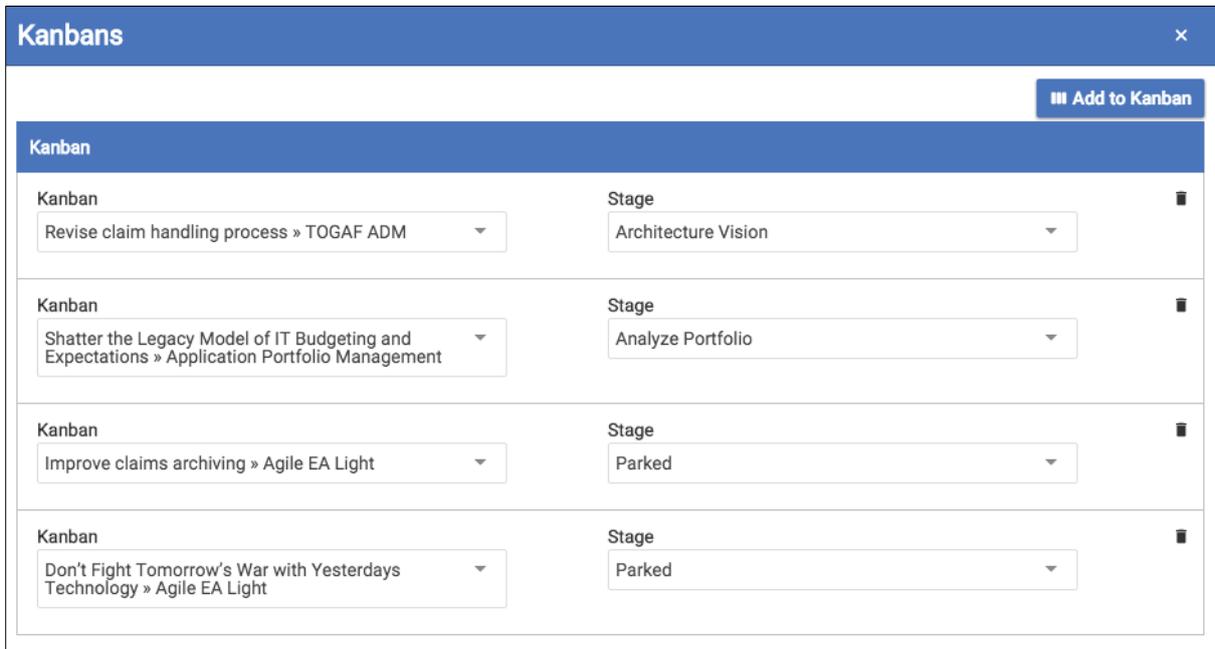
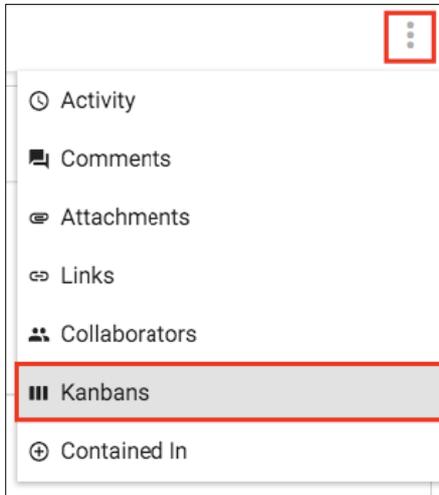


Users can also use the @ symbol to notify other users that a comment was made referencing them and the user will receive an email.

---

## Kanbans

The kanban menu shows the Kanban boards that you can move this concept through.



**i** Concepts may be removed from Kanban's here and added to other Kanban's (if the user has the correct privileges).

---

## Actions



The delete button will remove the object from the repository.



The save button will save the current object.



The duplicate button will create a new object with all of the same data, you will just be asked to provide a name.

---

# Smart queries

**Role Applicable to:** [Client Administrator](#), [User](#), [Contributor](#), [Reviewer](#)

Smart queries allow users to create queries that can filter the objects on the platform. Smart queries can be applied to all platform views.

Existing smart queries may appear as buttons (abbreviated name) on the main view of a module. These are only shown when they are 'bookmarked' for a user. Users can use smart queries created by other users, choosing to bookmark if they wish, by selecting the checkbox within the smart query list. Hover on the buttons to see a brief description of the query applied within that button.



## Smart Queries button

When you first enter a view the **Smart Queries** button will be hidden, to show it:

- Select the **Action Menu** button



(This will be in the bottom right of your view)

- Then select the **Smart Queries** icon from the list



The **Smart Queries** button will then be displayed.

---

The **Smart Queries** button is used to display the query manager. From here we can create new and manage existing queries.



The query builder form will be presented to you.

A screenshot of the "Smart Queries" configuration interface. The title bar is blue with "Smart Queries" and a close button. Below the title bar are two tabs: "CONFIGURE" (active) and "LIST". Under "CONFIGURE", there are three buttons: "Reset", "Filter" (selected), and "Colorize". The main area contains a criteria builder with an "Invert" checkbox, a "Criteria Type" dropdown set to "Name", an "Operator" dropdown set to "contains", and a "Value" input field. Below this are input fields for "Name" and "Description". At the bottom, there are fields for "Icon", "Abbreviation", and "Communities" (set to "All").

- When you add a name, notice that the abbreviation is created automatically for you. You can change this to be something more meaningful.
- Ensure that you enter a description that says what the query does.
- Queries can be tied to communities. Any other member of the community will be able to see and use your query.
- Criteria are entered for the query to use (see below).
- Queries can be used to **filter** the objects by the criteria, or **colorize** the objects based on the criteria.

**i** Tip: Build queries from a general perspective and not from your own viewpoint. This makes them more usable for others. e.g. Do not call the query - 'My applications' and then set the criteria to be your name!

## Criteria

You can build criteria with **AND** and **OR** conditions.

**AND** conditions are nested and **OR** conditions are in a different lane.

Criteria options currently are;

- Name
- Has Tag
- Relationship
- Created By User
- Created Date
- Type

- attribute
- By Kanban
- By Diagram
- By Campaign
- By Roadmap
- Incorporate Another Smart Query

We can also incorporate existing queries, using the incorporate another smart query criteria option. - When the query is then chosen from the list, the criteria from the chosen incorporated query is populated into the form.

The example below uses the **Filter** mode;

The screenshot shows a 'Filter' mode interface with three criteria rows. Each row has an 'Invert' checkbox (unchecked), a 'Criteria Type' dropdown, and an 'Operator' dropdown. The first row has 'Created By User' as the criteria type and 'aidan kelly' as the user. The second row has 'Created Date' as the criteria type, 'before' as the operator, and '05/01/2016' as the value. The third row has 'Name' as the criteria type, 'contains' as the operator, and 'claim' as the value. Each row has a trash icon and an '&' or 'OR' button. The '&' buttons for the second and third rows are highlighted with red boxes.

Notice the above query is for multiple criteria;

- where the objects **Created By User "aidan kelly"** AND
- the objects' **Created Date** is **before "05/01/2016"** AND
- the **Name** of the object contains the word **"claim"**, will be filtered.

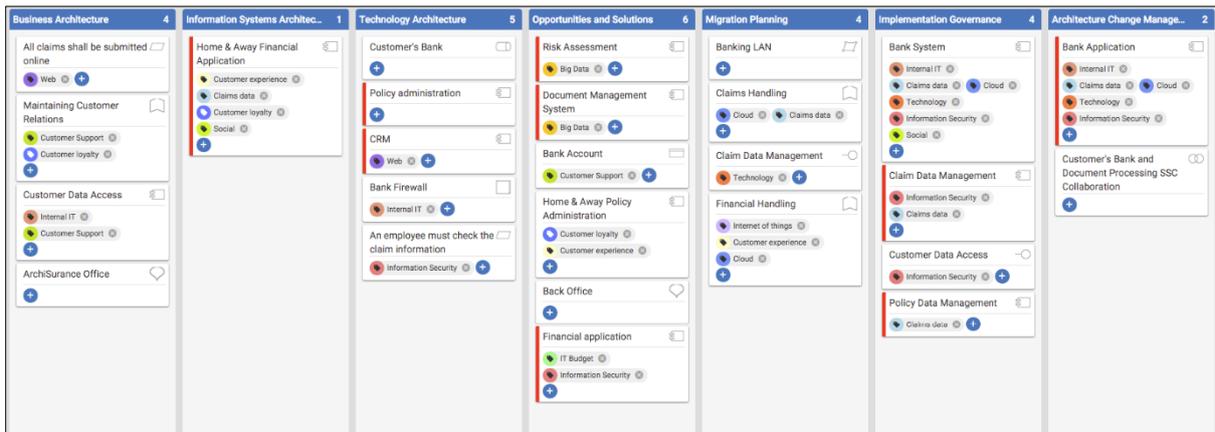
Note, that if the filtered list was required for "aidan kelly" and "martha kent" an **OR** condition would be required. The query above could be reorganised to; Created Date before "05/01/2016" AND Name contains "claim" AND Created By User "aidan kelly" OR Created By User "martha kent".

The example below uses the **Colorize** mode:

View set to color **Application component** object types red that have the **Business Criticality attribute** set to **High**.

The screenshot shows a 'Colorize' mode interface. It has a 'Reset' button, 'Filter' and 'Colorize' tabs, and a red circle icon. The criteria row has an 'Invert' checkbox (unchecked), 'Attribute' as the criteria type, 'Application component' as the object type, and 'Business Criticality (Array)' as the attribute. The value is 'High: 100'. There is a trash icon and an '&' or 'OR' button.

When used on a kanban board, it may look like this;



Objects that match the criteria now have a red marker down the left hand side.

## Saving and Bookmarking

Once a query is saved, it can be added to the tabs menu with the Bookmark button.



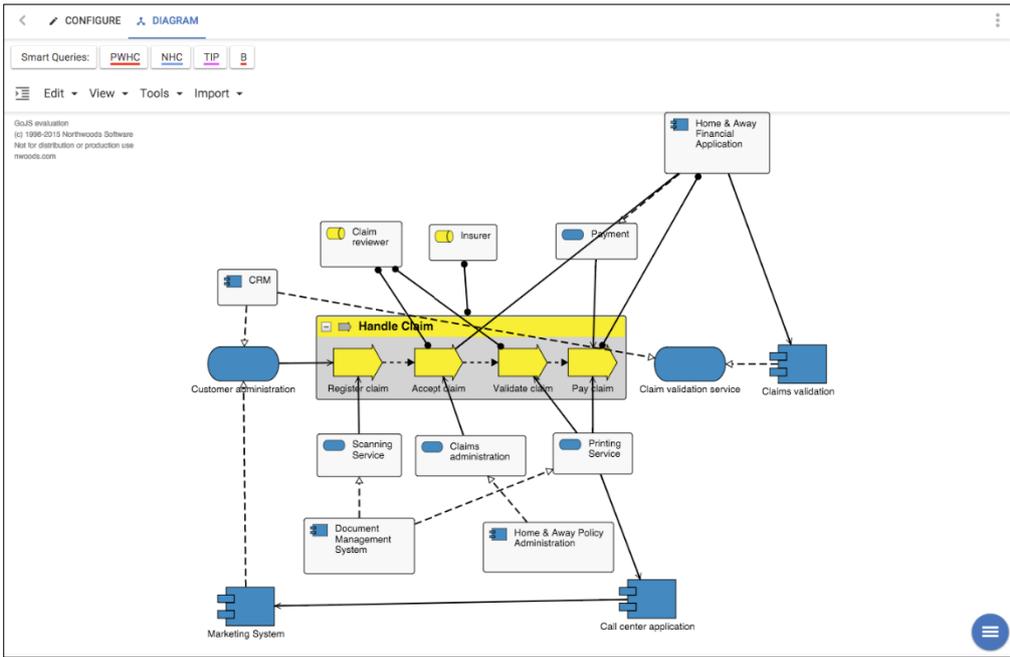
**Bookmarks** are only for the current module. For example, bookmarking a query in a roadmap will not bookmark it for the charts module. However, the query does appear in the list of queries in the charts module and can be bookmarked from there.

Queries are activated by clicking on the button for the query for your current view.

## Smart Query Nesting - e.g. Relationship

Smart queries have been extended to include 'Relationship', to allow users to highlight or filter the view they are seeing, based upon a condition that is related to a related object, not the objects in the current view themselves

E.g. We can filter/colorize the symbols on a diagram, if they are **related** to **Application Components** that have an **attribute** value of **Business Criticality** set to **Low**, to identify possible areas to look for application retirement.



If we create our smart query as normal, then select:

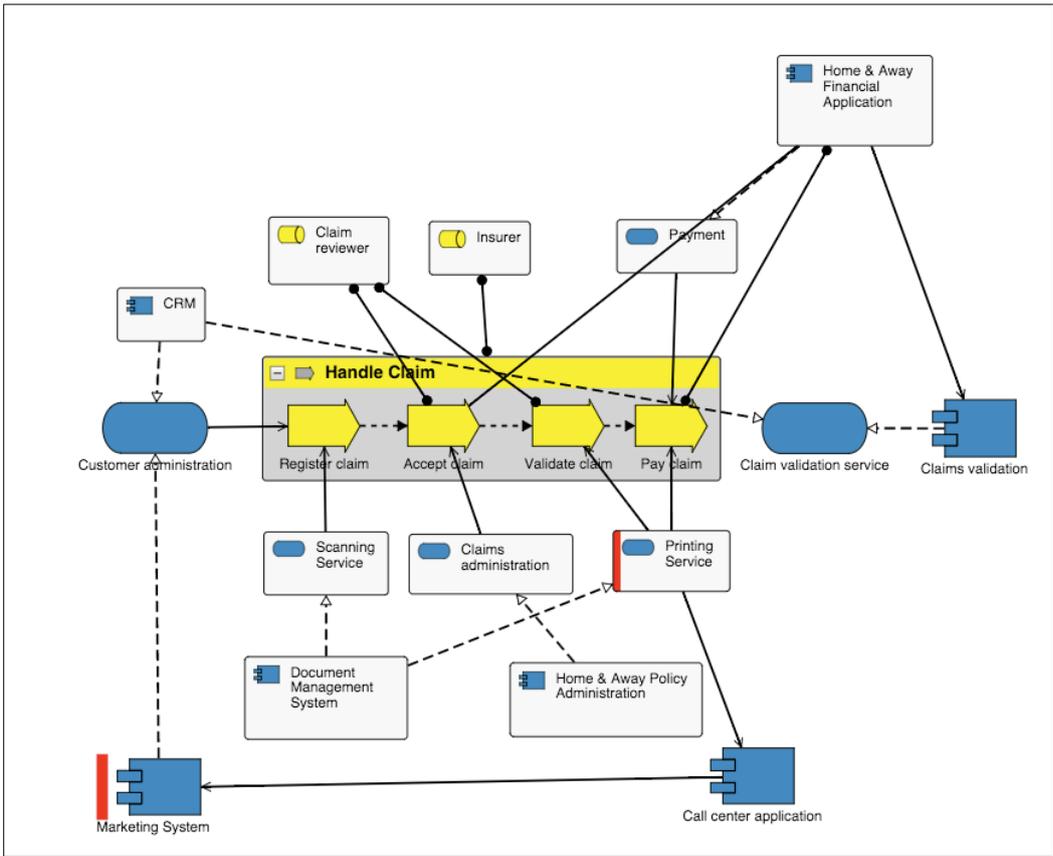
**Criteria Type:** Relationship

**Related To Type:** Select object type that the object is related to (in our example, associated to Application component)

**Relationship:** How is the object related to the chosen object type (or select Any)

Further criteria can be applied, as in the example below; an Attribute of related Application components, Business Criticality, has the value "Low: 10".

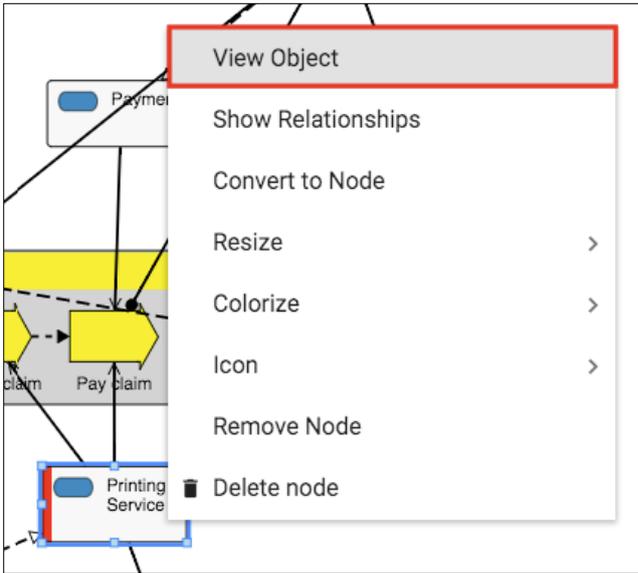
Your results should be displayed as below. You can close the Smart Query box and it will still be applied, clicking the Smart Queries button again will open it with the details still there.



In our example, one Application service has been identified as meeting these criteria - the "Printing Service" Application service, is related to at least one Application component with its Business Criticality attribute value set to "Low: 10".

There are other ways to view the relationship - we will look at the Attributes tab of the related object.

Right click it and select **View Object**.

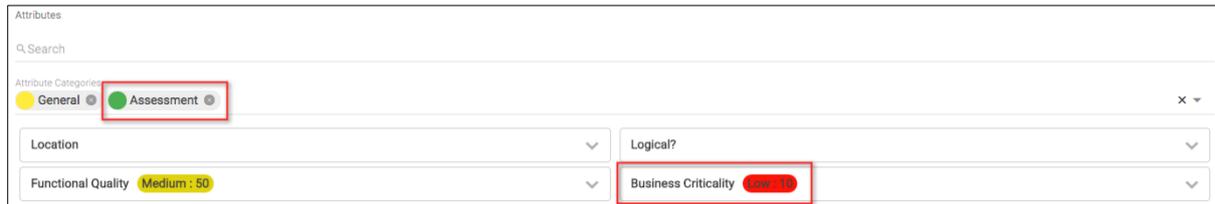


---

Navigate to the **Relationships** tab and see that it has two different types of relationships to two different Application components.

Hover over one of the Application components, Open it, and view its attributes.

Open the attribute category drop down and select the **Assessment** category (if it is not already selected) - find **Business Criticality** and note its value - here it is **Low** and so fits our criteria.



## Query Re-use and Applying Multiple

The platform allows users to use smart queries created by other users, if they are given the correct permissions using communities.

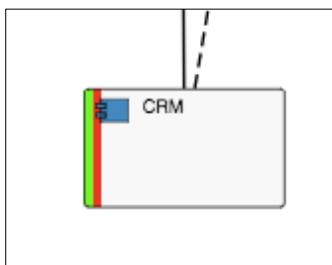
In order to re-use another user's queries that have been made available to you, you can select/bookmark the smart query shown to you in the smart query list.

Users can also create their own version of another user's smart queries, using the 'incorporate another smart query' criteria. A good reason to use this is that the other query can still be managed by the other user. The contents of the query are re-created in your query in this scenario, so if the other user edits or removes their query, it then will not affect yours.

To apply multiple Smart Queries, they will need to be bookmarked. From here simply select the ones you would like, this works with both filters and colors.



As you can see with CRM each object can have multiple colors applied to it.

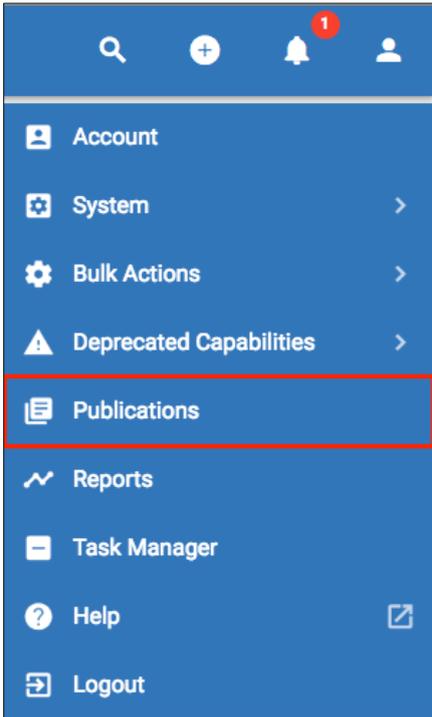


---

# Publications

**Role Applicable to:** [Client Administrator](#), [User](#), [Contributor](#), [Reviewer](#)

The platform provides a capability for producing documents. A publication can be created under the 'Publications' menu.



A publication is a unique document that is stored per user on the system.

The list will show your existing publications. The + button will create a new publication. Give the publication an easily recognizable name as this will be used to refer to the publication later on.

<p>Name</p> <input type="text" value="Upgrade claims systems business case"/> <input type="checkbox"/> Landscape
---

Press the save button at the bottom of the page to save the publication.

You can choose to have the publication in landscape format which will give you more horizontal space.

The publication can be constructed from modules on the platform and custom text (HTML).

Each view in the platform can be added to a publication.

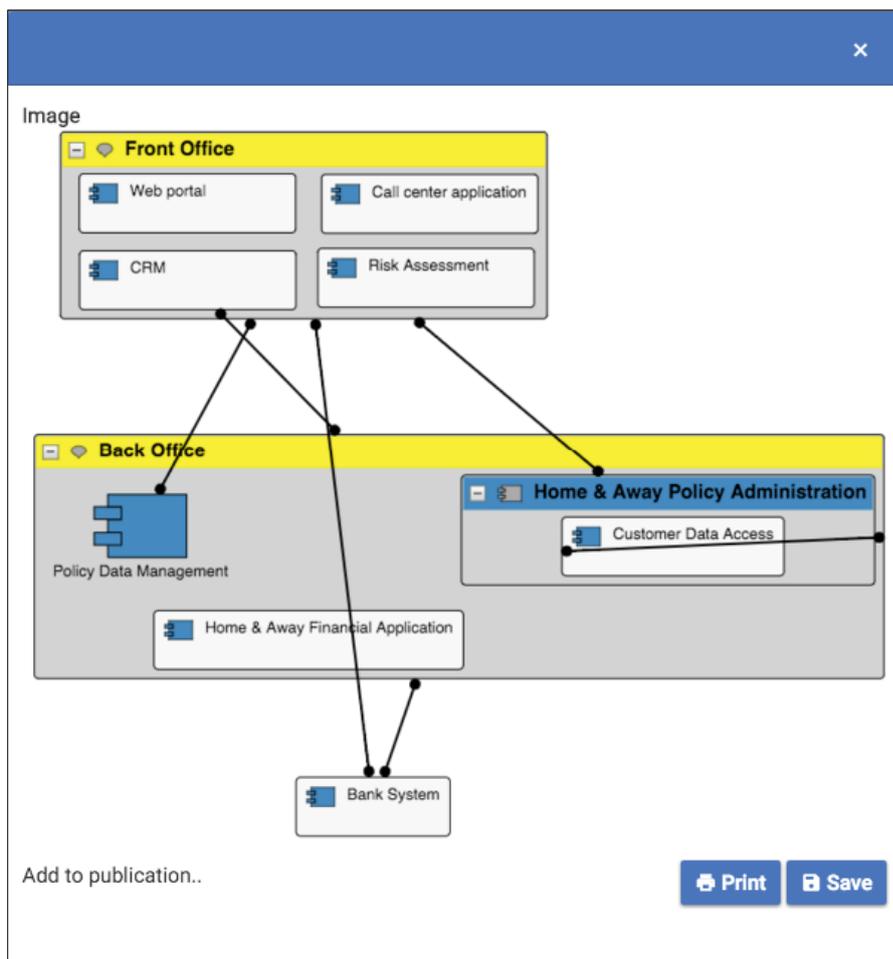
In order to add content to a publication, you need to be in a view on the platform. For example, when on a kanban board, you will see the following button;



Clicking this and then selecting the publication button, shown below, will open the "Publication" dialog window.



The Publications button opens a window that gives you three options for the view, **Add to Publication**, **Print** and **Save**.



**Print** will create a PDF in the a browser window, **Save** will download the image as a .png file.

**Add to publication** will open more options;

Add to publication.. [Print](#) [Save](#)

**Caption**

Diagram: ArchiSurance Application Co-operation

**Publication**

[Add](#)

The **Caption** text will be added under the image in the publication.

**Publication** has a drop down list of your publications that you can select from, to which the image will be added.

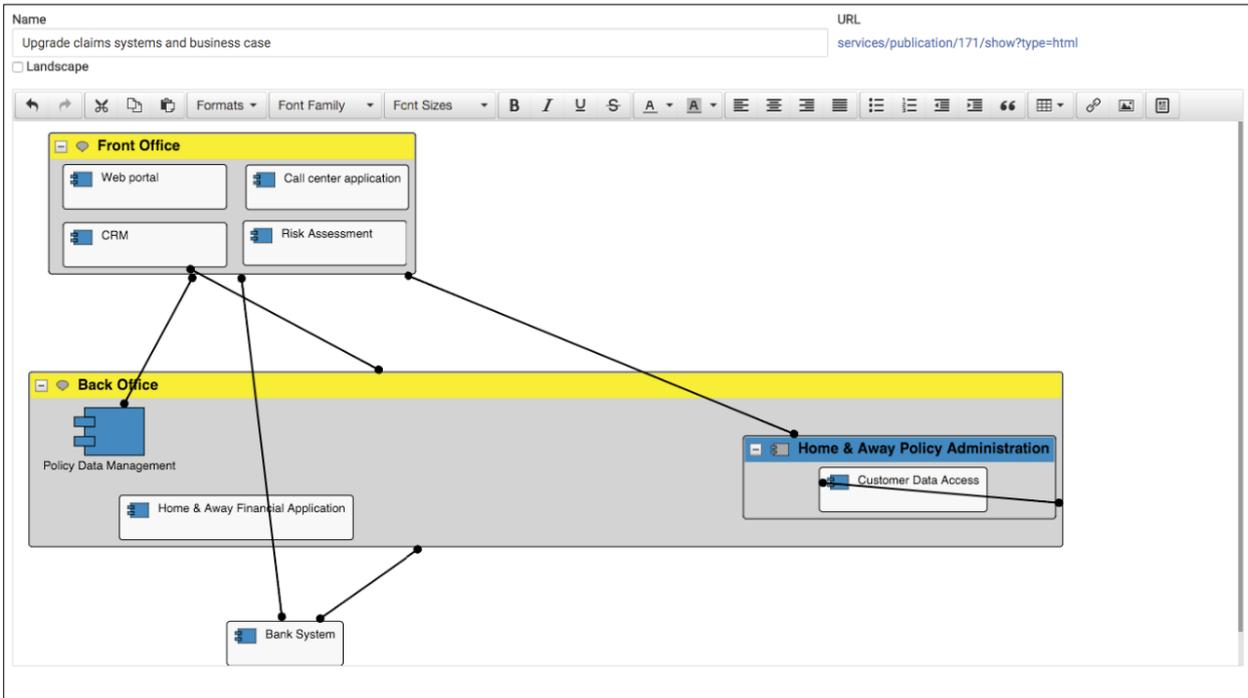
**i** | A caption can be set. It is worth naming the image with a meaningful name. The image can be reused in other publications and is found via the name that you specify as the caption.

Pressing the **Add** button will add the image to the publication.

**i** | The image that will appear in the publication is of the module as it is seen currently on the screen.

When you revisit the **Publications** module, and click on the publication, you will notice that all views that were added to this publication are available.

They are now placed inside an HTML editor where you can embellish the document with further text and graphics to build out your publication.



---

You can generate the publication to a PDF document by pressing the download PDF button.

You can save any changes you make to the publication using the **Save** button.

Using the 'Publication' tab at the top of the page, will allow you to view the changes that have been made to your publication. This is an ideal place to send other users to see your publication.

---

# Notifications

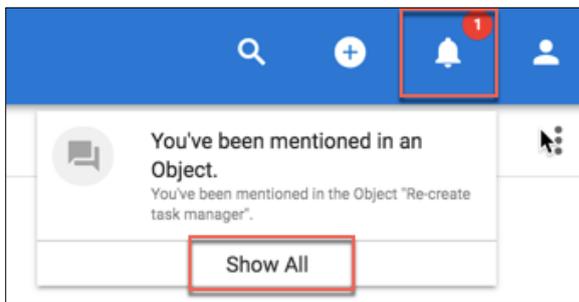
Notifications are triggered when users perform the following actions

- Completion of an erwin Import
- Use **Mention** action within a Comment (on objects, diagrams or views)
- Adding a member to a community
- Adding a user as a Collaborator (on objects, diagrams or views)

When a notification is triggered, the user will be sent both an email and an in-tool notification to alert them that either something has completed or that someone is asking them to do something

The in tool notification(s) will be highlighted by a number icon on the notification menu in the toolbar. This number represents the number of unseen messages

When this is selected, the user will see all unseen messages in a drop down pane - Clicking on the notification may cause an action to occur, depending on what the notification is in relation to e.g. clicking on a 'You've been mentioned in' notification will take the user to the object, daigram or view in question.



Selecting **Show All** will allow the user to see their complete list of notifications - these are still selectable



---

# View Manager

**Role Applicable to:** [Client Administrator](#)

**Views** allow administrators to select a data-set, containing one or many object types, based on chosen criteria, and visualize it in different ways, quickly and easily.

## What is the benefit?

Your data in Agile EA quickly builds up. Eventually, it becomes hard to manage even for the EA expert, and impossible to navigate for everyone else. You can use views to present only 'slices' of the complete set of EA data for different roles or communities.

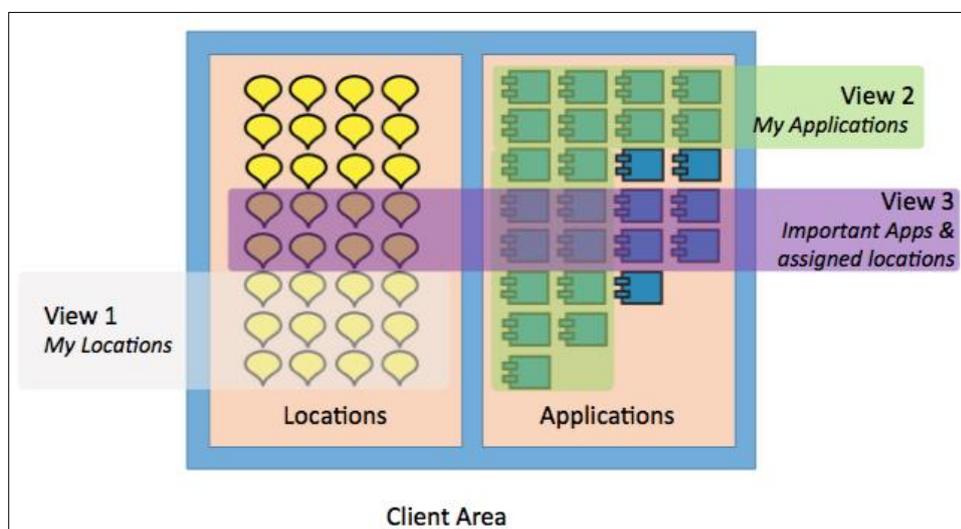
To the EA expert, it is now easier than ever to filter, manage and focus on your EA information. It simply is a better way to interact with your data.

For the non EA experts – you can give them simpler access, where they see less and can zero in on what they should work with.

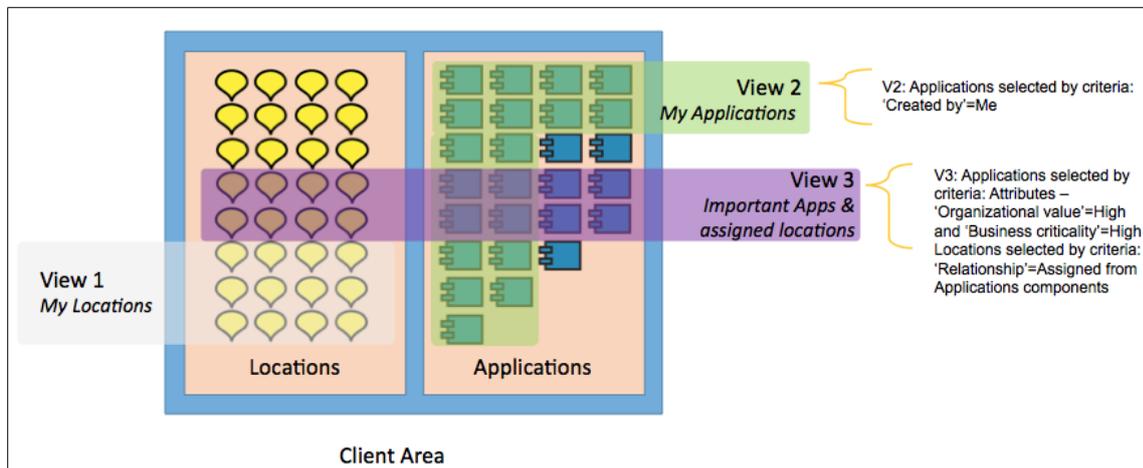
With views, you can easily navigate to e.g.:

- “My Applications”, and see those applications you are responsible for updating or for adding to diagrams.
- “Top 10 Risks” for a given business unit and its processes.
- “Pending Interfaces” to show those that we need to finalise and justify in EA work.
- “My business unit capabilities” – show me those Business Capabilities relevant for my department.
- “...Add anything...” – you can virtually in minutes design your own for ANY object type and with ANY set of rules in the system.

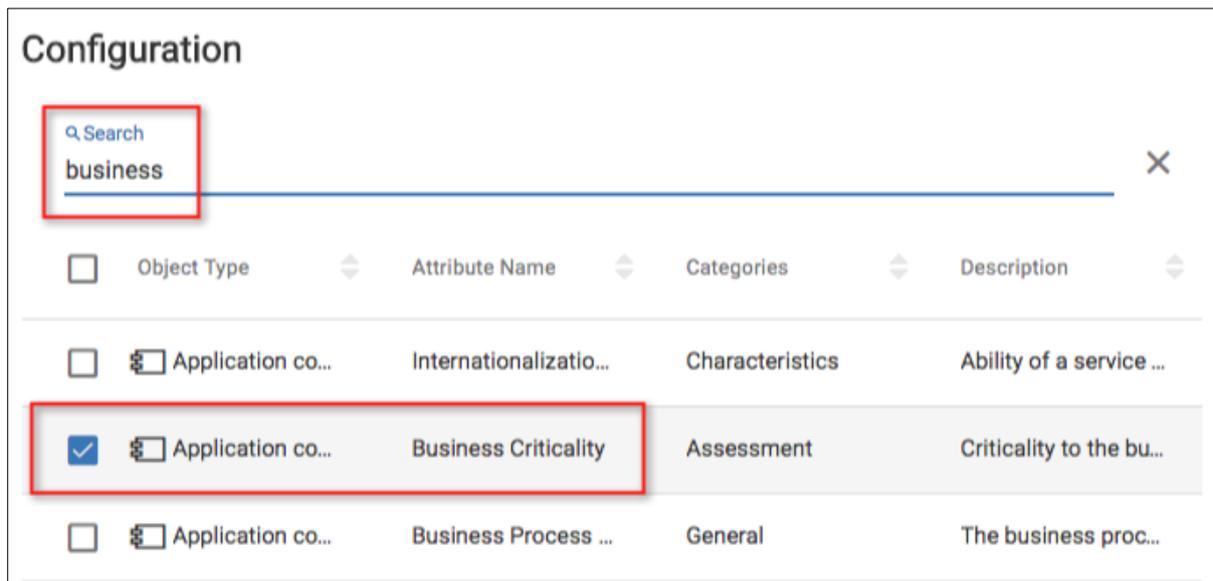
You can ensure that each role that you define, see relevant, actionable EA data on their Homepage via the dashboards.



Views allow users to create a dynamic data-set, that fit a given criteria, doing away with a static data-set that users continually need to add to when new objects are created. Views also remove the need for each visualization data-set to be defined individually within each module of the platform, as a single data-set can be used with multiple visualizations.



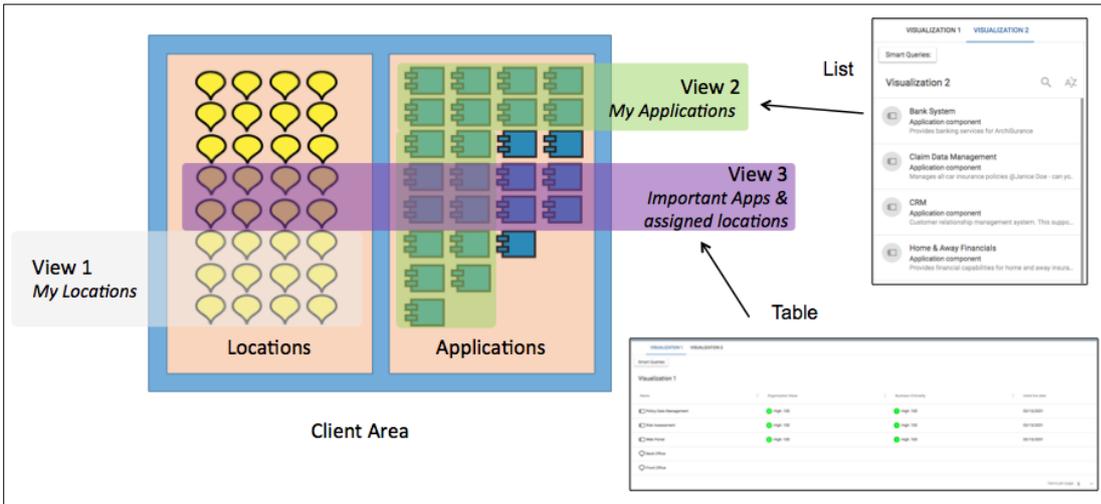
The visualizations within the view are set with the attribute values that will be available for the users.



Users can turn these on or off for the end visualization, making it easier for the user to display only needed information.

Current visualizations supported in the view manager are:

- Tables
- Lists
- Kanbans
- Roadmaps



Closer look at the visualizations;

VISUALIZATION 1 VISUALIZATION 2

Smart Queries:

Visualization 1

Name	Organization Value	Business Criticality	Initial live date
Policy Data Management	High: 100	High: 100	03/15/2031
Risk Assessment	High: 100	High: 100	03/13/2031
Web Portal	High: 100	High: 100	03/15/2031
Back Office			
Front Office			

Items per page: 5

VISUALIZATION 1 VISUALIZATION 2

Smart Queries:

Visualization 2

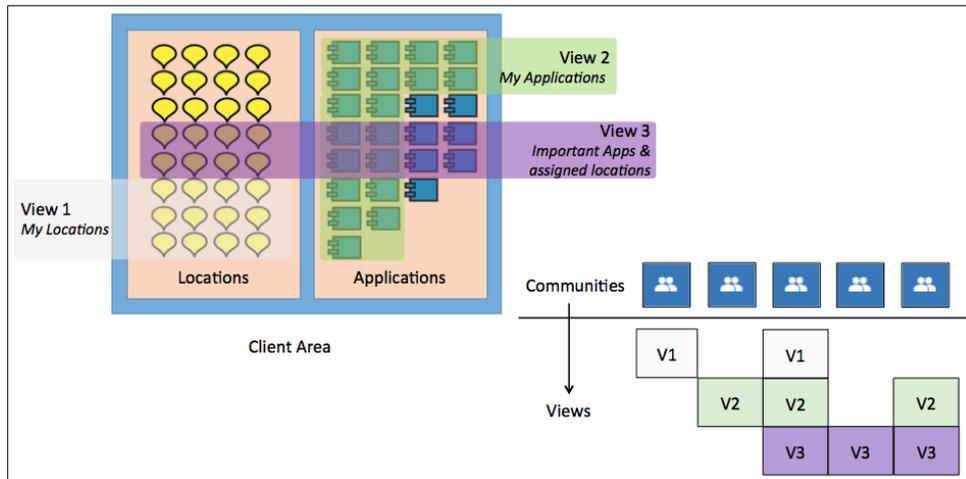
- Bank System**  
Application component  
Provides banking services for ArchiSurance
- Claim Data Management**  
Application component  
Manages all car insurance policies @Janice Doe - can yo.
- CRM**  
Application component  
Customer relationship management system. This suppo...
- Home & Away Financials**  
Application component  
Provides financial capabilities for home and away insura...

Views are created by administrators and are controlled by the communities that they are assigned to. Only members of an associated community can see and access the view.

Communities can have access to a number of views. Because different users have access to different views, through community permissions, view menus can appear different for users.

Users still need to have permission to access the data shown within a view or one of its visualizations – this is controlled at Object level, not by the view itself.

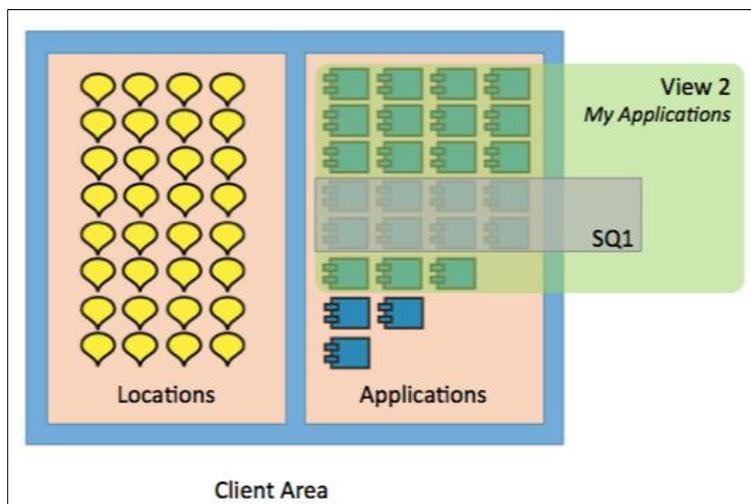
If the user does not have access to the data, it will not appear within the visualization for them.



## Further filtering of views with Smart Queries

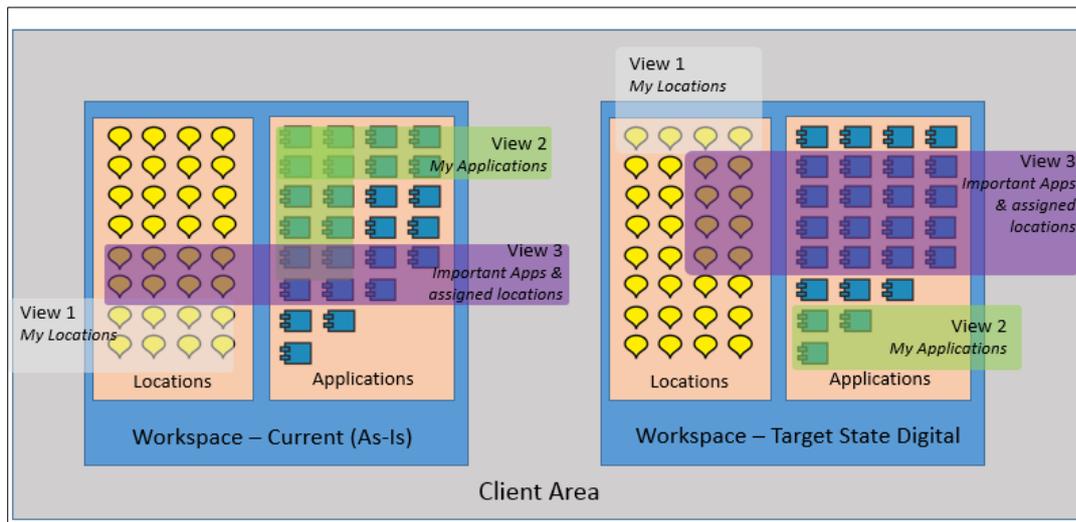
An accompanying feature to defining views is that the user can use a smart query to select/filter objects of interest within a view (also outside of it). Using smart queries can help with the admin overhead of needing to create lots of individual views. Smart queries can make a view fit multiple uses for many users.

Criteria are used to filter out which objects will be displayed when the query is applied to the view. By using smart queries each user can configure their own selection of objects and also then decide which attributes should be visible at the view level.



## View and Workspaces

- IMPORTANT:** Views are independent of workspaces - Visualizations created within views will show the dataset from the workspace that you are currently in - if you change your current workspace, then the visualization will change to display the new dataset for the view query. This saves users having to define views for each workspace. If you use views on the dashboard, then the workspace that the dataset is being seen from can be changed using the drop down at the top of the visualization.

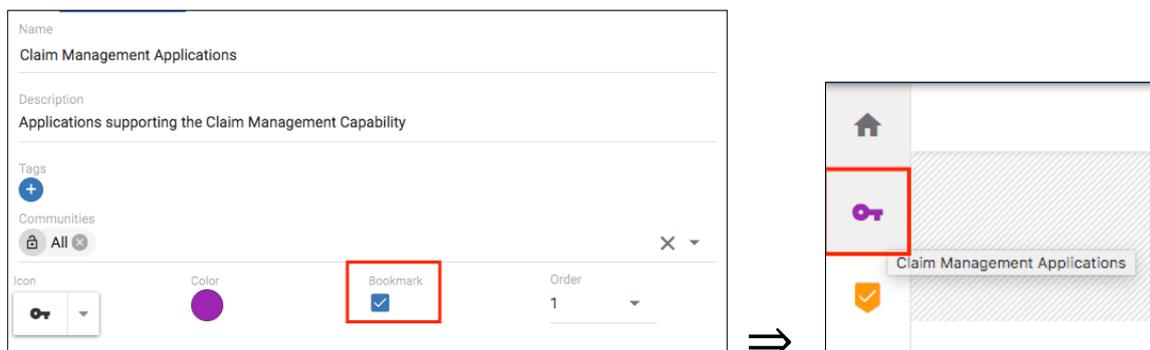


## Make views available to other users

Views can be displayed client wide on the left hand menu by administrators. They can do this using the 'In Menu' checkbox within the View Manager configuration.

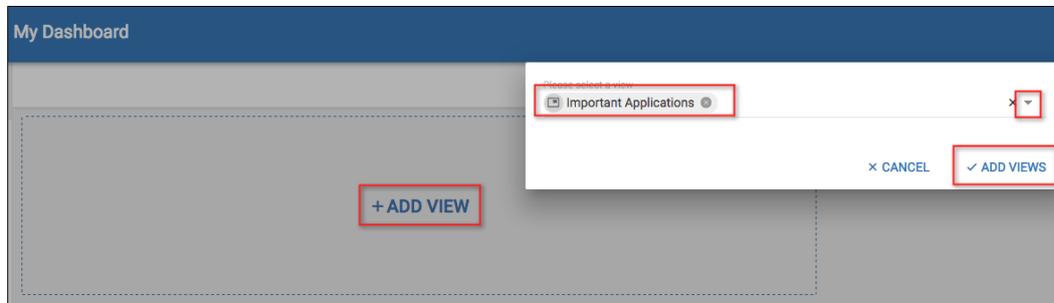
Any views on the menu, will be available for all users within the client area, that belong to the assigned communities.

Views on the menu can also be ordered here for importance.

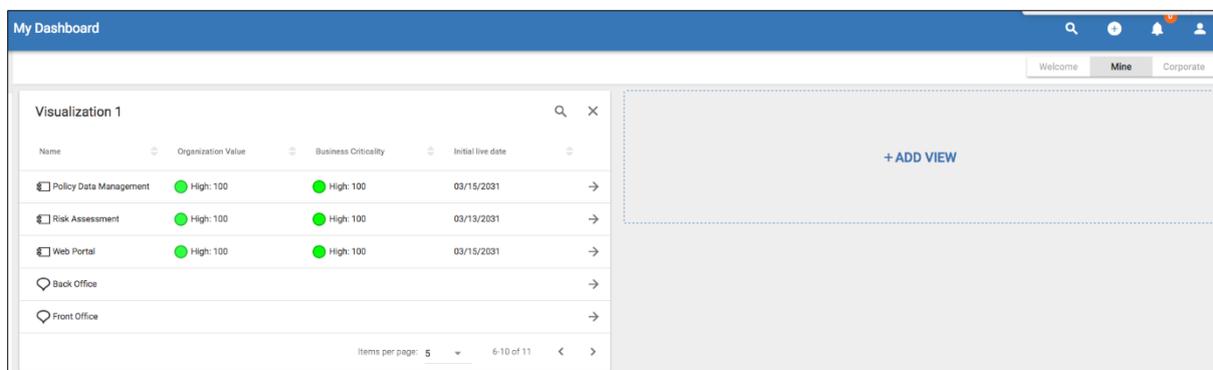


## Use Views on the dashboards

Views now replace the current dashboard - This will no longer be in use. The dashboard 'Mine' now allows users to add existing views (or views that they are allowed to see) from a personal perspective.



Currently, the dashboard will display the first visualization from a view (if there are multiple).



Administrators can make views available on the 'Corporate' dashboard also. All users in the client area can then see these views, but cannot edit them. This gives admins the opportunity to define the criteria and the attributes to show to all users.

## Creating a View and selecting data

- Select the **Views** in the left menu.



- Select the **Add** button to create a new, empty View.



- Fill in the view details on the **CONFIGURE** page - example below;

**Name:** Important Applications

**Description:** Application with high value and business criticality

**Icon:** Select an appropriate icon (this will be seen on the menu if chosen)

**Bookmark:** Check the checkbox if you would like the menu item to appear for users that have the correct permissions

**Order:** What position would you like the menu item to appear

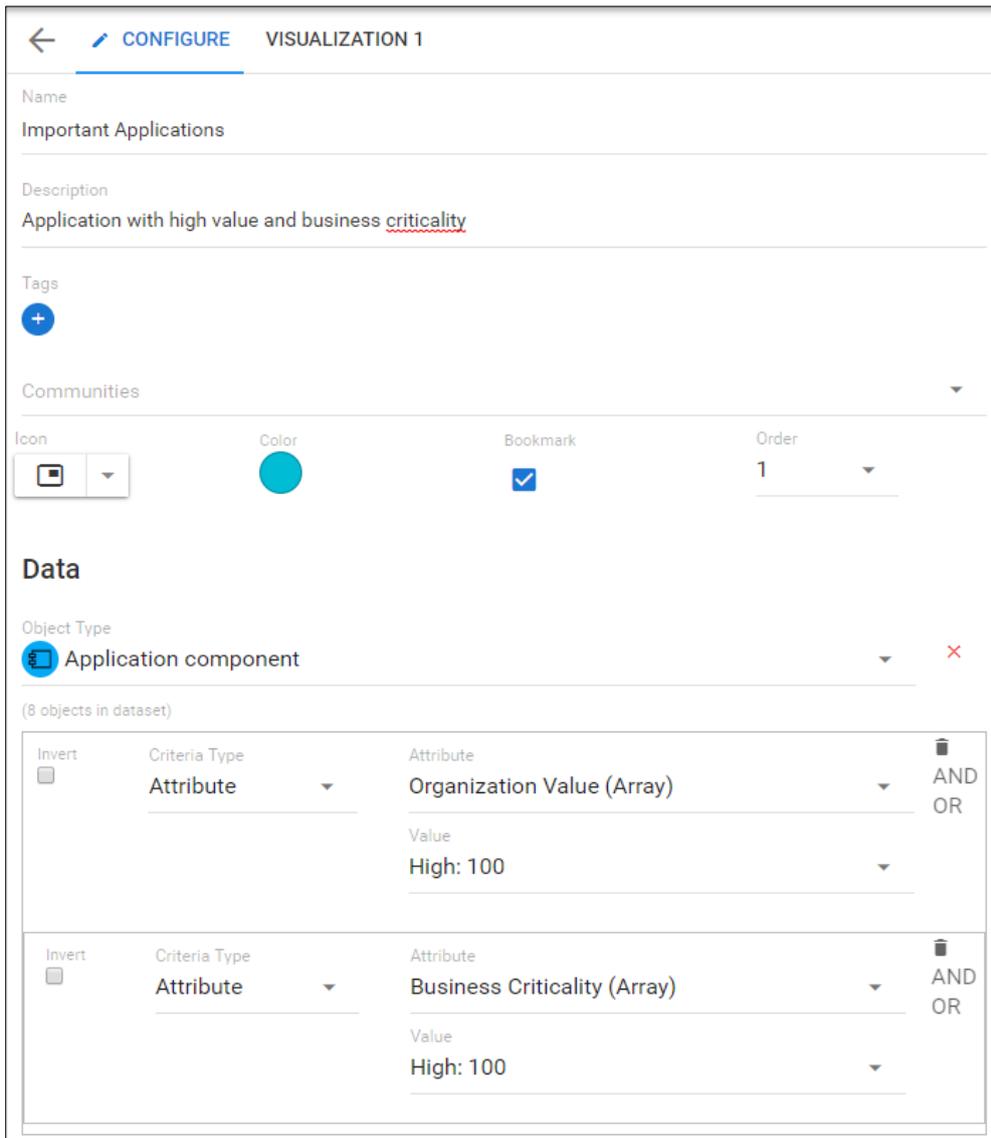
Select the dataset that will be used within the views...

**Data** - here we build the rules/criteria to narrow down the data set that we want - we can choose single or multiple object types and then dive deeper into criteria that the model has about the instances of the types

**Object type:** Select an object type (or multiple - using the Add Object Type button) - example below uses **Application component**

**Criteria Type:** Select a criteria - current choices are '**Name**', '**Attribute**', '**Relationship**', '**Has Tag**', '**Created Date**' and '**Created by User**' - example below uses **Attribute**

**Criteria options:** Select the criteria options that relate to your chosen criteria type - example below is to select specific values that must be met for 2 attribute arrays, using the **AND** to select a 2nd attribute criteria type.



We have now created a view that will pull out all of the Application components that have "High: 100" Organization Value, and "High: 100" Business Criticality. We now need to decide how we would like to visualize this data and what information we would like to pull out and display to users.

## Table Visualization

On the right side Visualization tab, we can see the two currently supported visualizations, Table and List. (Kanbans and Roadmaps have been introduced more recently.)

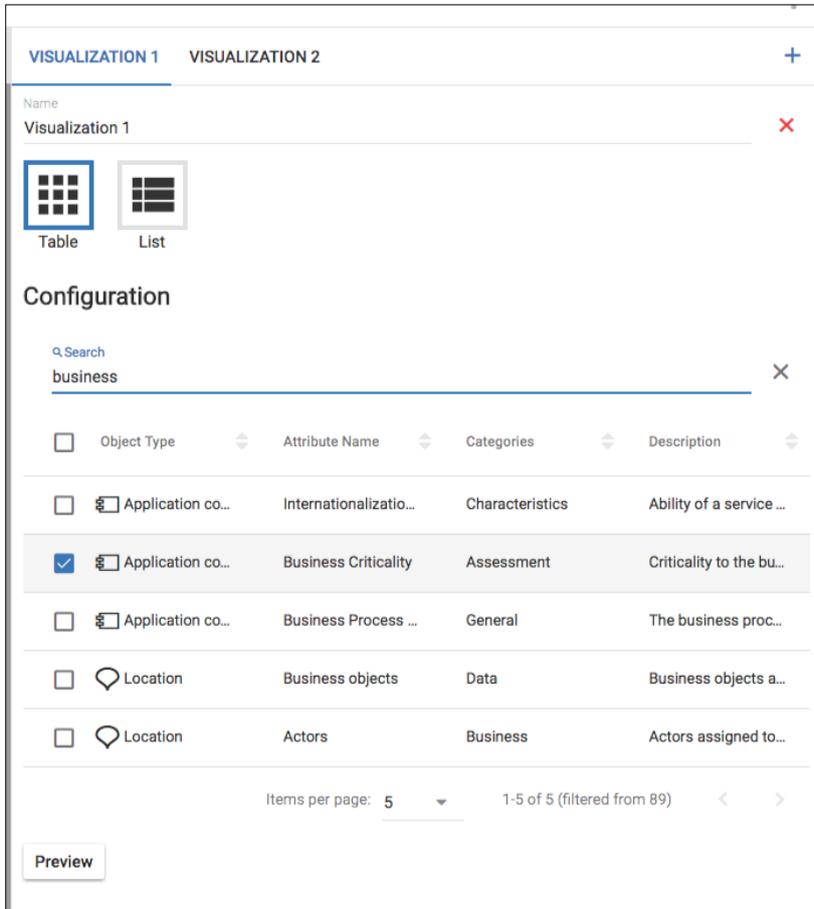
- Select **Table** from the visualization choice.

To add multiple visualizations, select the + button in the top right hand corner - more VISUALIZATION 'N' tabs will appear.

In the **Configuration** section;

- Use the **search** button to type and look for attributes that you want to appear on the table visualization. You can search using the attribute category name also. Select the **check box** to the left for the attributes you would like to use.

Example below chooses to display Business Criticality, Initial Live Date and Organizational Value attributes. So although our data-set was based on meeting 2 criteria, we can display any data attributes that we want to from this data-set, to the end user. These can be deemed visible or not by the end user when using the visualization.



- Use the **Preview** button to see what the output will look like.

The screenshot shows the 'Visualization 1 Preview' window. It displays a table with the following data:

Name	Organization Value	Business Criticality	Initial live date
Berk System	High: 100	High: 100	11/12/2001
Claim Data Management	High: 100	High: 100	01/04/2001
CRM	High: 100	High: 100	01/10/2001
Home & Away Financials	High: 100	High: 100	05/15/2031
Marketing System	High: 100	High: 100	05/15/2031
Policy Data Management	High: 100	High: 100	03/15/2031
Risk Assessment	High: 100	High: 100	03/13/2031
Web Portal	High: 100	High: 100	03/15/2031

At the bottom right, there is a pagination control showing 'Items per page: 10' and '1-8 of 8'.

- **Save** the View.

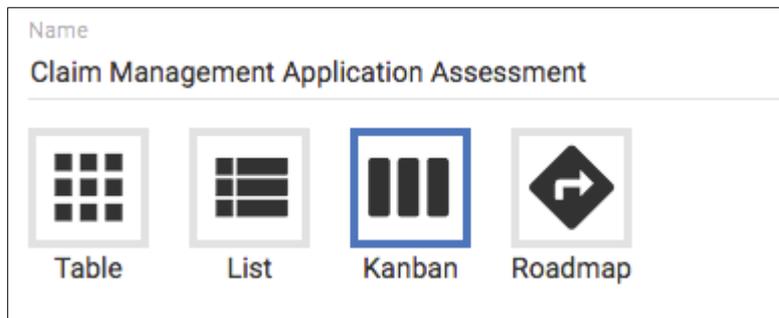
---

**i** | If you choose the list attribute, there are no attribute options to be chosen, it is displayed in standard list format, similar to what you see when you select the Objects menu.

Once saved, the view will appear in the View Manager list and can be edited from there (by admins). The view can be accessed from the menu (by users), if the check box was selected. The view is also selectable on a dashboard, if the user has permission to see it.

## Kanban Visualization

In addition to Tables and Lists, there are also Kanbans and Roadmaps.



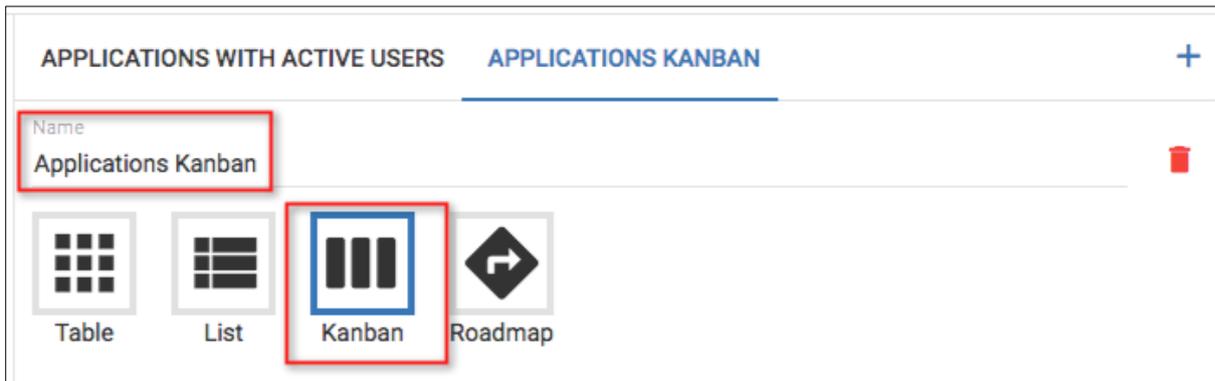
Kanban boards are the visual storyboard for a process or workflow. They represent the journey, and the objects within that journey. Objects are represented as cards on the board. Objects may be moved from one stage to another by dragging. For more information on using a kanban click [here](#).

To Create a kanban from an existing view, we simply select to create another visualization on the chosen dataset (we could create a whole new View if we wanted to and there were different data needs).

- In the View Manager, select a view from the list
- Select to add a new Visualization, using the '+' symbol in the top right of the right pane



- Select the new type to be **Kanban**
- Replace **Visualization 2** title with a **title** for the kanban



In the Configuration section, we select the object type that can be added to the kanban, and then the attribute within that object type that we would like to use to define the stages we will see in our kanban. Currently kanbans are limited to a **single object type** from our dataset and an attribute that defines the stages must be of type **Array** or **List** (users can create their own custom Array and List attributes for defining kanban stages).

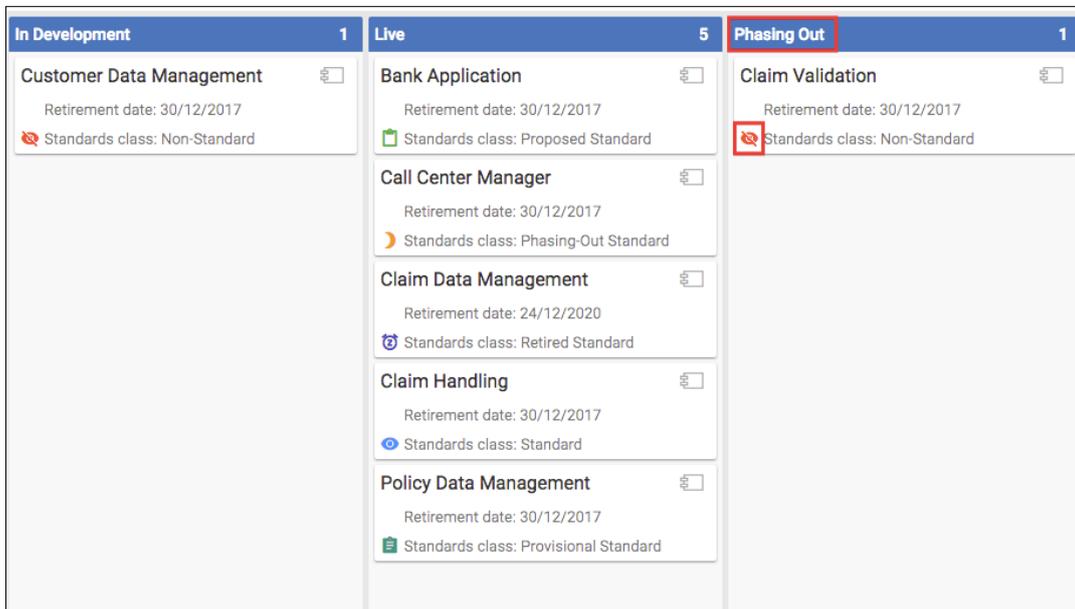
- The object type will be set to **Application component** (because this is the only type in our dataset)
- Select the Stage Attribute to be **Lifecycle Status**



- You can then select the attributes you might like to display on the kanban board - select the below
  - Action Cost Savings
  - Number of Users
  - Active Users
  - Initial live date
- **Save** the new Visualization
- Select the **View** from the left hand menu



- Once in the view, select the **APPLICATIONS KANBAN** tab to enter the kanban.



We can see that this is a very simple kanban board, allowing us to track progress of objects, based on a certain attribute - we can see that the applications are all in either the **Live** or the **Phasing Out** stages. This will map to the attribute that is set within them.

Here you can see a kanban in the view, each stage title is one of the available values for the attribute that was chosen.

The chosen display attributes are visible on each card and also display any colours/icons applied to them where appropriate. If you choose to display minimal attributes here, only the colours/icons will be displayed and you will need to hover over them to see the values.

Moving an object to a different stage will update the attribute value, and updating the attribute value will change the stage of an object.

## Roadmap Visualization

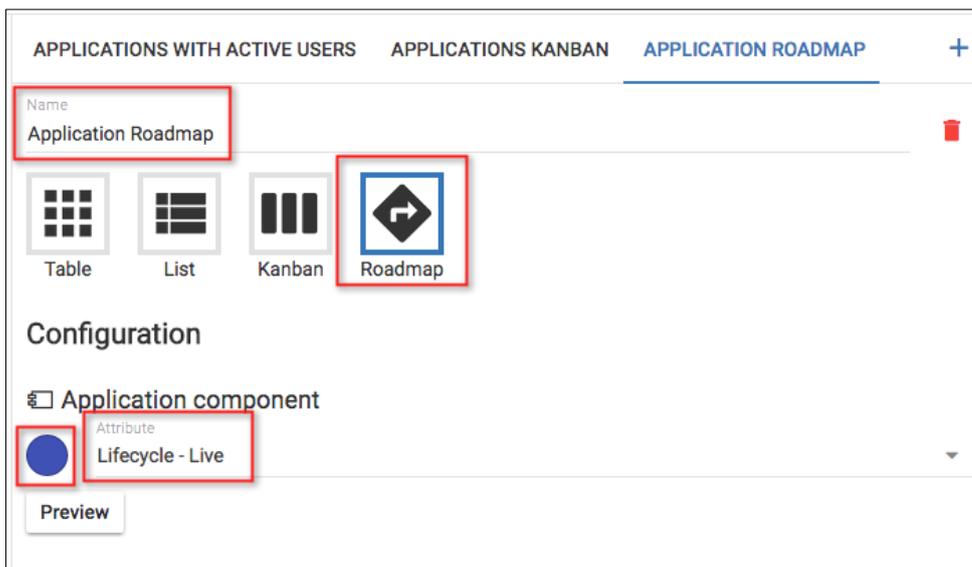
Roadmaps provide a mechanism to show any object over a time period. Most Enterprise Architecture (EA) objects will have a time dimension. Where objects don't have a time attribute, you can add them through the customization capabilities.

Roadmaps are now a visualization with the View that we created earlier, on our pre-defined dataset. To create a roadmap from our existing view, we simply select to add another visualization on the chosen dataset (we could create a whole new View if we wanted to and there were different data needs).

- Select the View window and select the **CONFIGURE** tab
- Add a new Visualization, using the '+' symbol in the top right of the right pane



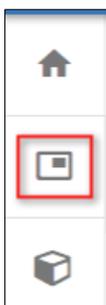
- Select the new type to be **Roadmap**
- Replace **Visualization X** title with a **title** for the view
- Choose a **color** - this will determine the color of the roadmap bars as seen in the view
- Select the roadmap attribute to be **Lifecycle - Live** (must be a date/date-range attribute)



**i** If we had more than 1 object type in our dataset, we could select an attribute from each object type, to get them to display in separate lanes on the roadmap.

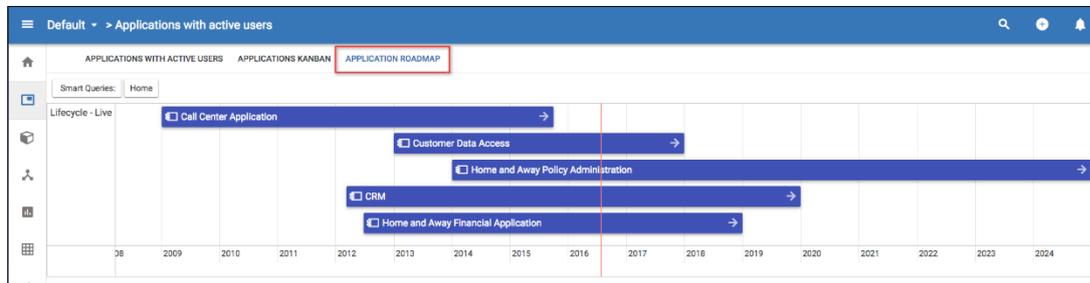
You can use the preview if you wish.

- **Save** the new Visualization
- Select the **View** from the left hand menu

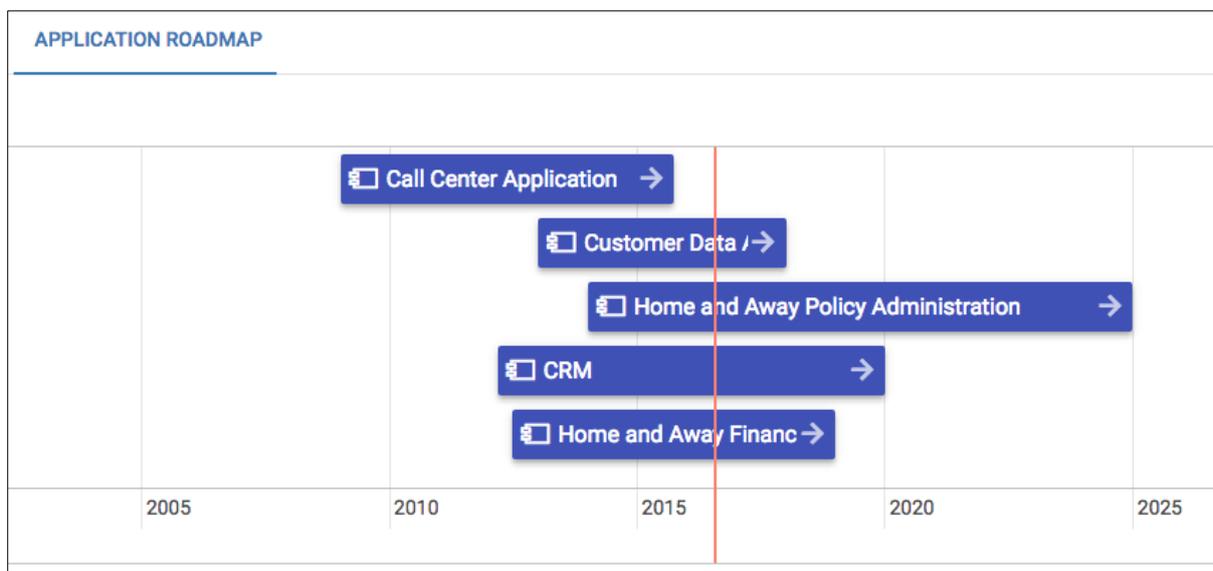


- Once in the view, select the **APPLICATIONS ROADMAP** tab to enter the roadmap

This roadmap shows us the time frames that the Application components are in, for the Live lifecycle status phase - clicking the arrows at the end of the colored blocks will open the object details.



The date scale can be changed from within the roadmap by 'pinching' the touchpad on a laptop, or by using a scroll wheel on a mouse.



**i** Only the objects that have a value in the selected attribute will show up on the roadmap.

**i** You can drag the bars, to change the date and the date-range within the attribute of the chosen object - to do this, select the object (click anywhere on it, other than the arrow), the object will appear to sink slightly, then either select the object and drag (date), or select the desired end and drag (date-range).

---

# Workspaces

Enterprise architecture often begins with modelling the enterprise model in an 'as-is' state (or sometimes just a target state). From a current state, you can model using an iterative process, where you capture the model at various moments in time (potential 'to-be' states), allowing the flexibility to try out different change scenarios and analysis, taking different work streams and project down different routes from the common 'as-is' root. Agile EA enables you to perform this modelling using Workspaces, within a client area, which you can use to suggest scenarios, dedicated team areas or projects.

Important information about workspaces include the following:

- There is only one default Workspace for a client area.
- All workspaces in the client area, share the same underlying metamodel.
- Users can only work in one workspace at a time.
- Teams can collaborate on the same Workspace.
- Workspaces can be assigned to Communities, for access control
- Workspaces are blank after creation
- Model Data can be moved between workspace via export/import

## Agile EA Concepts

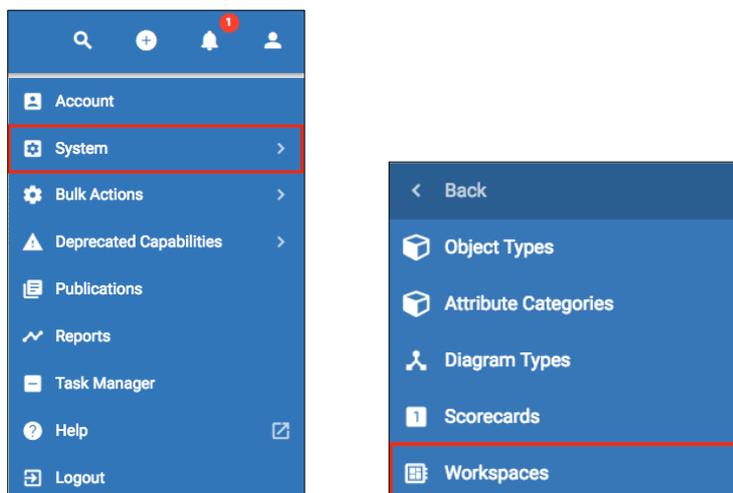
### Client Area

A client area is the unique client area available to you as a client.

A client area contains diagrams (views), objects and relationships. By default, each client area has a default workspace which is a namespace for all views and objects.

### Workspace

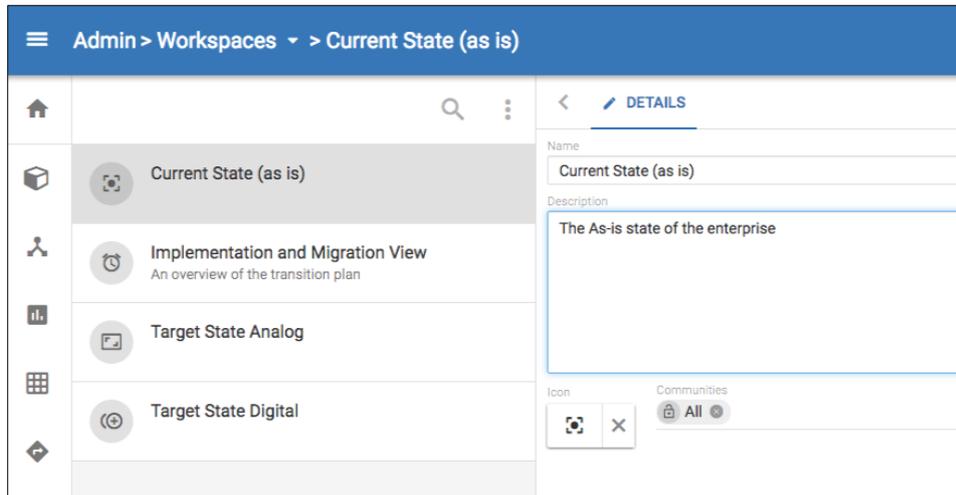
Client areas support workspaces. Workspaces may be set up by an administrator using the **User Menu > System > Workspaces option**.



---

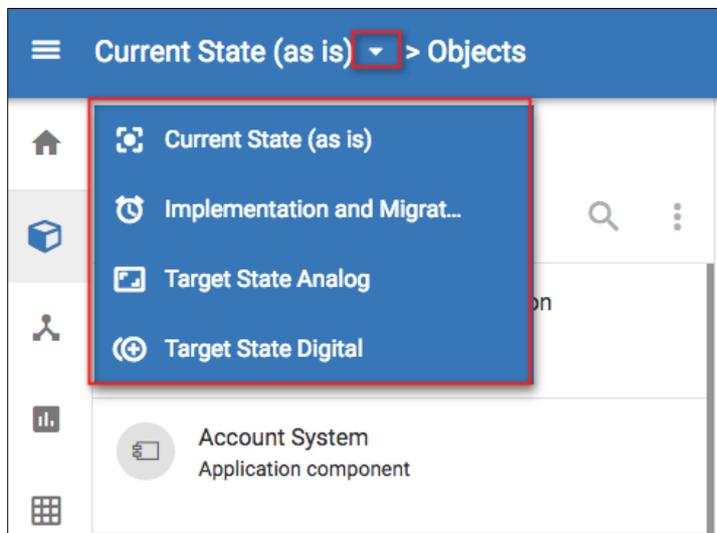
Workspaces are extremely useful for modelling architectures within specific bounds and for modelling current and future states.

Workspaces are created in a flat structure and the tool allows users to navigate between workspaces at ease.



Whilst working within a workspace, users can select the drop down in the toolbar and select a different workspace to work in.

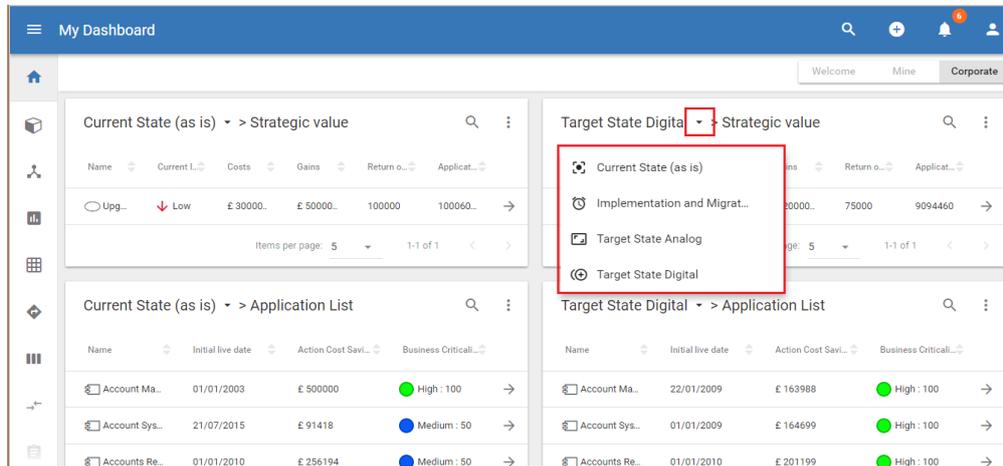
If you are within a module (i.e. Object list), you will stay within that module and be presented with the data that exists within the chosen workspace.



## Showing workspace content in a dashboard

View manager can be used to construct object views which can then be shown in the dashboard. The dashboard segregates views by workspace.

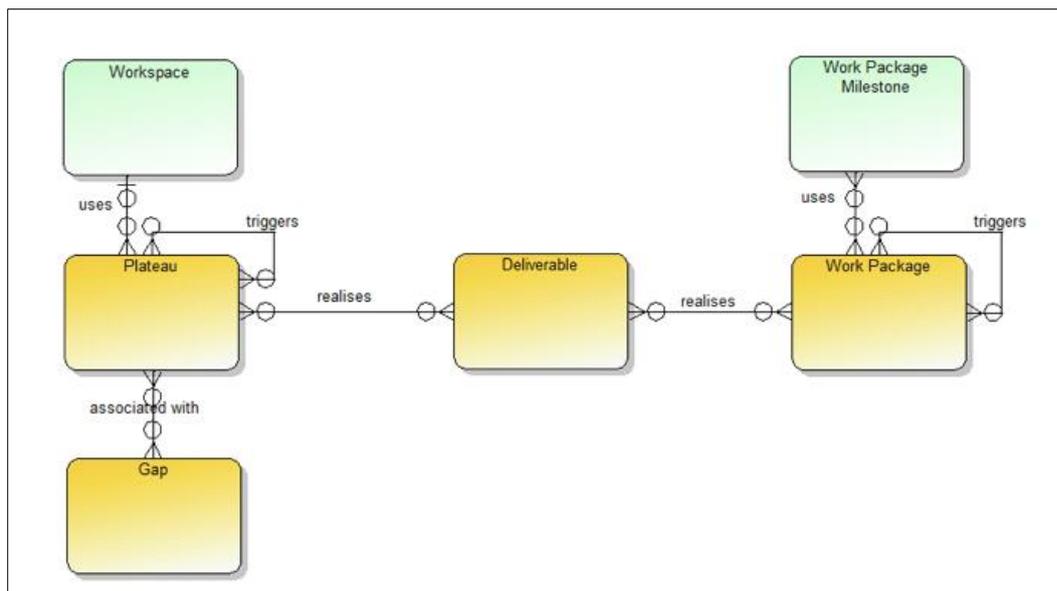
This is useful for showing current v future state rollups and views across workspaces.



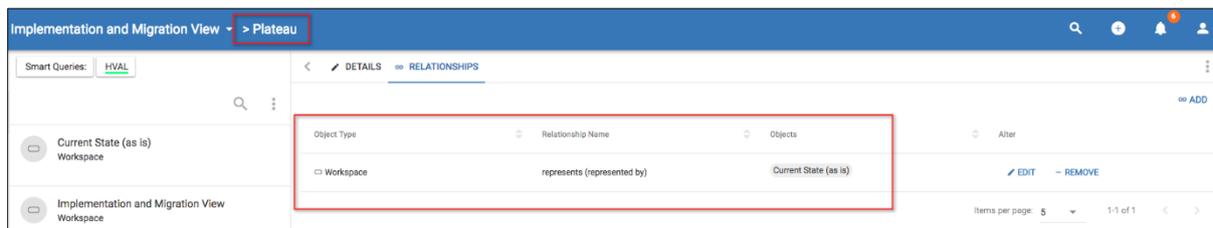
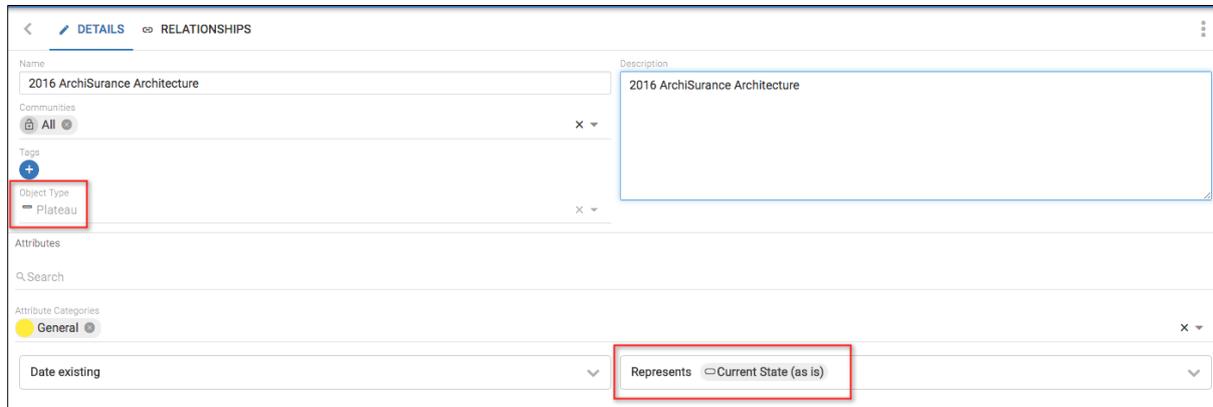
## Extensions to ArchiMate to support workspaces

In order to make ArchiMate pragmatic for roadmapping, we have extended the core metamodel to include both workspaces and work package milestones.

A Plateau is mapped to a Workspace object which itself represents a workspace. A 'Workspace' model object will be created/removed when a workspace is created/removed - this gives users the ability to model their workspace architecture.



Plateaus can be connected using the 'Represents' relationship list, which creates a relationship to the workspace object that it is 'representing' - This will also populate the return view, creating a relationship from the workspace object to the Plateau, using the 'represented by' relationship.

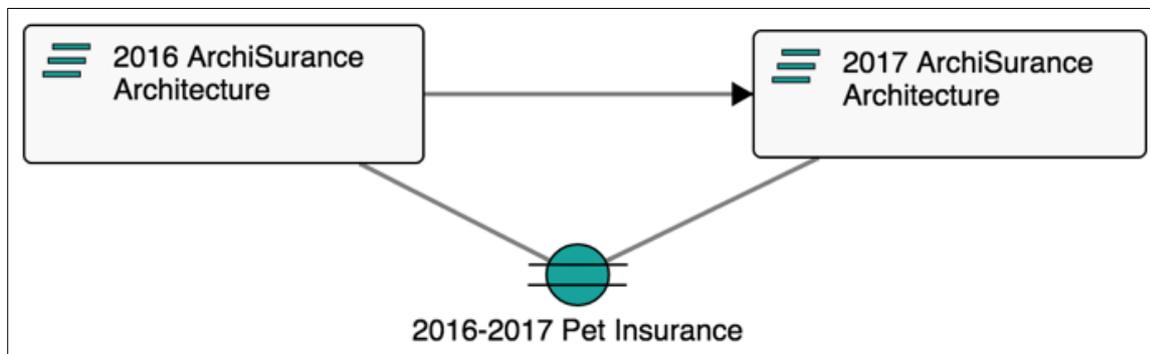


## Current v Future State Architectures

With the additional Object types introduced, we can build a view of our architecture using plateaus and gaps.

In erwin Agile EA, we use an ArchiMate Implementation & Migration diagram to represent the view.

**i** TIP: It is good practice to create a separate workspace that contains overall transition plan models.



In the example above, we can see that the plateau '2017 ArchiSurance Architecture' is triggered by '2016 ArchiSurance Architecture'. The Gap between these two plateaus is 2016-2017 Pet Insurance.

---

# Roll-ups

Roll-ups are an attribute type that allow data to be pulled in from related object attributes, and 'rolled-up' into aggregated values.

Roll-up attributes can be dates, date-ranges or numeric value. These values can be combined with scorecards to perform complex calculations.

Our example uses work packages. We would like in our parent work package to be able to tell;

1. If we have enough resource availability to be able to complete this parent work package and the child work packages utilised within
2. The overall timeframe for the parent work package

In this example we will use work packages to pull through values (resource requirements and durations) from other sub work packages, related using the comprises relationship. Our roll-up attributes will be 'overall resource requirements' and 'overall duration'. Once we have our overall resource requirements, we can use a scorecard to work out if this number is within the resource capacity we have for the parent work package, or if we are in deficit and this needs to be looked into.

We can then roadmap our work packages and use smart queries to visually identify parent work packages that do not have the needed resource capacity to be able to complete the package.

## **Our example uses**

### **Base attributes (created within work package object type)**

- Resource Required (Integer)
- Capacity (Man Days) (Integer)
- Duration (Date-Range)

### **Roll-up attributes**

- Overall Duration (Date Rollup)
- Overall Resource Requirements (Numeric Rollup)

### **Scorecard Attributes**

- Availability (Scorecard)

### **Define a Roll-up attribute**

- Within an object type, create a new attribute and select for it to be of type Numeric Rollup or Date Rollup
- Name the new Attribute and provide a description
- Select the relationship type that you would like to use, to traverse the model into the objects that you would like to pull values from (single relationship type)

- Select the object type that you would like to pull values from (single object type - the list is filtered to only those valid for the chosen relationship type)
- Select the attribute within the object type that the value you want resides in
- Select the aggregator - what you would like to do with values - for Numeric, options are Sum, Max, and Min, for Date, options are Max and Min, and for Date-Range, Range

**Overall Resource Requirements (Numeric Rollup)**

Attribute Type  
Numeric Rollup

Name  
**Overall Resource Requirements**

Description  
**Overall resources needed for the work package (man days), aggregated from the comprised work packages**

Relationship Type  
**comprises**

Object Type  
 **Work Package - comprises (is part of)**

Attribute  
**Resource Required (Integer)**

Aggregator  
**Sum**

Categories  
**Aggregates**

Ensure that the objects at the end of the relationship have values within their attributes (to be rolled-up).

The roll-up attributes will then pull values from any object that fits the relationship and type specified - in our example it will pull the 'Resource Required' attribute values from all work packages comprised from our parent work package, and SUM them.

Below are the 3 sub work packages that fit our relationship.

Relationship Name	Object Type	Objects
delivers (is delivered by)	Business capability	Claim Management
assigned from (assigned to)	Role	CDO, Customer Experience Manager
comprises (is part of)	Work Package	Upgrade Claims Handling, Office 365 Upgrade, Extend CRM Lifespan

Items per page: 25 | 1-3 of 3

The date roll-up will gather the earliest and latest dates from the sub work packages that fit the relationship criteria specified.

Once we have these roll-up values, we can combine them with a scorecard to perform more complex functions - in our example we will take our 'Overall resource requirements' value away from the capacity

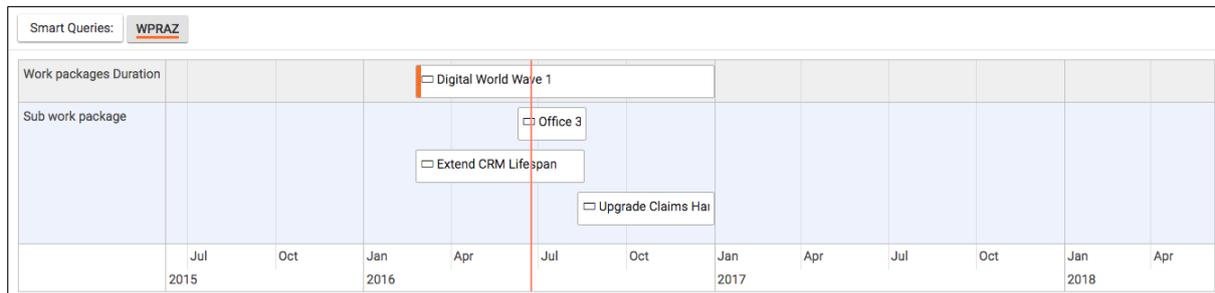
---

value that has been set for the work package - we will then have a + or - value that will identify if the work package can be completed.

In our example

- Create a new scorecard with 2 simple metrics and the formula <1>-<2>
- Create a new attribute that uses the scorecard and uses existing attributes - set them as <1> being 'Capacity' attribute and <2> being the 'Overall resource requirements' roll-up attribute

We can then see the resource availability for the parent work package shown by a smart query on a roadmap (where resource availability scorecard attribute is < 0), that shows the overall duration of the smart query and also the individual durations of the sub work packages that are 'part of' the parent work package.



**i** Extending the earliest date or the latest date of the sub work packages, will cause the roll-up attribute to be re-calculated and may change the overall duration dates on the fly.

---

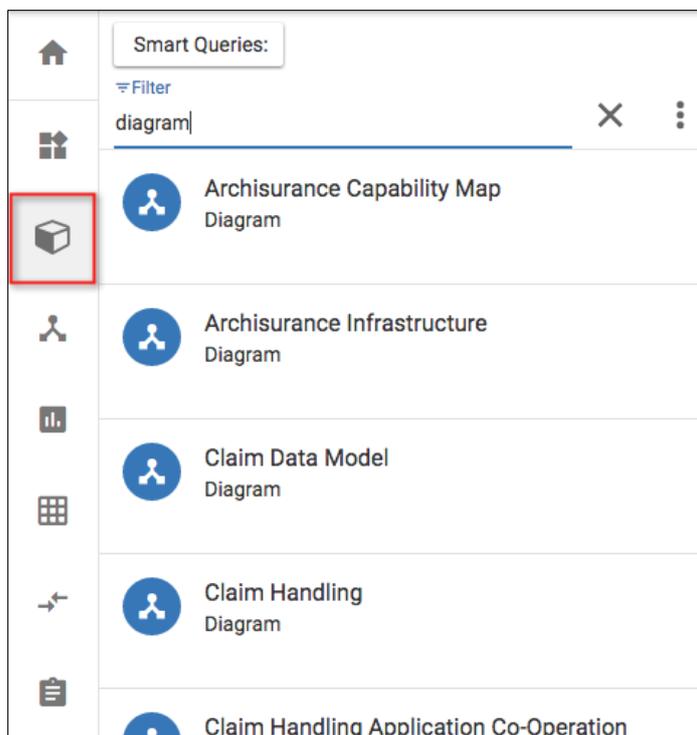
# Diagram/View/Publication Object representations

Diagrams, Views and Publications are now 1st class citizens and are represented as object types within the metamodel.

Any time a diagram, view or publication is created within its own module, an object instance will be created within the all objects list.

This now means that users:

- Can model these types within diagrams
- Can use these types within relationship lists
- Can use these types within views

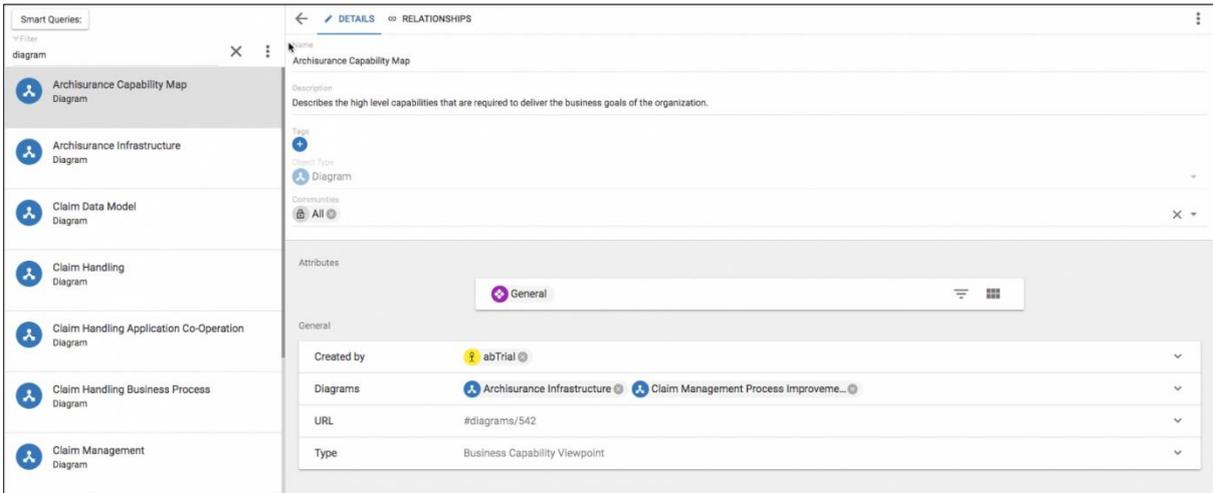


There are some attributes stored for the diagrams, including who created it, associated diagrams, its URL extension and what type it is

There are some attributes stored for the views, including who created it, associated views and its URL extension

There are some attributes stored for the publications, including who created it, associated publications and its URL extension

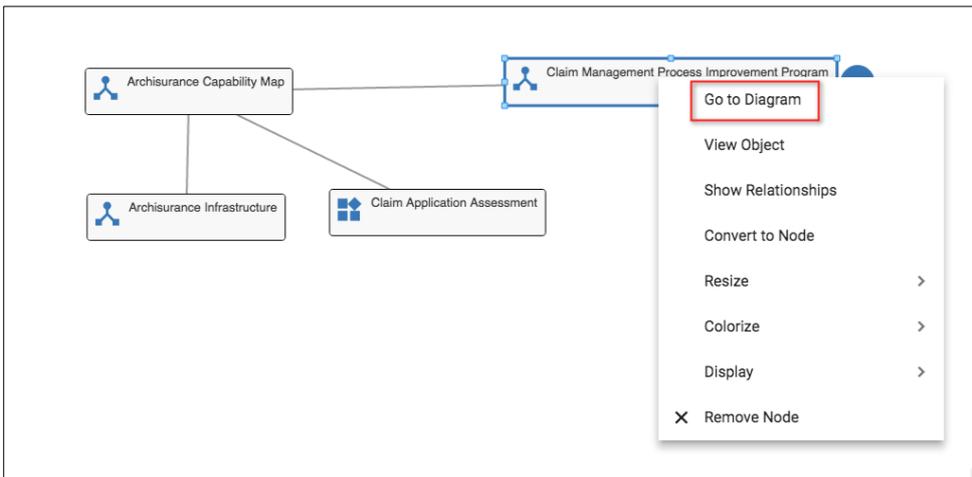
Currently these attributes are read-only for users, however relationships to these objects can be created.



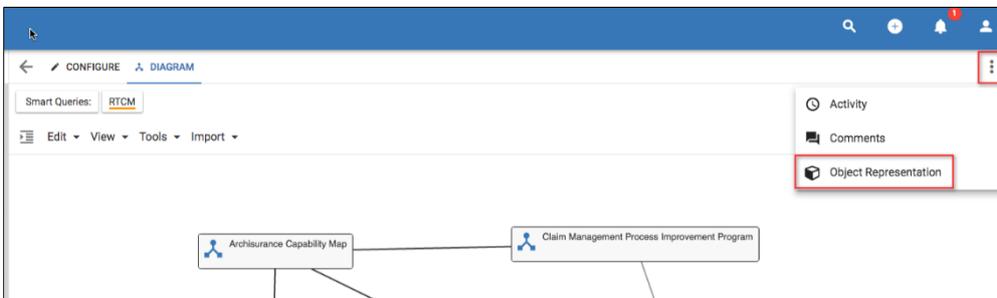
## Model these types within diagrams

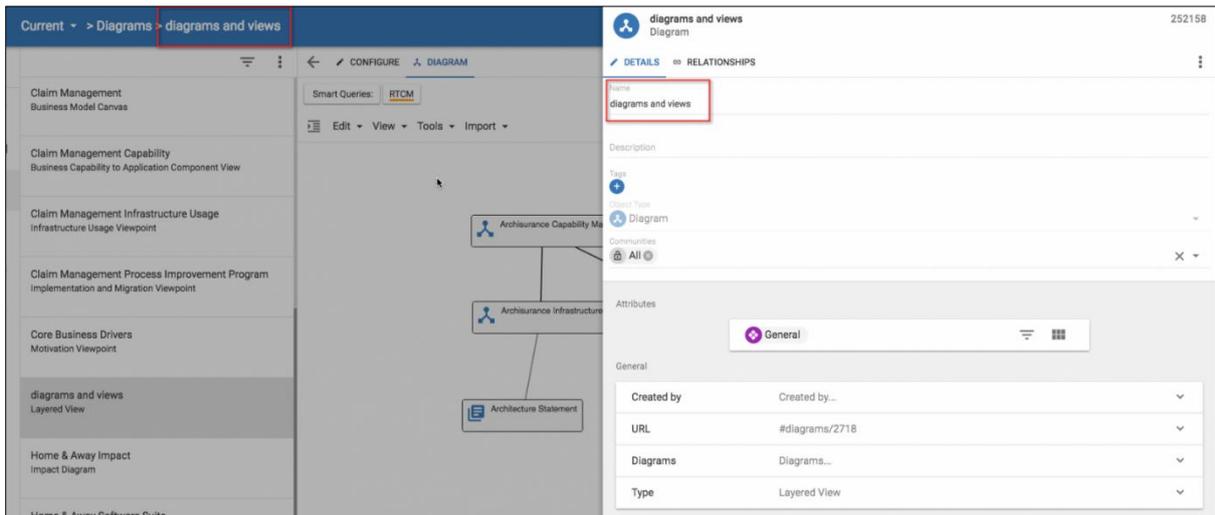
Any associated objects will be shown by relationships, as within standard diagramming.

Right clicking will allow you to navigate into the diagram/View/Publication itself from within the diagram



The representing object details and relationships can be accessed when in a view or diagram, using the 3 dot menu and selecting the Object Representation menu item

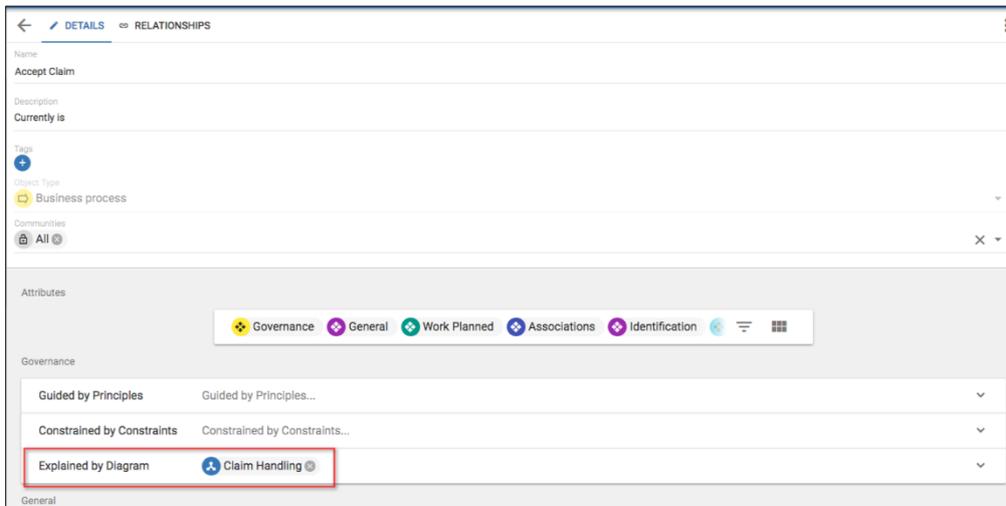




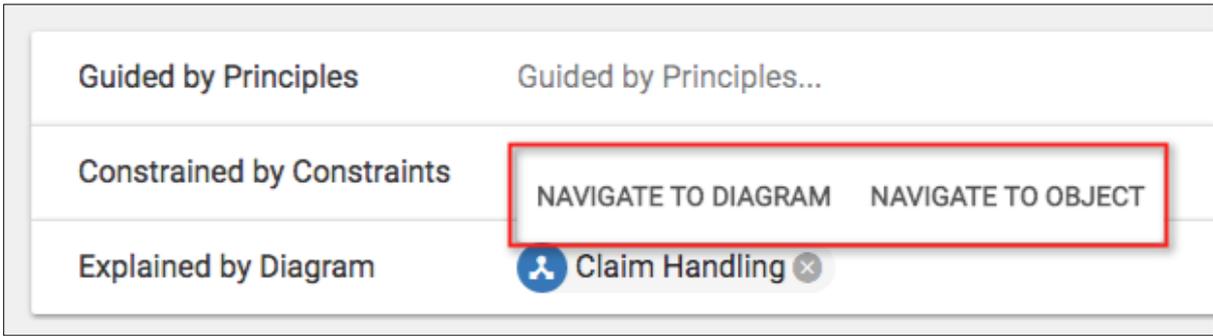
## Use these types within relationship lists

This allows us to explain objects with a diagram, a view, or a publication.

To do this, create a new attribute, of type relationship list, e.g. 'Explained by diagram' and use an 'associated with' relationship to the diagram or view object type.



Hovering over the relationship list values, will offer up the chance to navigate into the visualization or to open the representing objects details in a panel



**Use these types within views**

The diagram, view and publication types can now be included within searches in views

Diagrams, views and publications

Description

Tags

Communities

All

Icon

Color

Bookmark

Order

6

**Data**

Object Type

Diagram

(18 objects in dataset)

Invert

Criteria Type

Name

Operator

contains

Value

AND OR

Object Type

View

(9 objects in dataset)

Invert

Criteria Type

Name

Operator

contains

Value

AND OR

Object Type

Publication

(4 objects in dataset)

Invert

Criteria Type

Name

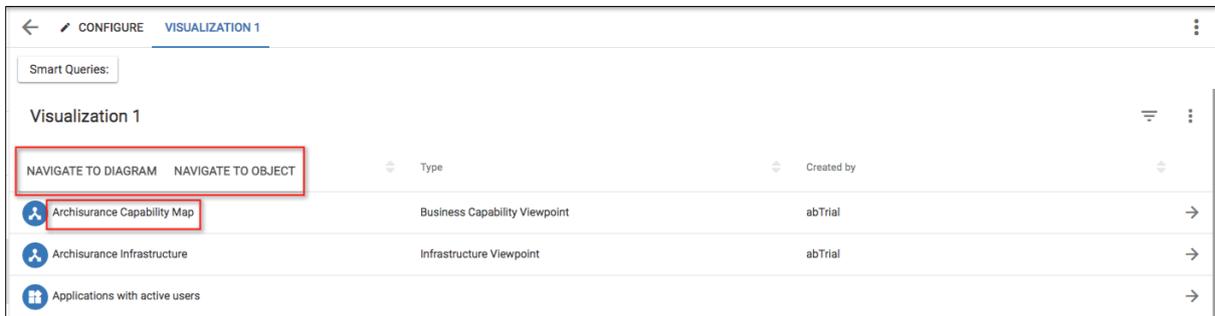
Operator

contains

Value

AND OR

Views including the objects can be created and used to navigate into the representing objects details, or the visualization itself - to do this you can hover over the name of the object within a view and choose.



---

# Campaigns

## Campaign

**Role Applicable to:** [Client Administrator](#), [User](#), [Contributor](#), [Reviewer](#)

A campaign is often referred to as a 'challenge'. You can use campaigns to ask a set of users for specific ideas to solve a particular business problem, or to encourage innovation around a specific topic.

Campaign permissions are set according to the 'goal' object (of any object type) that is set within the campaign. The campaign picks up the community permissions that are assigned to the 'goal' object (of any object type) chosen. Users who cannot see the 'goal' object will not see the campaign within their list.

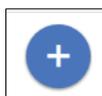
A focused campaign can yield a high level of quality responses. They are aligned to a specific goal in the organisation and are therefore likely to be implemented if they make business sense.

Campaigns are initiated by a sponsor (campaign administrator), who wants to invite a number of stakeholders to propose and collaborate around objects.

- To create a campaign, navigate to the campaign icon from the views menu on the left of the screen.



- Then use the + button to add a new campaign.



- You will then be in the CONFIGURE tab and see an empty campaign form.



- Name: The name of the campaign.
- Start Date: The date the campaign begins.
- End Date: The date the campaign ends, optionally click **(Make Open Ended?)** above to remove the end date field.
- Admin: The user that will be the administrator for your campaign
- Sponsor: The user that will be the sponsor for your campaign.
- Tags: Any tags that are applicable to the campaign.

- Goal Type: The object type for the goal.
- Goal: The object that will be your goal for the campaign (This does not have to be of object type Goal it can be any object you wish).
- Allowed Object Types: Which object types you would like people to be able to contribute to the campaign, (e.g. Ideas).
- Purpose: A small message describing the purpose of the campaign.
- Logo: This will be shown to everyone that contributes to the campaign, as well as appearing on any dashboard tiles that use the campaign.
- Description: This description field is in HTML so you can apply lots of styling. This will be the bulk of the information that contributors will read about the campaign.

! A campaign may not just be for ideas but can be for any type of information.

i The campaign description provides an opportunity to lay out the aims of the campaign in HTML, so that it can be designed to look aesthetically pleasing.

Once saved, you will be able to delete the campaign from within this configure tab.

## Viewing a campaign

Once the campaign is saved, the **CAMPAIGN** tab will show you the output of your campaign.



The campaign tab contains

- Information on the campaign for users

- 
- Areas where users can add new objects (limited to the allowed object types selected)
  - A list of objects that have been related with the campaign already, so that reviewers and viewers can view existing ideas, requirements etc. - they can also rate them in this table
  - Information about the campaign - Admin, sponsors, timings etc.
  - An image selected by the admin, and the goal that the campaign is intended to satisfy

Users will need to log-in to the tool with a valid account to be able to view the campaign.

After objects have been vetted and collaborated upon; an evaluation/review team created by the administrator reviews, evaluates, and selects the top objects. If these are ideas, they can be further elaborated as requirements. The evaluation/review team is often set up as a private community with access to the campaign.

The countdown deadline is a key aspect of a campaign as it drives participation and also aligns with any corporate timing goals. Some campaigns will be endless and will satisfy strategic objectives such as 'optimise business processes'.

### ***Example ideas management campaign:***

#### **Example -1:**

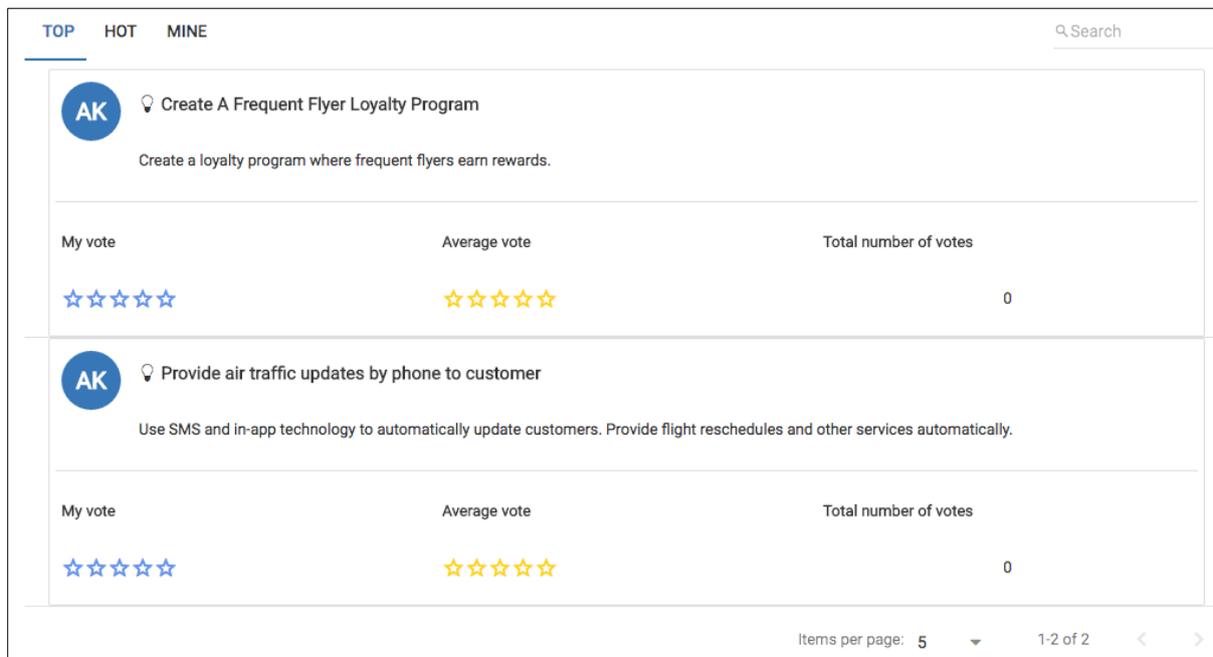
CM Enterprises is a global IT services company with many satellite offices. A directive from the board of directors to senior management is to grow Q4 sales in 2017 by 15%. The company also wants to make better use of social media. The company's culture is to ensure that employees feel empowered to contribute to the company's success.

The company will run a campaign that lasts for 1 month in Q1 2017.

# Gamification

**Role Applicable to:** Client Administrator, User, Contributor, Reviewer

Gamification is available for a campaign.



Users may rate objects with a star rating. Their individual stars are colored blue. The overall star (gold) indicates the rating of the object within the context of all other ratings. Ratings are specific to the campaign so if an object is used across multiple campaigns their ratings will be independent for each.

The leader board tabs can show different filters to further refine the list of objects.

<b>Hot</b>	Sorts concepts by those with the most comments associated with them.
<b>Top</b>	Sorts by overall rating
<b>Mine</b>	Shows only concepts that you have entered.

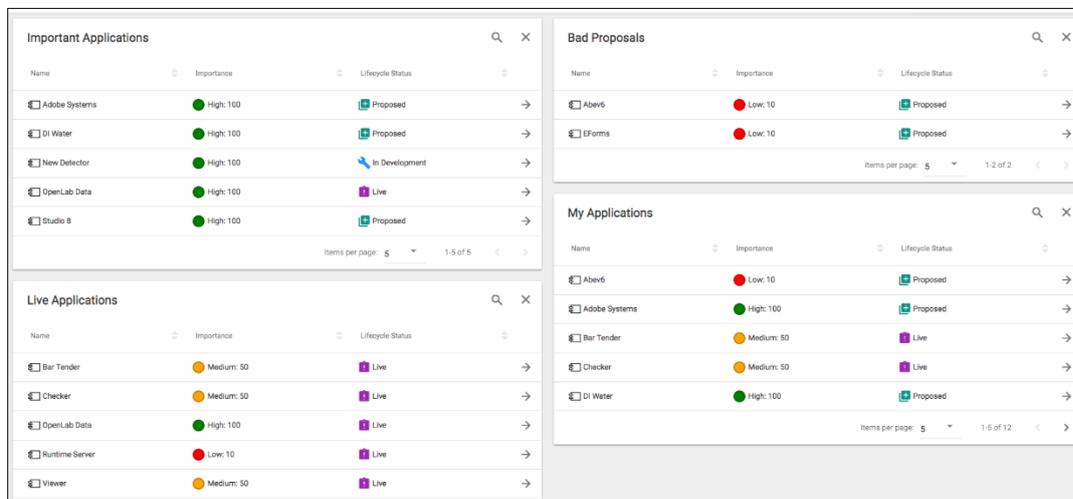
# Dashboard

**Role Applicable to:** Client Administrator, User, Contributor, Reviewer

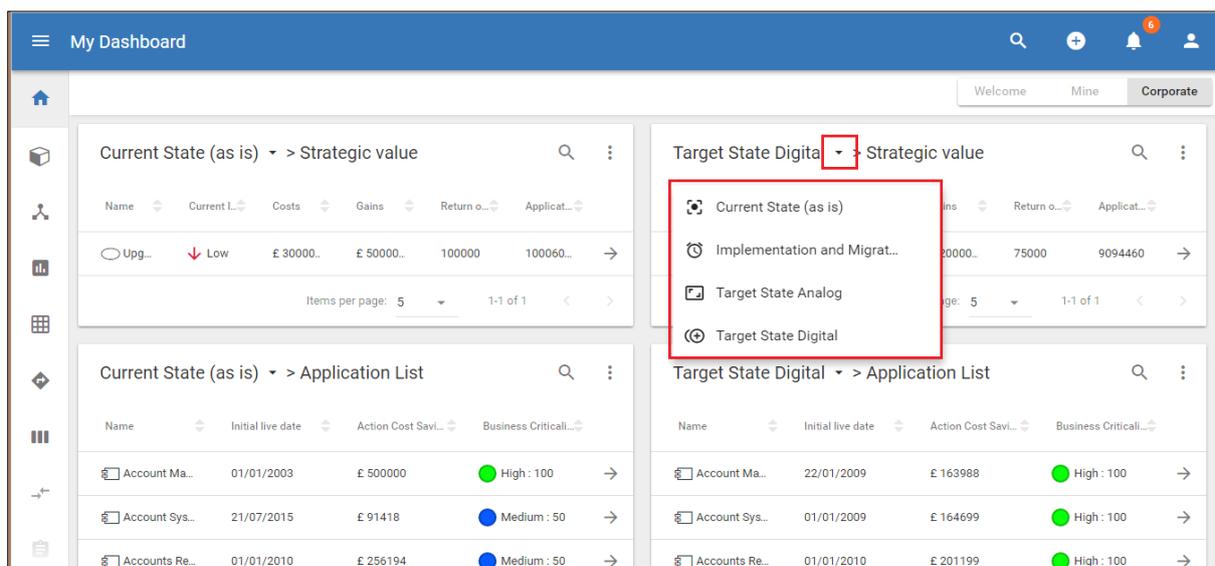
## Personal Dashboard

The personal dashboard provides the user with the ability to add 'views' that the administrator has created for them. They can be added using the '+ Add Visualization' option on the dashboard.

If the user has the correct permissions and can see the view, they can select it from the dashboard and be taken into the view. In the view they can use the option menu, to choose which columns are displayed within the visualization.



Views can switch between workspaces on the dashboard. This makes it very easy for users to see the different datasets contained in different workspaces, run against the queries set in the views, and compare them in a simple way.



---

## **Corporate Dashboard**

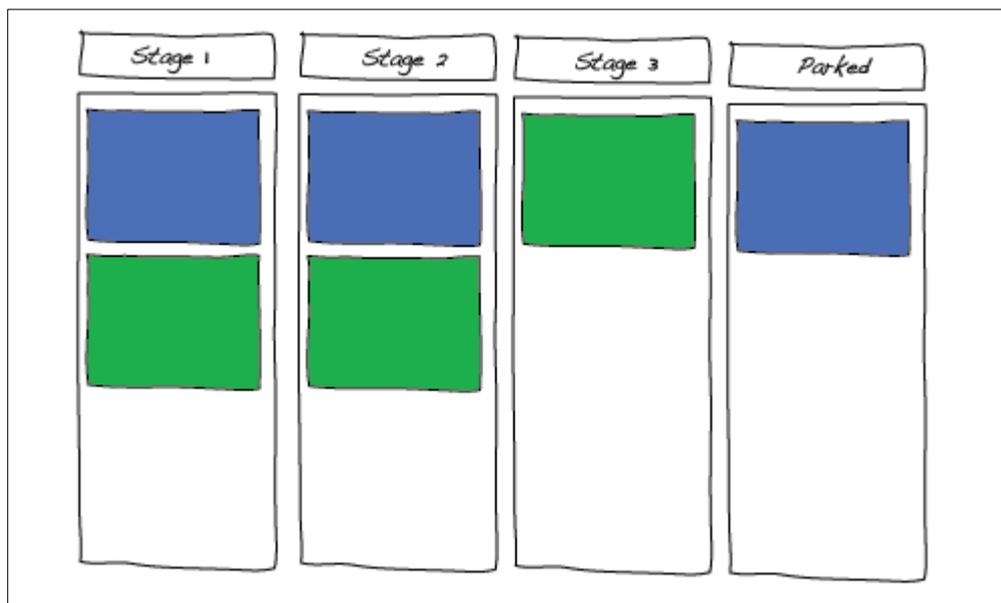
The corporate dashboard is provided by your administrator to show company wide views and visualizations on the platform. The administrator controls these views and they are not editable by the users.

---

# Kanban Boards

**Role Applicable to:** Client Administrator, User, Contributor, Reviewer

Kanban boards are the visual storyboard for a process or workflow. They represent the journey and the objects within that journey. Objects are represented as cards on the board. Objects may be moved from one stage to another by dragging.



**i** Kanban boards are based on the concept of Kanban. A Kanban is a tool to visualize, organize and complete work. The first official use of Kanban can be traced back to Taiichi Ohno's work at Toyota. He needed a way to quickly communicate to all workers how much work was being done, what state it was in and how the work was progressing. His goal was to make information and processes transparent to everybody and not just the management team.

## Creating a Kanban Board

Kanban Boards and Roadmaps are the first of the existing views to be migrated into the **View Manager**. We can now use them with queries, to create dynamic datasets with multiple visualizations, rather than having to define a static dataset for each module.

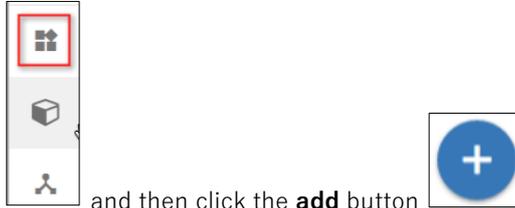
Kanban boards are used to track work in progress. They are key to working in an agile manner and can be used to drive teams to deliver work.

EA objects can be added to kanban boards to show where they are in a work stream.

Objects are represented as cards on the board. Objects may be moved from one stage to another by dragging a card.

---

First we will define our dataset;



- Select the **Views** module and then click the **add** button to create an empty View.
- Fill in the view details on the **CONFIGURE** page - some example details are shown below
  - **Name:** Applications with active users
  - **Description:** Application components with an Active Users attribute set to 1 or more.
  - **Icon:** Select an appropriate icon, and color, to identify the view.
  - **Bookmark:** Check the checkbox, so that the view will appear in the left-hand menu.
  - **Order:** Select 1 from the drop down, so that the bookmarked view will appear first in the menu.

The **Data** section contains the dynamic query that selects object instances to be included in the view. The selection of an Object Type will include all objects of that type in the Dataset. Any selection criteria will then filter the dataset.

An example can be seen below

- **Object Type:** Select **Application component**
- **Criteria Type:** Select **Attribute**
- **Criteria options:**
  - **Attribute:** Select the **Active Users (Integer) attribute**
  - **Operator:** Select **Greater Than >**
  - **Value:** set the value to **0**

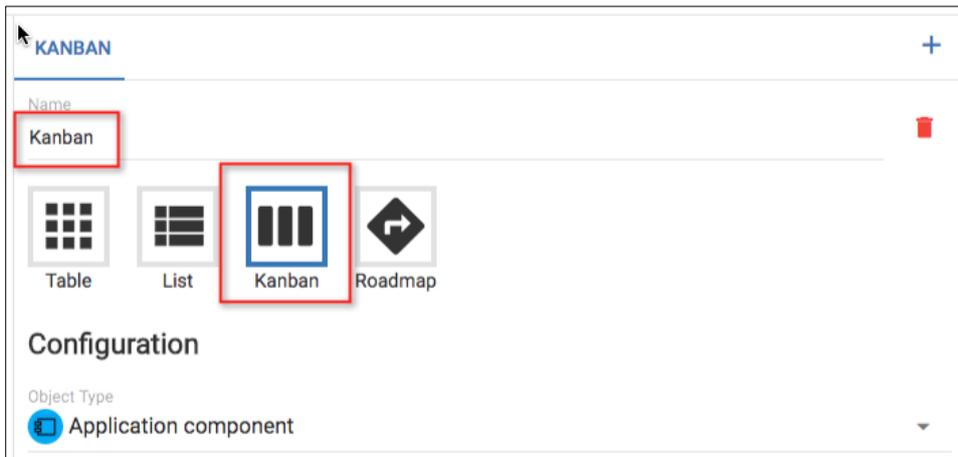
The query can be made more complex by combining further selection criteria. Clicking **AND** will make the query more restrictive, while **OR** will add to the possibility of being included in the dataset.

**i** Note that multiple Object Types, with their own queries, could be part of the view by clicking the + **Add Object Type** button.

- **SAVE** the View.

A view of, Application components that have active users, now exists. Visualizations can be created for views. To create a kanban visualization from our existing view;

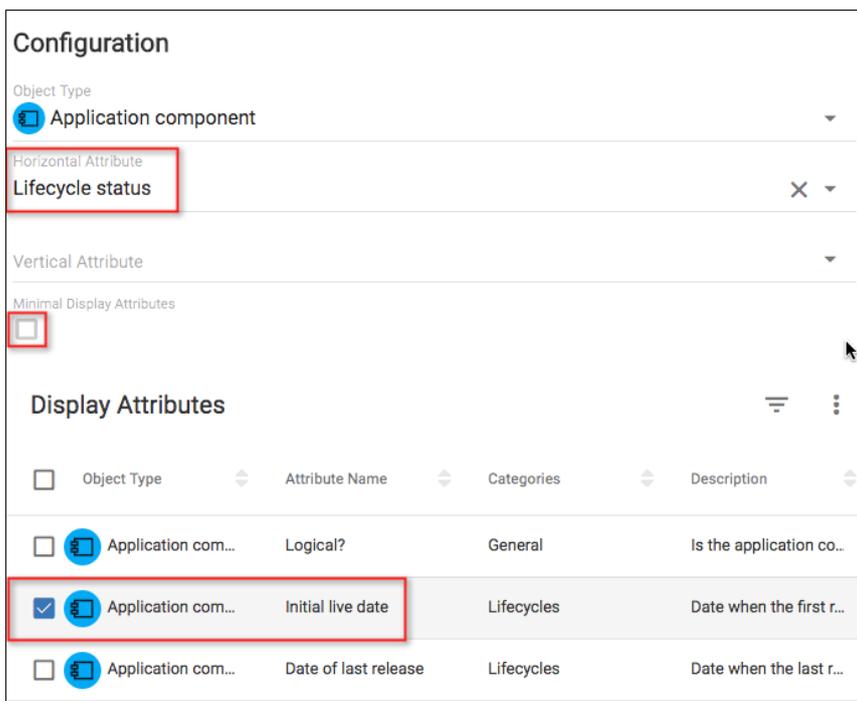
- Select the **Applications with active users** view and click the **CONFIGURE** tab.
- Click the + symbol in the top right corner
- Replace "Visualization 2" with an appropriate name for your kanban
- Select the visualization type **Kanban**



In the **Configuration** section of the visualization, the object type to be used in the kanban can be selected, along with the attribute that will define the stages of the kanban (Horizontal attribute) and the attribute that you may wish to group by (Vertical attribute).

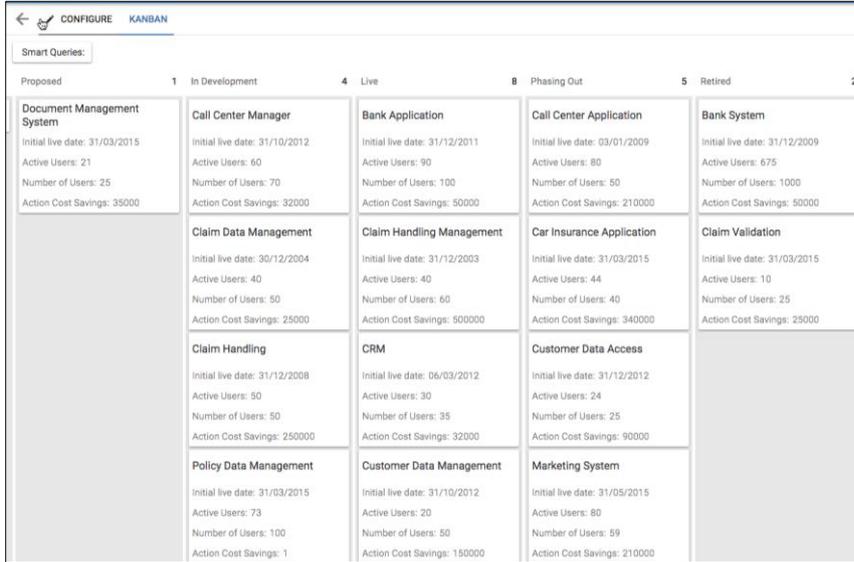
**i** | Kanbans are currently limited to a single object type.

- Set the Object Type
- Select a Horizontal Attribute to define the stages of the kanban
- Uncheck the Minimal Display Attributes checkbox (this will show just an icon if one is set for the attribute)
- Check some of the attributes to display on the kanban 'cards' from the object type



- **SAVE** the new visualization.

- Click the tab with the created visualization name on it to view the kanban.



**Accept Claim**

Process ● High: 100

criticality:

Standards class: ● Standard

Attribute icons and colours can also be set to show on the cards

The kanban board displays Application components, in stages that represents the value of their **Lifecycle status** attribute.

Note that the requested Display Attributes are shown on each kanban card.

- Open on of the object's edit window, by clicking on its name.
- Locate the chosen attribute to see that its value matches its stage name.

**Call Center Application (292898)**  
Application component

[DETAILS](#) [RELATIONSHIPS](#)

---

Name  
Call Center Application

---

Description

---

Tags  
+

---

Object Type  
Application component

---

Communities  
All

---

Attributes

Filter status

Unassigned

Lifecycle status	<span style="color: green; font-weight: bold;">●</span> Live	▼
Health Status	<span style="color: green; font-weight: bold;">●</span> Excellent: 100	▼

- Click away from the edit window to close it.



If the object's card is dragged to a different stage of the Kanban board, the Lifecycle status will be changed. If the Lifecycle status is changed while editing the object, the object's card will be displayed in the appropriate stage of the Kanban board.

## Kanban Grouping

Kanbans can now also be 'grouped', allowing users to select an attribute that cards on the kanban board can be grouped under

Within the visualization, you will see a field called **Vertical Attribute** - this will present a list of all of the attributes that exist under the chosen object type.

Selecting a value for this attribute will group the cards under matching values...e.g. we may have a relationship list attribute that is called 'Location' and this is set to assign an application object to a location, the cards will be grouped into swimlanes where the Location attribute values match.

Name  
Kanban

Table List Kanban Roadmap

Configuration

Object Type  
Application component

Horizontal Attribute  
Lifecycle status

Vertical Attribute  
Locations

Locations / Lifecycle...	Unassigned	1 Proposed	1 In Development	4 Live	8 Phasing Out	5
Front Office	3			<b>Call Center Manager</b> Initial live date: 31/10/2012 Active Users: 60 Number of Users: 70 Action Cost Savings: 32000	<b>Risk Assessment</b> Initial live date: 31/03/2013 Active Users: 50 Number of Users: 60 Action Cost Savings: 45000	<b>Web Portal</b> Initial live date: 31/12/2009 Active Users: 440 Number of Users: 250 Action Cost Savings: 208500
ArchInsurance Of...	1				<b>CRM</b> Initial live date: 06/03/2012 Active Users: 30 Number of Users: 35 Action Cost Savings: 32000	
Back Office	2				<b>Customer Data Management</b> Initial live date: 31/10/2012 Active Users: 20 Number of Users: 50 Action Cost Savings: 150000	<b>Home &amp; Away Financials</b> Initial live date: 31/05/2015 Active Users: 60 Number of Users: 70 Action Cost Savings: 200000

---

# Analysis of Objects

## Comparisons

**Package:** Innovation Management

**Role Applicable to:** [Client Administrator](#), [User](#), [Contributor](#)

The compare capability uses a technique known as the Analytic Hierarchy Process (AHP). AHP focuses on group decision making, and is useful where communities can come together to get consensus on a defined goal.

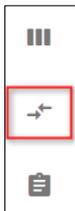
AHP provides a framework for structuring a problem and for relating objects to achieving a goal. The AHP decomposes a problem into a hierarchy, so that it may be easily understood.

The hierarchy then provides a set of individual objects which can be systematically evaluated against each other in turn. This is known as 'pairwise comparison'. When each of the objects has been evaluated against each other, they are ranked in order of importance of that decision.

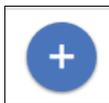
The same object can have a different ranking in a different hierarchy. Each hierarchy is represented by a goal that the decision is satisfying, criteria by which alternatives are grouped and alternatives themselves.

### Creating a comparison

Firstly, navigate to the comparisons module.



Then press the add button in the bottom left of the list.



Next you will need to fill in the fields on the form:

- **Name:** The name of the comparison
- **Description:** Describe the comparison, maybe add what you aim to achieve?
- **Goal:** The goal object, this can be of any object type but it should be something you wish to achieve.
- **Criteria:** At least one criteria to compare the alternatives against is required although you can have multiple.

- **Alternatives:** These are the objects that you wish to compare against each other.

Press **Save** in the bottom right when you've completed the form.

Next to the save button, a new button will appear called **Start**.

Once you press start the comparison will no longer be able to be modified through the form, but you will be able to begin comparing the objects. Pressing start will give you a warning message to confirm that you won't be able to modify the comparison further. Press OK and the comparison will be created.

## Using a comparison

In order to prioritize which objects are the most important, objects can be ranked against each other.

Select the **Rank** tab, choose the **criteria** and choose the **Show/Hide Results** option. Perform the comparison of the objects presented.

Ranking is a group activity and all members of a comparison community may rank objects.

The comparison is used to provide a structured framework for quantifying the importance of objects.

The platform helps you systematically evaluate different objects (known as alternatives) by comparing them to one another two at a time, with respect to the criteria that they belong to.

The comparison takes the paired evaluations and compares these over the entire set of pairs. A priority (or rank) is given to each object, with the most important objects have the highest rank.

The results of comparisons can be visualized in charts, or as a table.



The client administrator will have set up comparison sets which represent the goal, criteria, and alternatives.



It is expected that ranking of objects takes place at the end of a campaign once all of the objects have been collected. Comparisons cannot have new objects added to them once they have been set up.

When carrying out a 'pairwise' comparison, use the scale at the bottom of the page to select the object weighting that you prefer.

Slide the slider toward the object that has the highest weighting between the two objects.

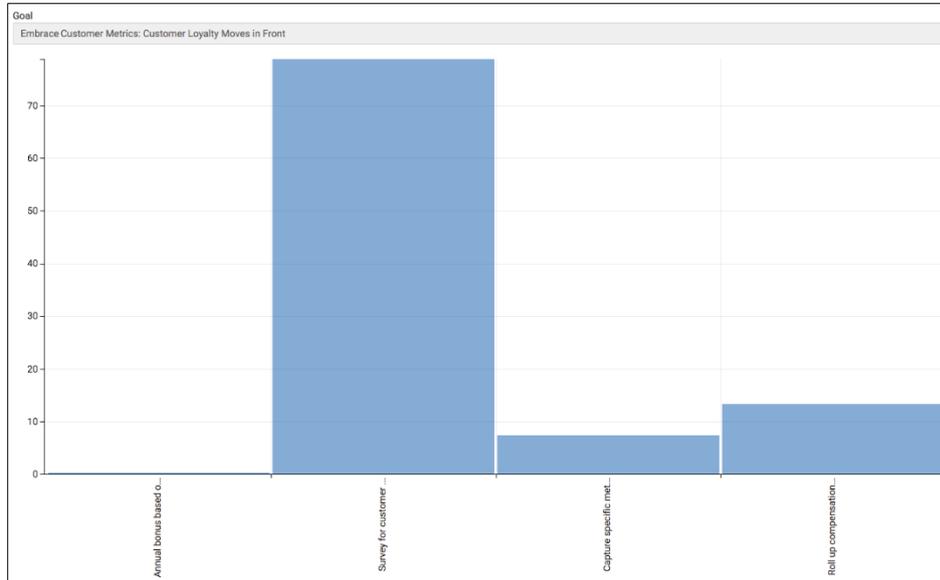
Press **Save** to reveal the next pair to compare.

---

Once the pairwise comparison is complete, you will receive a completion message and a ranking table will be displayed.

To see the overall rank of the concepts in the comparison, press the comparison tab at the top of the page.

From there you will be able to see a chart of your results.



# Charting - Using and Creating charts

**Role Applicable to:** Client Administrator, User, Contributor, Reviewer

A full charting capability provides the option for representing objects and their attributes.

The chart user interface allows the user to select either a comparison or a object type for the focus.

When choosing a comparison, then the object type for the set of alternatives is used and the attributes for that object type are selectable.

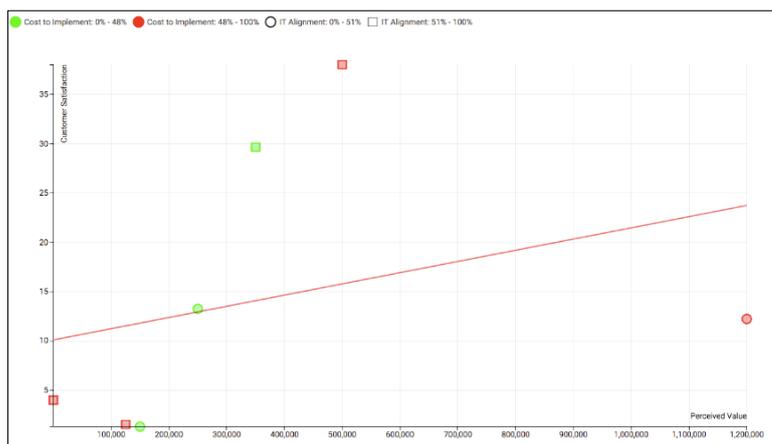
When choosing an object type, then the attributes that are applicable to the object type are displayed.

There are currently three chart types: **Stacked Bar, Scatter and Word Cloud.**

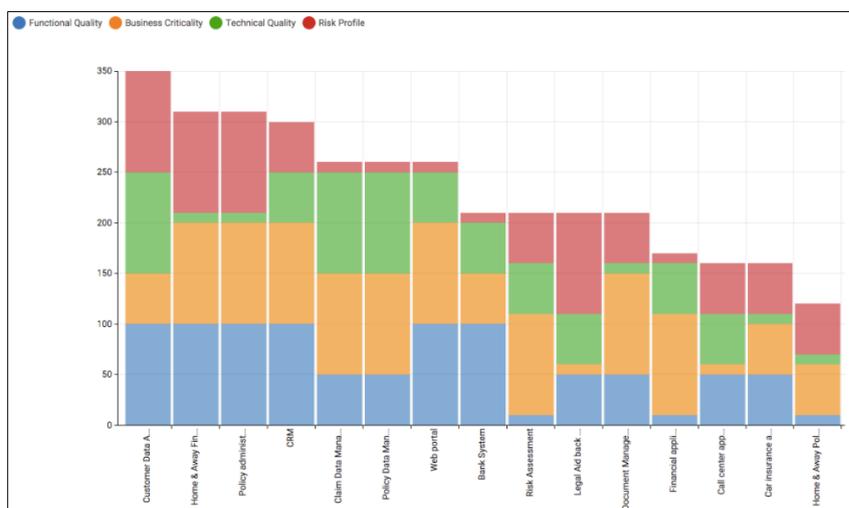
Both stacked bar and scatter have X and Y values that can be mapped to concept attributes.

A scatter chart has a further axes (colour and shape) which represents the concepts on the scatter chart.

- Numerical ranges can be defined for the colour and shape axes



A stacked bar chart can have multiple y values to make up the stack.



---

**i** | Charts usually require numerical values. Properties such as numerals or arrays (value pairs) should be used.

**!** | It is also important to note that on a stacked bar, all y axis properties should be in the same numerical range, otherwise some stacked properties will not be visible.

*For example: An attribute that represents perceived value in a scale up to millions, should not be stacked on top of an attribute that represents Risk on a scale of 1-100.*

Word clouds can identify trends and patterns that would otherwise be unclear or difficult to see in a tabular format.

Frequently used keywords stand out better in a word cloud. Common words that might be overlooked in tabular form are highlighted in larger text making them pop out when displayed in a word cloud.

They can be an effective method for analyzing text-data. The word cloud chart analyzes the description contents of an object type and provides a summary of the keywords that are used.

This is useful for example when you want to explore campaigns and trending words and phrases.

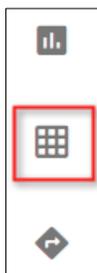
---

# Pivot tables

**Role Applicable to:** Client Administrator, User, Contributor, Reviewer

The pivot table provides the ability to slice and dice your objects by their attributes.

The pivot table is located in the view menu.



If Pivot tables exist already, you can see them in the list and each one can be clicked on to access it.

You can add a new pivot table by clicking on the Add button in the bottom right of the list.



Enter a name for the pivot table in the name field. Enter a description for the pivot table in the description field. Please be as descriptive as possible, as the description is shown when you next look through the list of existing pivot tables.

Each pivot table is for a specific object type in the tool. Choose an Object Type, for example, an Idea.

## Adding Objects to the Dataset

To populate the Dataset it will be necessary to provide some selection criteria using the Search feature (magnifying glass icon).

Individual objects, returned by the search criteria, can then be de-selected from the list before **Adding** to the Dataset.

## Removing Objects from the Dataset

Objects can be selected, using the checkboxes on the left, before using the Remove feature (trashcan icon).

Multiple checkboxes can also be checked using the Filter feature.

## Displaying Objects in the Dataset

Clicking on the **Pivot Table** tab will display a table of the objects in the Dataset.



Application components	In Development	Live	Mainstream	Phasing Out	Totals
Bank System			Low : 100		Low : 100
Call center application		Medium : 50			Medium : 50
Car insurance application				Low : 100	Low : 100
Claim Data Management		Medium : 50			Medium : 50
CRM				Medium : 50	Medium : 50
Customer Data Access				Low : 100	Low : 100
Document Management System	Low : 100				Low : 100
Financial application		High : 10			High : 10
Home & Away Financial Application		High : 10			High : 10
Home & Away Policy Administration		Low : 100			Low : 100
Legal Aid back office system		Low : 100			Low : 100
Policy administration		Low : 100			Low : 100
Policy Data Management		Low : 100			Low : 100
Risk Assessment		High : 10			High : 10
Web portal				High : 10	High : 10
Totals	Low : 100	Low : 100, High : 10, Medium : 50	Low : 100	Medium : 50, Low : 100, High : 10	Medium : 50, Low : 100, High : 10

Feel free to drag and drop the attributes from the column or the row headers to create your pivot table.

Experiment with the table type and cell fields to create different table options:

Application components	In Development	Live	Mainstream	Phasing Out	Totals
Bank System			Low : 100		Low : 100
Call center application		Medium : 50			Medium : 50
Car insurance application				Low : 100	Low : 100
Claim Data Management		Medium : 50			Medium : 50
CRM				Medium : 50	Medium : 50
Customer Data Access				Low : 100	Low : 100
Document Management System	Low : 100				Low : 100
Financial application		High : 10			High : 10
Home & Away Financial Application		High : 10			High : 10
Home & Away Policy Administration		Low : 100			Low : 100
Legal Aid back office system		Low : 100			Low : 100
Policy administration		Low : 100			Low : 100
Policy Data Management		Low : 100			Low : 100
Risk Assessment		High : 10			High : 10
Web portal				High : 10	High : 10
Totals	Low : 100	Low : 100, High : 10, Medium : 50	Low : 100	Medium : 50, Low : 100, High : 10	Medium : 50, Low : 100, High : 10

Clicking on the drop down arrow next to an attribute will allow you to filter by the values of the attribute:

Failure Risk ▾

Lifecycle status ▾

Lifecycle status (4)

- In Development (2)
- Live (18)
- Mainstream (2)
- Phasing Out (8)

Application components				Phasing Out	Totals
Bank System					6.7%
Call center application					6.7%
Car insurance applicatio				6.7%	6.7%
Claim Data Managemen					6.7%
CRM				6.7%	6.7%
Customer Data Access				6.7%	6.7%
Document Management					6.7%
Financial application					6.7%
Home & Away Financial Application		6.7%			6.7%
Home & Away Policy Administration		6.7%			6.7%
Legal Aid back office system		6.7%			6.7%

Press the **Save** button at the bottom of the page to save the pivot table.

---

# Relationship Analysis

**Role Applicable to:** Client Administrator, User

The relationship analysis feature allows you to find direct and indirect paths between two objects.

This is useful when you want to see the impact of one object on another, or to understand which objects are affected by (related to) other objects.

You can also create derived relationships which are only visible in the diagram and not in the model using this capability.

The analysis feature is only available when in a diagram.

## Example: Application Components and affected Business Processes

Which Business Processes would be affected when **Bank System** is removed?

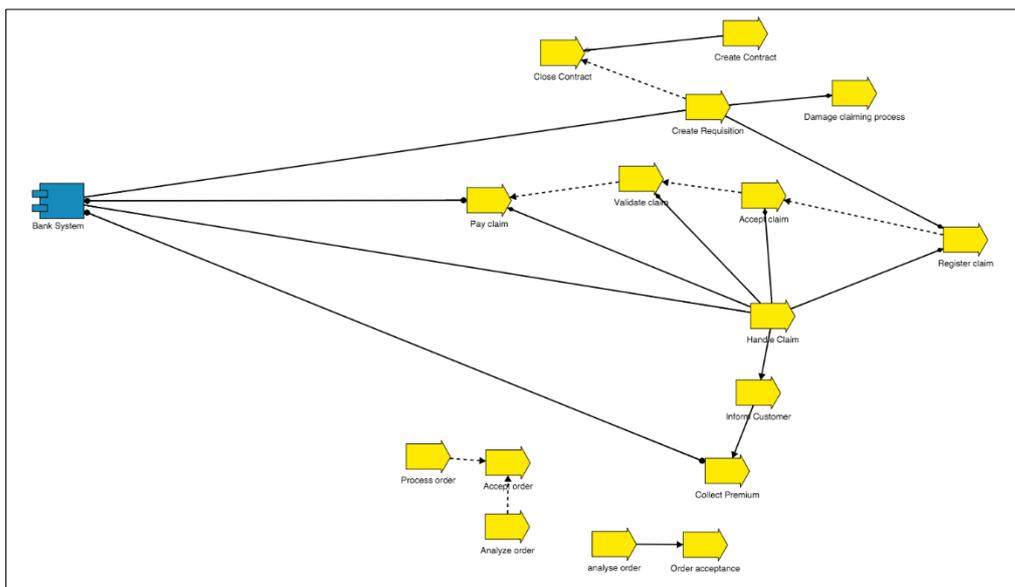
In this simple example, we use this feature to allow us to

- a) find indirect paths between objects visually (on the same view), or
- b) to populate from our model into our current view, using indirect paths that we may not have been aware of. When doing so, we are ensuring we can see all of the possible impacts of changing our application architecture, before we commit to do so.

### Steps

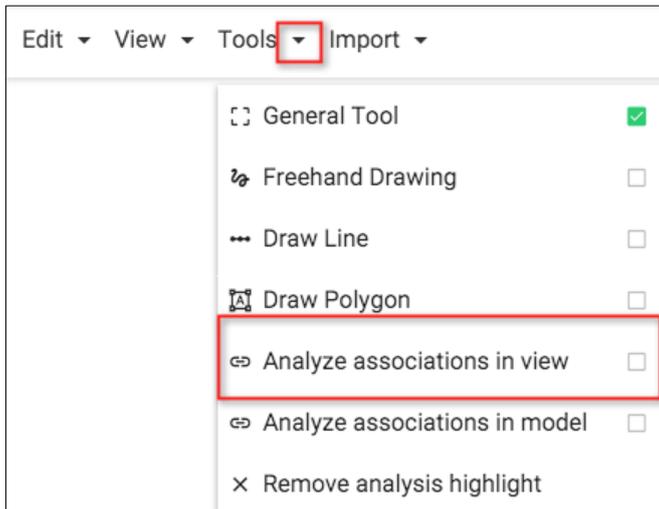
- Create a new diagram (e.g. Layered) and add to it some objects - e.g. an application component and business processes.

You can then investigate the relationships between them.



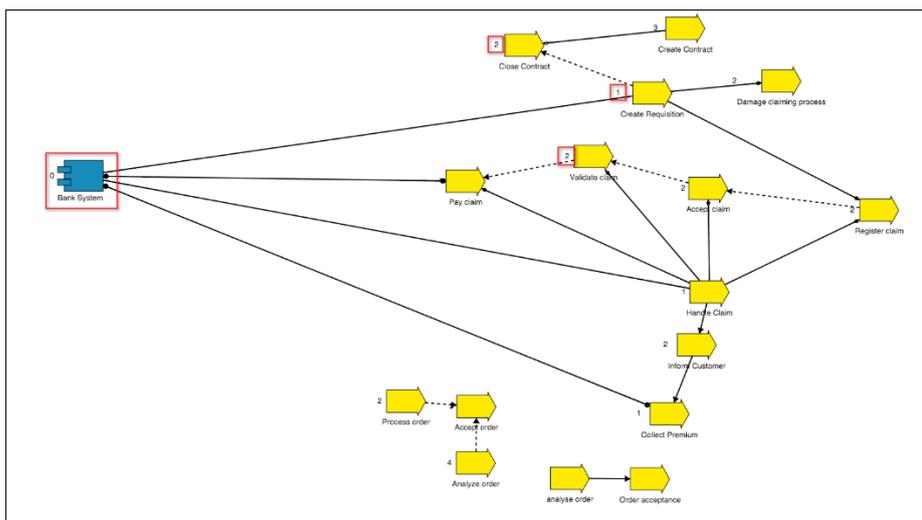
- Select the **Tools > Analyze relationships in view** option from the menu.

**i** Associations in view will only show relationships on the current diagram.

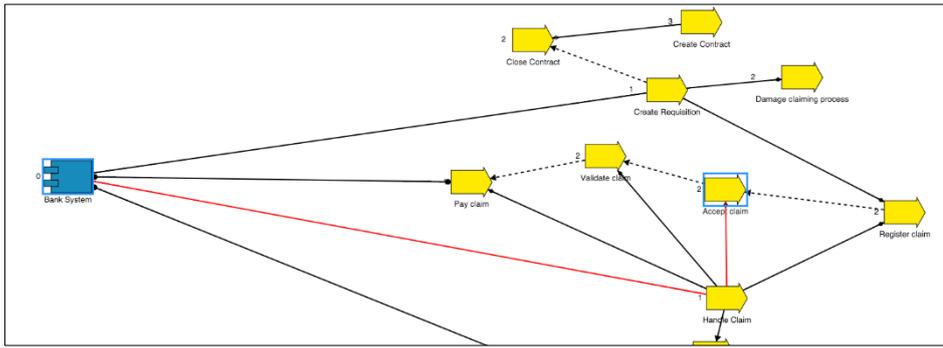


**!** Any right click context menu will not be usable until you exit the analyze mode - this can be achieved using the Tools > General Tool option.

- Once in the Analyze mode, clicking on any object will display the shortest route (in number of relationships) to all of the other objects within that current view.



- If you select a 2nd object, the shortest indirect path on the view will be highlighted

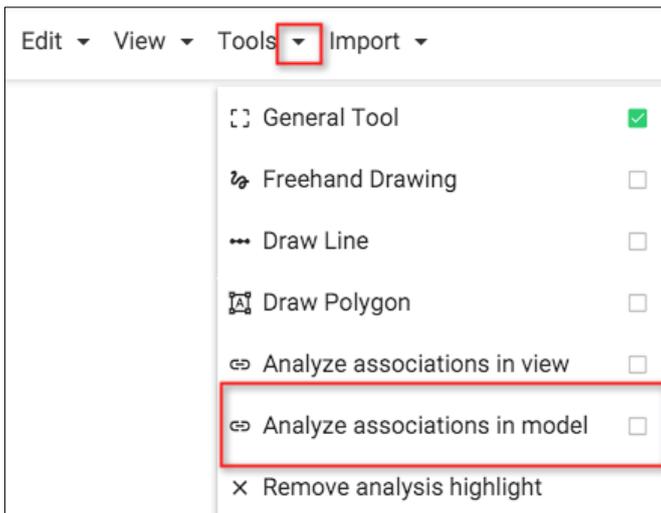


There may be some objects on this view that are either not related with the chosen object, or have a shorter route elsewhere in the model - to look at these you need to look at the indirect relationships at model level.

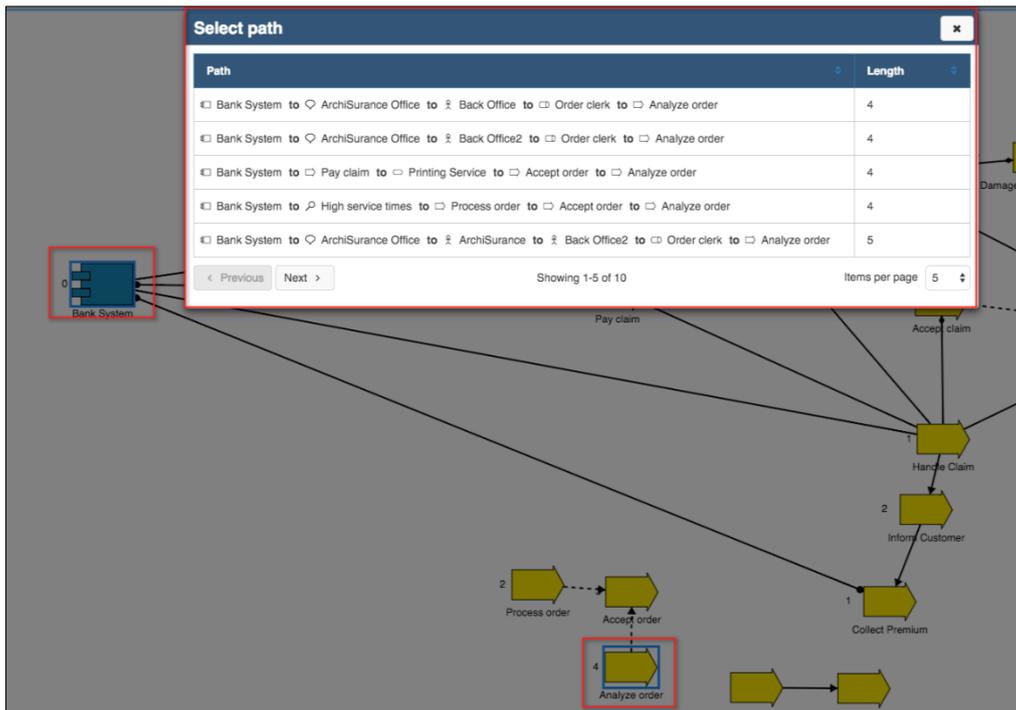
- If you swap, to select the **Tools > Analyze relationships in model** option, and re-select the same starting object - the objects' relationship count will be re-calculated and you will be presented with the shortest relationship count found within the entire model (this may include some relationships to objects not seen on this view).



Relationships in model examines all relationships regardless of the current diagram.

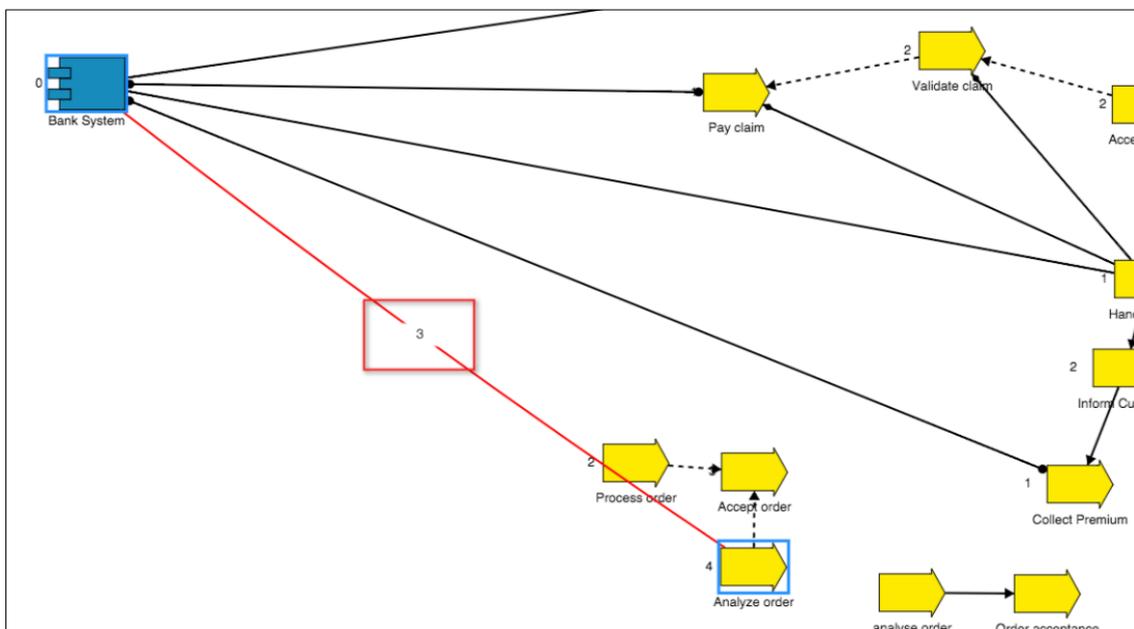


- Again, if you select a second object, the shortest paths will be shown that exist within the model - however this time, a table of the 10 shortest paths will be shown

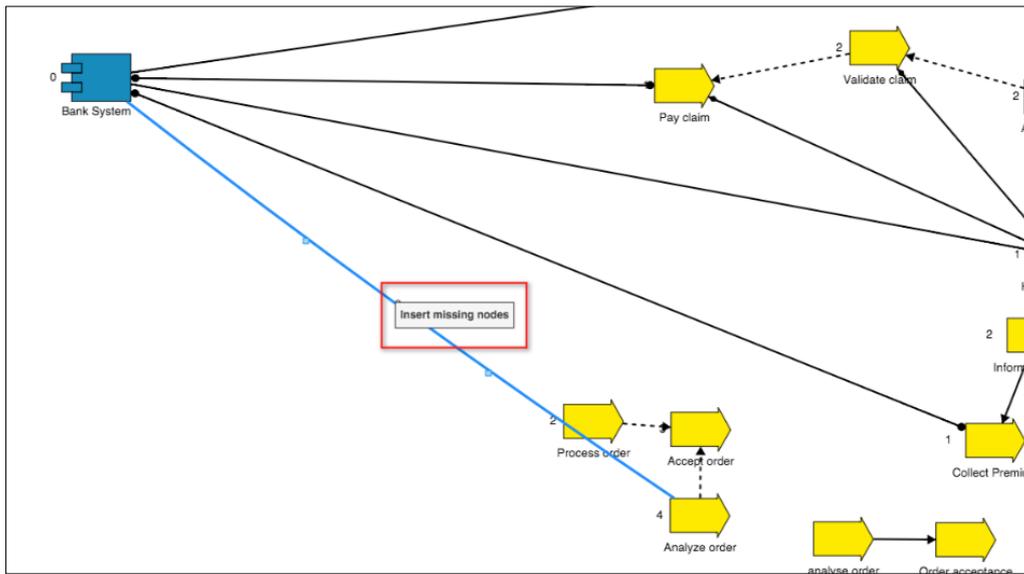


You can select one of these paths and it will be highlighted within the view if it exists, or highlighted with missing objects made evident on the current view

This can be seen by the number 3 within the red line - this means there are 3 relationships used in this path chosen that are not currently shown on the diagram.

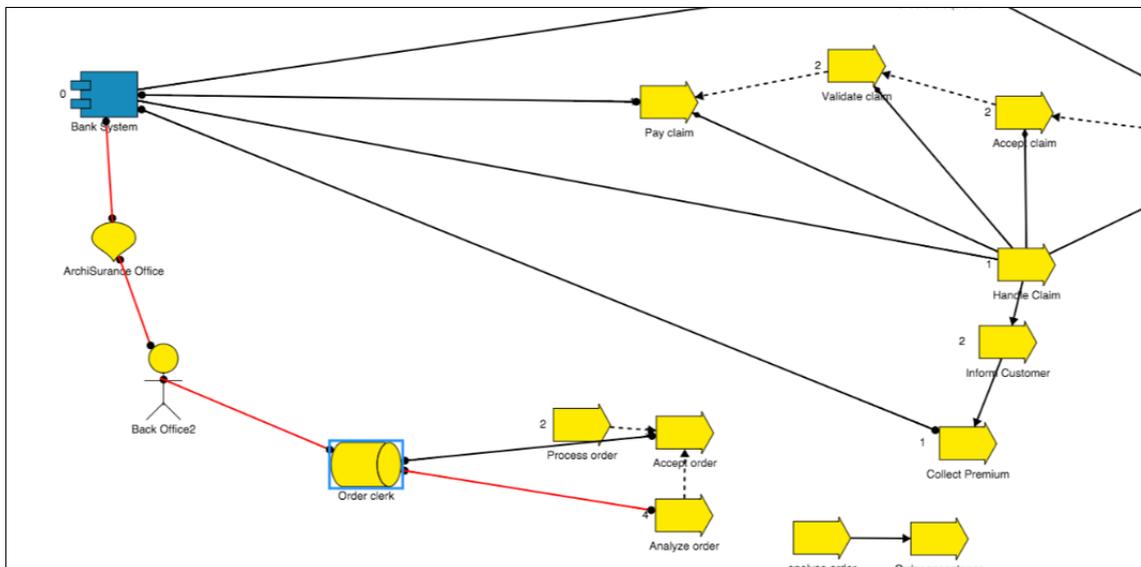


- If we choose to do so, we can right click on the number 3 and choose to **'Insert missing nodes'** - this will then add them to our current diagram.



**i** 'Missing' objects will only be added to the diagram if those object types are allowed to be on the diagram type.

- This allows us to see all of the impacted business processes to do with our application component Bank System, at diagram level and at model level.



**!** The red indirect relationship line can be saved on a diagram - but the next time the diagram is opened the indirect relationship that it was highlighting will not be maintained - it will just show an area of interest/to be analyzed again. This is because the model may well have changed since the diagram was last used, and the chosen indirect relationship path may no longer be valid.

---

You can also use a right click on the first object and then **Select by Objects** - from this point, you can select the options for objects from the drop down menu - this allows you to save time having to find the object on a large diagram if you know the name.

Also whilst in the **Analyze Relationships (view)** mode, using right click on a node will give you the option **Select by Type** - from here all nodes of that type on the diagram will have the shortest routes to them from the selected node highlighted.



Please note that this capability does not follow the direction of relationships but will in a future release.

---

# Roadmaps

## Road Mapping

**Role Applicable to:** [Client Administrator](#), [User](#), [Contributor](#), [Reviewer](#)

### Creating a Roadmap

Kanban Boards and Roadmaps are the first of the existing views to be migrated into the **View Manager**. We can now use them with queries, to create dynamic datasets with multiple visualizations, rather than having to define a static dataset for each module.

Roadmaps are based on date or date-range attributes and provide a time-based graphical representation of the object, in comparison to others within the dataset.

First we will define our dataset;



- Select the **Views** module , and then click the **add** button  to create an empty View.
- Fill in the view details on the **CONFIGURE** page - some example details are shown below
  - **Name:** Applications with active users
  - **Description:** Application components with an Active Users attribute set to 1 or more.
  - **Communities:** All (default)
  - **Icon:** Select an appropriate icon, and color, to identify the view.
  - **Bookmark:** Check the checkbox, so that the view will appear in the left hand menu.
  - **Order:** Select 1 from the drop down, so that the bookmarked view will appear first in the menu.

The **Data** section contains the dynamic query that selects object instances to be included in the view. The selection of an Object Type will include all objects of that type in the Dataset. Any selection criteria will then filter the dataset.

An example can be seen below

- **Object Type:** Select **Application component**
- **Criteria Type:** Select **Attribute**
- **Criteria options:**
  - **Attribute:** Select the **Active Users (Integer)** attribute

- **Operator:** Select **Greater Than >**
- **Value:** set the value to **0**

The query can be made more complex by combining further selection criteria. Clicking **AND** will make the query more restrictive, while **OR** will add to the possibility of being included in the dataset.

**i** Note that multiple Object Types, with their own queries, could be part of the view by clicking the + Add Object Type button.

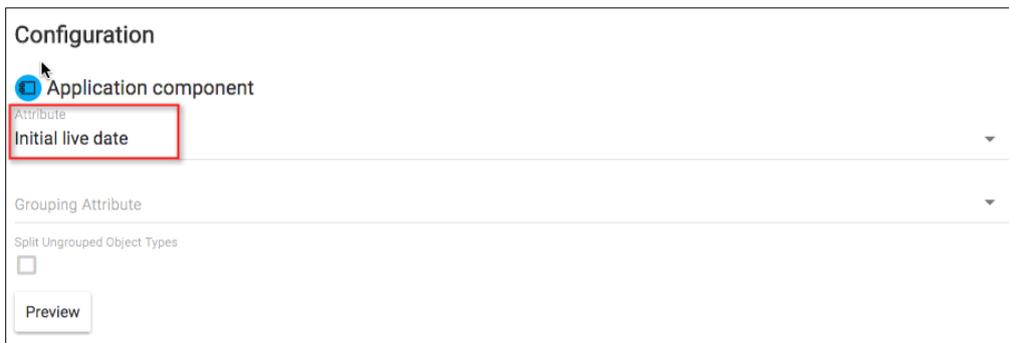
- **SAVE** the View.

A view of Application components that have active users, now exists. Visualizations can be created for views. To create a roadmap visualization from our existing view;

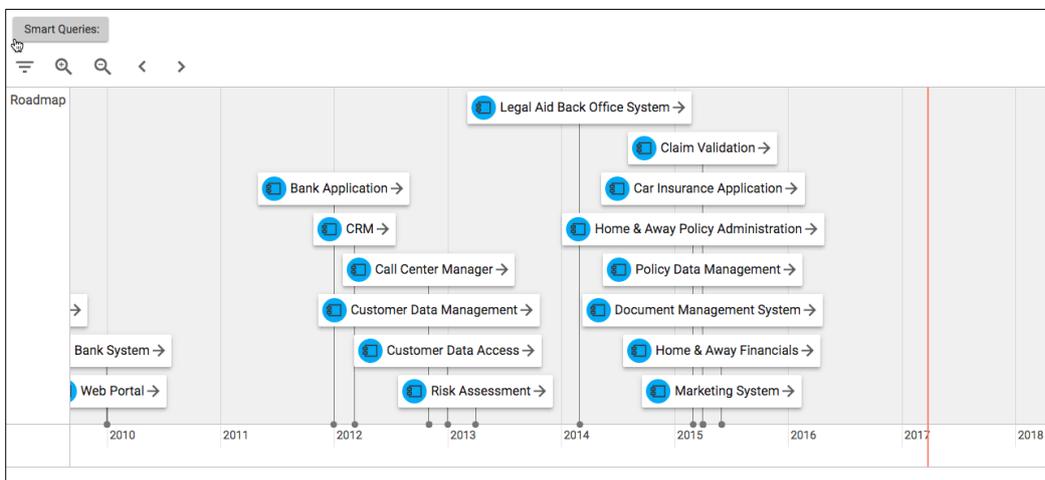
- Select the **Applications with active users** view and click the **CONFIGURE** tab.
- Click the **+** symbol in the top right corner
- Replace "Visualization 2" with an appropriate name for your roadmap
- Select the visualization type **Roadmap**



In the **Configuration** section of the visualization, the object types that can be used in the roadmap are presented separately, and different date attributes can be selected for each in the **Attribute** field. This is the field that will provide the timeline information to the roadmap



- **SAVE** the new visualization.
- Click the tab with the created visualization name on it to view the roadmap. These object are being shown by their 'Initial Live Date' attribute



**i** Click on an object to focus in on it - Objects may be moved on the timeline, or the date ranges be extended, by selecting and dragging...to update the attribute value being used.

Clicking the arrow on an object will open the object's details in a pane.

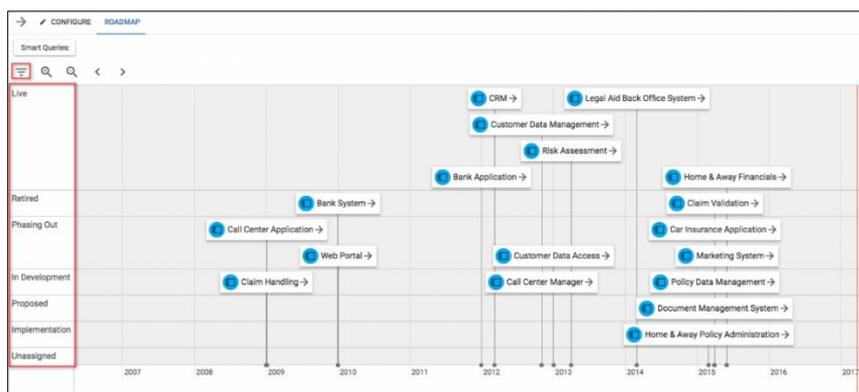
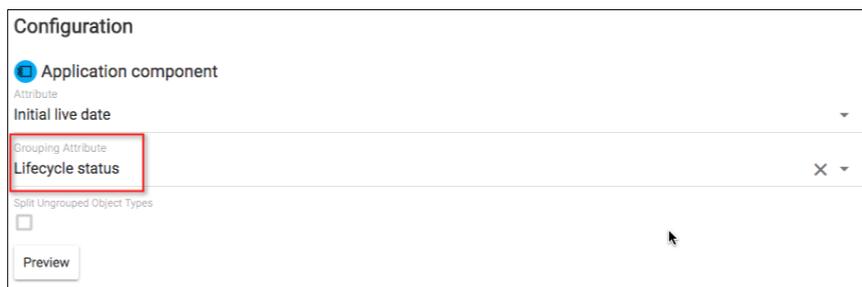
The + and - magnifying glasses can be used to zoom the timeline in and out.

## Roadmap Grouping

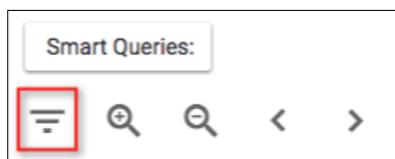
Roadmaps can now also be 'grouped', allowing users to select an attribute that objects on the roadmap can be grouped under

Within the visualization, you will see a field called **Grouping Attribute** - this will present a list of all of the attributes that exist under the chosen object type.

Selecting a value for this attribute will group the objects under matching values...e.g. we may have a 'lifecycle status' attribute...the roadmap objects will be grouped into swimlanes where the lifecycle status attribute values match.



Selecting the Filter option in the top left hand corner, will provide you with a new toolbar, allowing you to hide/show groups (clicking on it), or to select and drag groups into the position you want to see them in within the roadmap.



---

# Agile Enterprise Architecture and Roadmaps

**Package:** Agile Enterprise Architecture

Roadmaps provide a mechanism to show any concept over a time period. Most Enterprise Architect (EA) concepts will have a time dimension.

## The importance of roadmaps for Agile EA

The focus of Agile EA is on increasing the agility of the business. This means that it is essential that an organisation understands where it will be at any given period of time. The enterprise architecture concepts provide a blueprint of the organization. Many aspects of these objects can be described with a time dimension. The time dimension can be used to either display a milestone date at which something is expected to happen, or a date range within which something will take place.

As 'time to market' and the ability to change quickly is vital for organizations to keep pace and innovate, then roadmaps are a critical view on what, or how complex, the impact of change is.

## Roadmap views

Roadmaps are views on the enterprise architecture objects from a time dimension. A roadmap is usually defined as a view for a specific time period. e.g. 1 year or the next 3 months.

Roadmaps may be dynamic and reflect the state of the object at any moment in real-time, or they may be static and show how a set of object looked at any moment in time.

Many EA objects can have multiple time attributes that represent different time properties.

- For example, an application component might have a set of lifecycle times that are associated with it, such as "Live" or "Sunset".
- Time attributes may simply be a single date, such as a milestone, or be a time period between two dates.

A roadmap view can consist of lanes. The lanes will show any theme or category for a set of objects.

Time usually flows from left to right on a roadmap diagram.

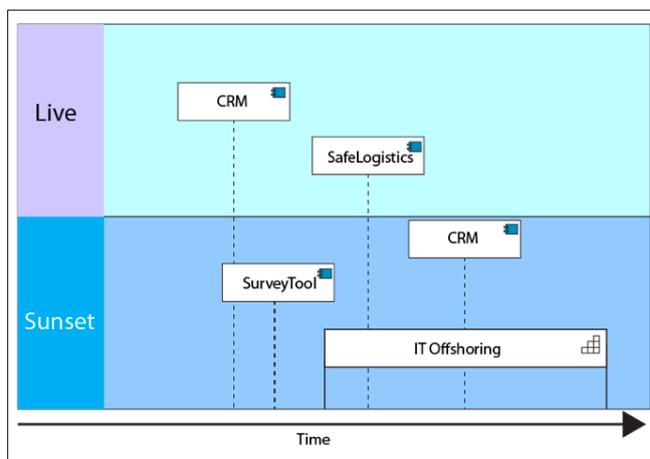


Figure 1 Simple Roadmap

---

As we can see in figure 1, we have two lanes; Live and Sunset. These are themes that we may well be interested in.

We are showing on a single roadmap view both application components (CRM, SafeLogistics, SurveyTool) and a business capability (IT Offshoring).

We can show application components with the live date attribute in the live lane.

We can also show the business capability but with a sunset time period. The time period is between two dates. We can show the application component sunset milestone dates in this sunset lane too.

In this example, it takes a much longer time to phase out a business capability and we only want to show the single milestone period for our applications.

This view gives us a good visibility of concepts with differing time dimensions.

We can also see for a single application component, its live and sunset dates in different lanes, so we can compare.

## **Roadmap snapshots**

We can take a snapshot of a roadmap at any particular moment in time by adding it to a publication. In this case it is static.

You can use a published document to look at differences in roadmaps over time.

A dynamic roadmap can be generated 'on the fly' and saved through the roadmap menu.

---

# Customization

## Import

**Role Applicable to:** [Client Administrator](#)

The platform provides a capability for importing objects, their attributes, and related objects.

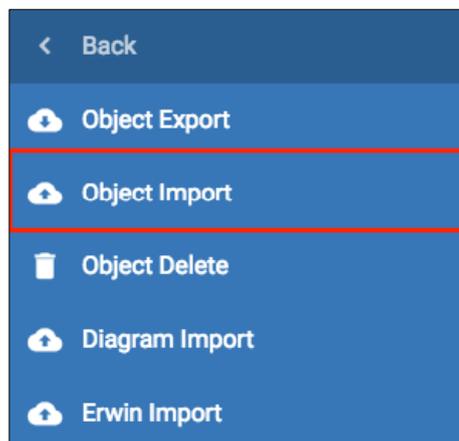
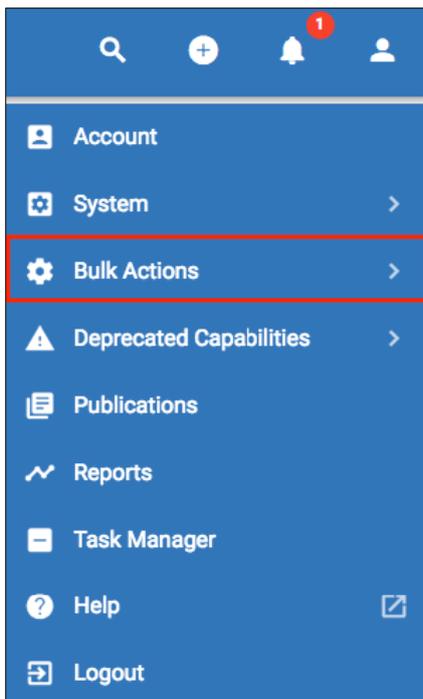
As of **5th August 2016** the import has received a major update, the file format is now **.csv** rather than **.tsv**, relationships are now imported in a separate file, and many of the required headers have been removed. .

Please note that erwin Professional Services can provide import support as a service. Contact [sales@erwin.com](mailto:sales@erwin.com)

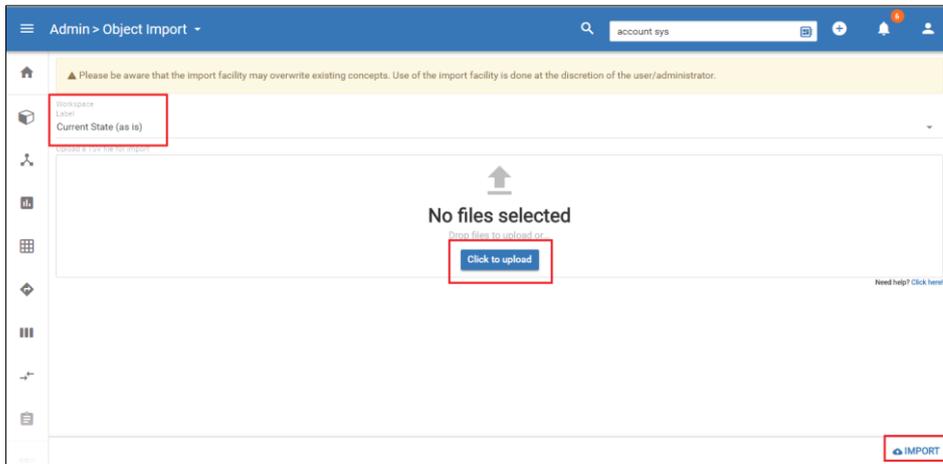
### Importing

Users of the import capability must have a knowledge of the [meta-model](#).

Import is available under the **User Menu > Bulk Actions > Object Import** menu



- Select the workspace that you would like to import into.
- Either, drag the file that contains the data you want to import into the upload box, or press the **Click to upload** button and select the file.



- Click the **IMPORT** button.

If successful, you will see a success message appearing on your screen.

For any other error, please examine carefully the contents of your CSV file.

## Example Imports

Below are 2 example sets of import files:

1. CSV files supplied at the link below (for Application component object type, ideal for Agile EA package).

[/userfiles/1621/2466/ckfinder/images/Corso\\_Agile\\_EA-Application\\_Component\\_import.zip?dc=201608160713-3](/userfiles/1621/2466/ckfinder/images/Corso_Agile_EA-Application_Component_import.zip?dc=201608160713-3)

2. CSV files supplied at the link below (for Requirement object type, ideal for Innovation Management package).

[/userfiles/1621/2466/ckfinder/images/Corso\\_Agile\\_EA-Tutorial%20Req%20Import.zip?dc=201608160710-2](/userfiles/1621/2466/ckfinder/images/Corso_Agile_EA-Tutorial%20Req%20Import.zip?dc=201608160710-2)

The downloaded files can be imported straight into the SPP tool. You do not need to unzip the files, you can simply import the zipped files!

Alternatively, you can download, unzip and view/edit the downloaded .csv files - adding attributes, relationships, and your own data entries, if you wish.

## Example walkthrough (Agile EA)

In this example, we are going to import a set of application components into the platform.

### Step 1 - Create any custom attributes

For this step, you need to create any attributes, which are custom to your import data, within the Agile EA meta model. Again, a good working knowledge of the meta model customization is important.

In the example, we can see that there are custom attributes such as **In Service Date**.

For the object type 'Application component', we can add a new attribute to match each attribute in the CSV file that we want to map and then import.

1	Name	Description	Technologies	Application Type	In Service Date	Effective Date	Perpetual License Cost	Yearly Maint	Yearly Rent	Build Cost	Cost Center	Vendor	Renewal Dat	Purchased Li	License Note	Requires Dat	Requires Dat	Provides Services	Consumes Services	Supports Components	Owner
2	Account Management Application	This is a system architect de COTS	java Eclipse			01/01/03	500	3			Corporate IT							"Competitor Analysis" "Assess Risk"	"Market Analysis"	"Account Administration" "Account Planning" Collections "Relationship Management"	
3	Account System	Handles all o XP	ASP dotNet "Windows Vista" "Windows"	COTS	01/03/05	01/01/09	300	42			Corporate IT									"Account Administration" "Credit Administration" "Relationship Management"	
4	Accounts Receivables	This is a system architect de COTS				01/01/10	1200	680	23	1300	Corporate IT							"Market Analysis" "Sales Processing System"	"Market Analysis" "Competitor Analysis"	"Account Administration" "Credit Administration" "Relationship Management"	
5	app 1	This is a system architect de COTS				05/01/12	40	4			Corporate IT										
6	ATM Card Management	This is a system architect de COTS				05/01/09	800	4			Corporate IT										
7	Automated Clearing House (ACH)	This is a system architect de COTS				12/01/01	90	5			Corporate IT										
8	BHIS System	This is a system architect de COTS				14/01/12	10	5			Corporate IT										
9	Business Applications	This is a system architect de COTS				11/01/09	120	5			Corporate IT										"Account Planning" "Credit Administration"

### Step 2 - Edit CSV file to match attributes

The only two **required** columns for a file to import are **Name** and **Object Type**.

Each column just requires the name of the attribute in the first row, the order of the columns does not matter.

So, we can now map our data to attributes as follows:

Name	Object Type	Description	Lifecycle - Live	Lifecycle - Phasi	Lifecycle status	Initial live date	Number of Users	Active Users	Action Cost Savings	
Web Portal	Application Component				31/03/2015, 16/1	Phasing Out		250	440	208500
Home and Away Financial Application	Application Component		23/06/2012, 31/12/2018		Live		23/06/2012	100	58	200000
CRM	Application Component	Customer relatio	07/03/2012, 31/12/2019		Live	07/03/2012		35	30	32000
Home and Away Policy Administration	Application Component	Managing home	01/01/2014, 31/12/2024		Live	24/01/2014		16	15	87999
Call Center Application	Application Component		04/01/2009, 30/09/2015		Live	04/01/2009		50	80	210000
Customer Data Access	Application Component		01/01/2013, 31/12/2017		Phasing Out			25	24	90000

(In this example the column names are highlighted yellow but that isn't required.)

At this point we have a simple import file (without relationships) that we could import.

### Step 3 - Map relationships in CSV file

The importer has the capability to import relationships and related objects into the platform.

- In order to map relationships, you need to create a separate .csv file (one file can contain relationships between any object types).
- The relationship file requires 6 columns (highlighted in yellow below)

- The objects in the relationship file must exist before a relationship can be created for them, if you upload an object file and a relationship file, the object file will always be imported first.

Relationship	Source Name	Relationship	Target Name	Source Object Name	Target Object Name	Source Object Type Name	Target Object Type Name
realizes		realized by		An employee must check the claim information	Revise claim handling process	Requirement	Goal
realizes		realized by		All claims shall be submitted online	Revise claim handling process	Requirement	Goal

## Step 4 - Import CSV file

- Simply ensure the file is saved as a .csv file and upload the file into the object import. You can upload zip files that contain multiple CSV's to import multiple object types at once.

## Important

- Import can no longer be used for deleting objects - please see the Object Delete page.
- Ensure your file is a **UTF-8** encoded comma separated file with extension .csv.
- If you are importing to amend existing objects, then the Object Name (case sensitive) and Object Type must be the same. The following rules apply:
  - If a value of an attribute in the CSV file is blank, then no amendment is made to the existing object attribute value.
  - Text and Numeric attribute values are replaced with any supplied values in the CSV file.
- Date attributes need to be entered in ISO8061 format (e.g 2016-12-31T12:00:00+00:00 would be 12pm on 31st December 2016 in GMT) this applies to all users regardless of location or timezone, as this is appended on the date.
- Commas can be used like they would in regular text except in list and array values, because you can select multiple values for these attributes. If you want a list or array value to have a comma in the name you need to surround the value in quotations e.g "One, Two and Three" this would then be understood to be a single value.

⚠ Please be aware that the import facility may overwrite existing concepts. Use of the import facility is done at the discretion of the user/administrator.

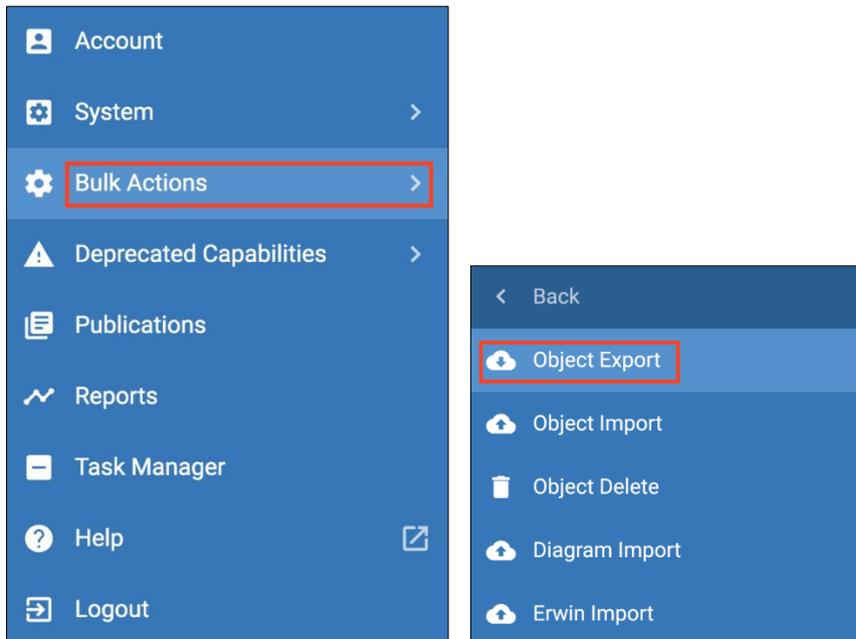
---

## Export

Users can export their data from the SPP platform into CSV format.

Exporting an object type can give you the template for import also.

**Object Export** is available under the **User > Bulk Actions** sub-menu.



- The first drop down available is to change the workspace you are exporting from.
- Then select the object type that you would like to export.
- You can apply filters similar to smart queries to limit the data being exported if you wish.
- You can see how many objects would be exported by checking the counters under each object type.
- You can add more than one object type by using the '+ Add Object Type' button.
- Select the Export option in the bottom right of the page.



A zip file containing your chosen data will be output, there will be one file for each object type and one file for relationships.

Object files will be named with today's date and time followed by the object type and the word object. Relationship files will be named with today's date and time followed by relationships.

---

# Scorecards

**Role Applicable to:** Client Administrator, User, Contributor, Reviewer

Scorecards can help you manage and evaluate metrics that can be applied to objects. With many objects, creating an automated score is the best way to calculate value.

Scorecards automatically appear as attributes within an object. Scorecards may be assigned to different object types.

Scorecards can be defined by a client administrator on the platform. Metrics (attributes) contain the values that are used in determining the score of an object.

*For example, a scorecard might use a person's age, credit rating, and employment status, to calculate the person's score.*

When a Scorecard is filled in for an object, it generates the scorecard **Score**.

**i** When defining a set of scorecards on the platform, an administrator may want to consider using Scorecards that have a consistent maximum value (e.g., 100) to allow comparison between objects across the portfolio.

## Name and Description

A name and description should be entered for each metric. This is important as it lays out the context for the metric within the scorecard.

A screenshot of a mobile application interface for configuring a scorecard. The form has a white background with a light gray border. At the top left is a back arrow icon. Below it are three input fields: 'Name' with the text 'Application Lifetime Cost', 'Description' with the text 'The cost of the application over its expected lifetime.', and 'Formula' with the text  $(\$2*\$3*12*\$1)+(\$1*\$5)+\$4$ . A blue button with a white plus sign and the text '+ Add Value' is located at the bottom right of the form.

Each metric is allocated an identifier such as \$1 or \$2. New metrics can be added using the **+Add Value** option

A screenshot of a mobile application interface showing a list of metrics for a scorecard. At the top, the 'Formula' field contains the text  $(\$2*\$3*12*\$1)+(\$1*\$5)+\$4+\$6$ . A blue button with a white plus sign and the text '+ Add Value' is located at the bottom right of the formula field. Below the formula field is a list of metrics, each with a name, a value, and a dropdown arrow. The first metric, 'Expected lifespan', has a value of '\$1' which is highlighted with a red box. The other metrics are 'Monthly User Cost (SaaS)' with value '\$2', 'Number of Users (SaaS)' with value '\$3', 'License Cost (Perpetual)' with value '\$4', and 'Maintenance Cost (Perpetual)' with value '\$5'.

Expected lifespan	\$1	▼
Monthly User Cost (SaaS)	\$2	▼
Number of Users (SaaS)	\$3	▼
License Cost (Perpetual)	\$4	▼
Maintenance Cost (Perpetual)	\$5	▼

---

The Formula to generate the score uses the identifiers as variables and works in the following way using arithmetic operators: \$1 + \$2

Valid arithmetic operators are:

Example	Name	Result
-\$1	Negation	Negative \$1
\$1 + \$2	Addition	Sum of \$1 and \$2
\$1 - \$2	Subtraction	Difference of \$1 and \$2
\$1 * \$2	Multiplication	Product of \$1 and \$2
\$1 / \$2	Division	\$1 divided by \$2

## Open Metrics

Metrics can be simple and consist of purely a metric name. The user will simply enter a value for the metric used in the scorecard equation, or an attribute value could be used. This will be an **Open** value

## Restricted Metrics

Each metric could be made **restricted**, to specify a continuous range of numeric values, or set increments.

For example, setting a value for a minimum of 1, and a maximum of 10, with an increment of 2 and a starting value of 2, will create 4 other range values.

The 4 ranges would be (4), (6), (8) and (10).

The screenshot shows a configuration window for a metric named 'Expected lifespan'. At the top, there are two tabs: 'OPEN' and 'RESTRICTED', with 'RESTRICTED' selected. Below the tabs, the 'Name' field contains 'Expected lifespan'. The 'Description' field contains the text 'What is the expected lifespan of this application in years?'. At the bottom, there is a table with four columns: 'Minimum', 'Maximum', 'Increment', and 'Initial'. The values in the table are: Minimum: 1, Maximum: 10, Increment: 1, and Initial: 5.

Minimum	Maximum	Increment	Initial
1	10	1	5

This would offer the user the options below for this metric, as part of the overall scorecard



## Formula

A formula can be a mix of identifiers and constants, separated by operators;

<b>Name</b>
Future Market Share
<b>Description</b>
A calculation of the future market share.
<b>Formula</b>
$\frac{2}{1} * 100 + 3$

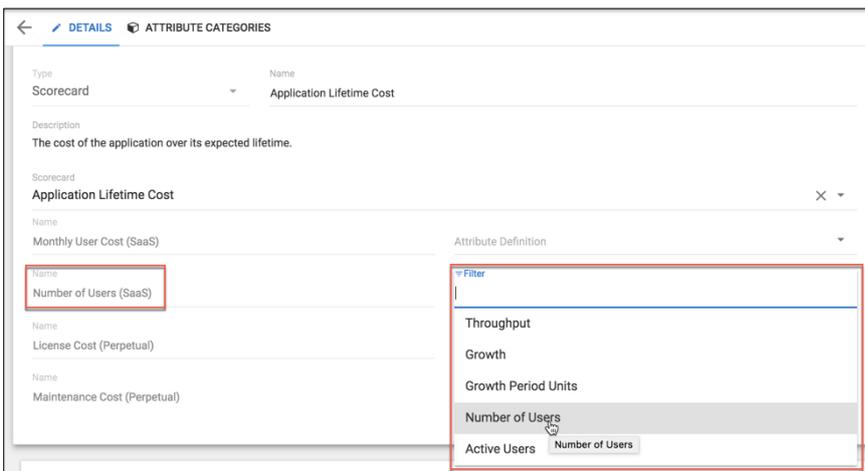


Parentheses () may be used to change the order that operators are processed. The standard algebraic order of operations applies; (Parentheses, Multiplication and Division, Addition and Subtraction).

## Metrics made up of other attributes

Metrics can be made up of other attributes. It may be useful to reference the value of another attribute in the calculation of a scorecard value.

The referenced attribute can be selected when adding a scorecard attribute to an Object type.



To do this, click on the arrow to the right of the name of the field (in the Attribute Definition field). A drop down list of attributes that can be selected will be presented.

---

## Examples of scorecards

### Scorecards for Innovation Management

#### ***Scorecard – Innovation Score***

- Attractiveness – High = 10, Medium = 5, Low = 1
- Time to Market – 1 Month = 10, 2-6 Months = 8, 6-12 Months = 6, 1-2 Years = 4, 2-3 Years = 2, 3 Years+ = 1
- Anticipated Risk – High = -5, Medium = -3 Low = -1
- Development Effort – High = -10, Medium = -5, Low = -1
- Increase in Brand Value

Innovation Score = Attractiveness + Time to Market + Anticipated Risk + Development Effort + Increase in Brand Value

#### ***Scorecard – Requirements Score***

- Resources Required - High = 10, Medium = 5, Low = 1
- Resource Experience rating
- Business Alignment - High = 10, Medium = 5, Low = 1
- IT Alignment - High = 10, Medium = 5, Low = 1
- Project costs – High = -10, Medium = -5, Low = -1

Requirements Score = (Resources Required / Resource Experience Rating) + Business Alignment + IT Alignment + Project Costs

#### ***Scorecard - Marketing Score***

- Brand Impact
- Word of Mouth Potential
- Fits customer demographic
- Case studiable (weighting x 2)

Marketing Score = Brand Impact + Word of Mouth Potential + Fits customer demographic + (2 x Case studiable).

---

# Customizable metamodel

**Role Applicable to:** [Client Administrator](#), [User](#)

Each client area is created with the default system metamodel.

Often organizations require the ability to configure the underlying metamodel within the platform. The entire metamodel may be customized. Types of objects can be added (Object Types), along with attributes and new relationship types.

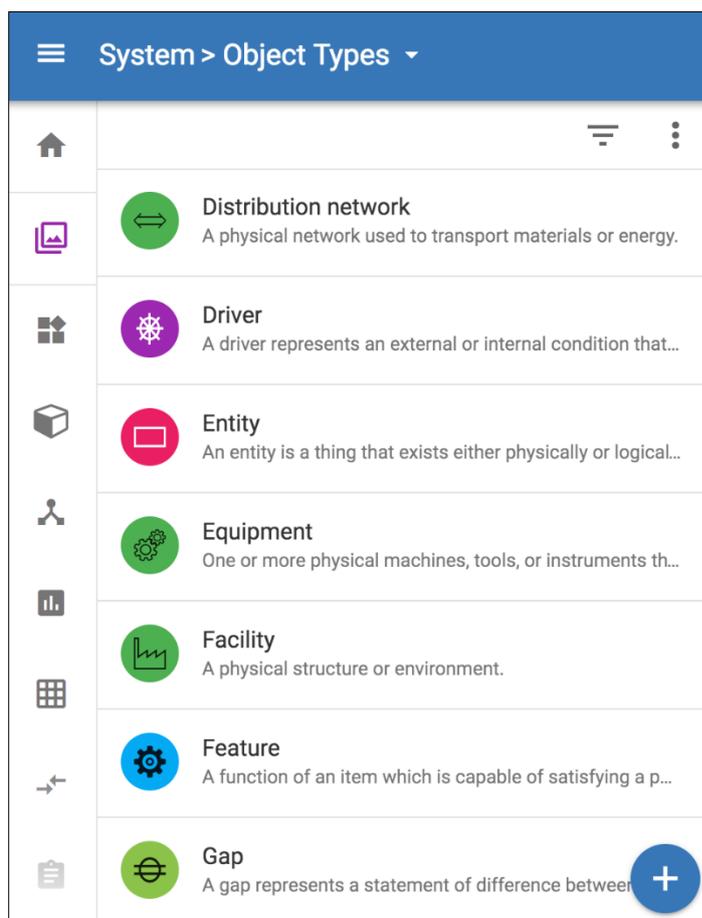
Additional metrics can be added via the [Scorecard](#) capability.

A metamodel is consistent for an entire client area. All workspaces conform to the core metamodel of a client area.

## Object Types

Object types can be customized, by adding new types or deleting existing types.

To view a list of existing object types; access the **User** menu > **System** > **Object Types**



Object types can be created by Client Administrators, by clicking on the + button in the bottom right of the list.



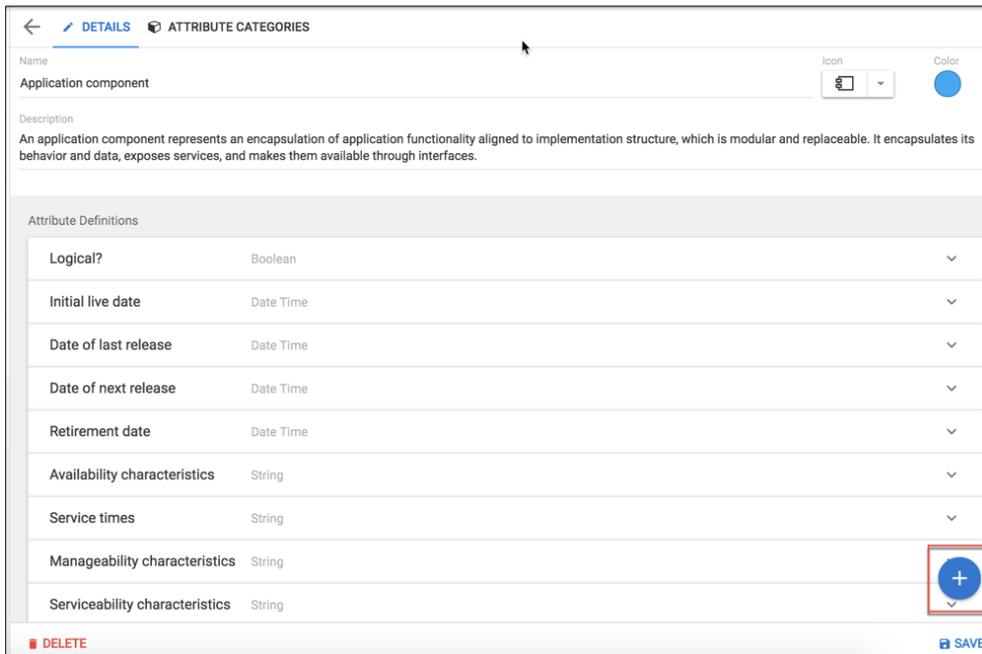
An object type will require a Name and an Icon, before it can be saved. The Name must be between 2 and 50 characters in length. An Icon can be chosen from 100's of available images. It is good practice to also provide a short description, describing the purpose of the object type and select a colour.

Object types can be deleted by Client Administrators, by clicking on **DELETE** in the bottom left of the DETAILS edit screen.

## Attributes

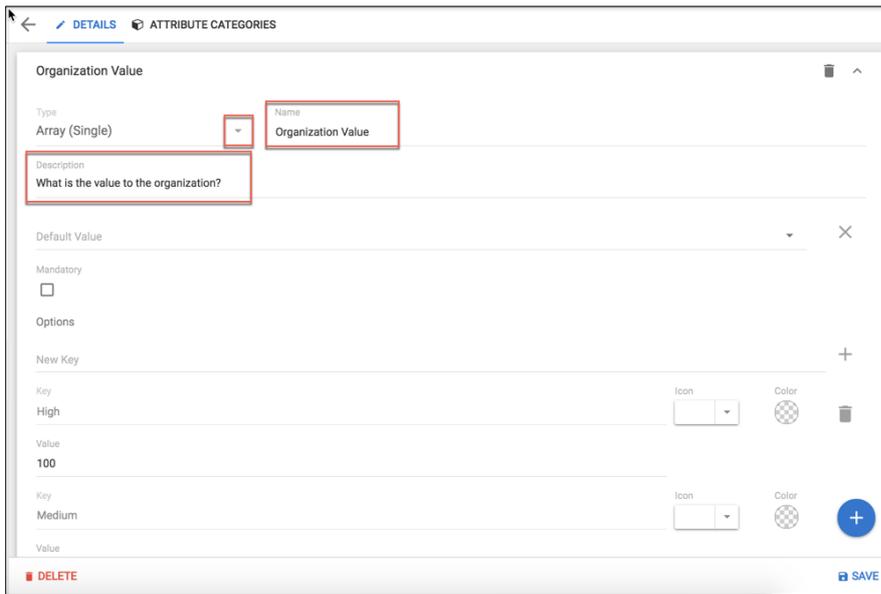
Attributes are used for storing information in objects. All objects will have a Name and a Description, but it is expected that more specific detail will be required.

Attributes can be created by Client Administrators, by clicking the blue circle with the white plus sign, found in the bottom right-hand corner of the screen



Clicking this button will create a new entry at the bottom of the existing attribute list.

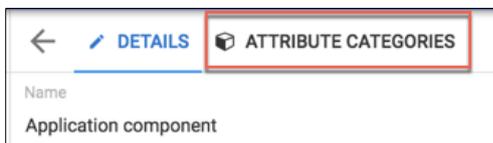
Each attribute will require its Attribute Type to be identified, by selecting from the drop down list of available types. A Name will also be required, and a short description is recommended.



In the example above, the Array attribute type has been selected. It requires a series of Key:Value pairs to be devised, but other attribute types will have different settings. A default value can then be chosen, that each instance of the attribute is initially populated with.

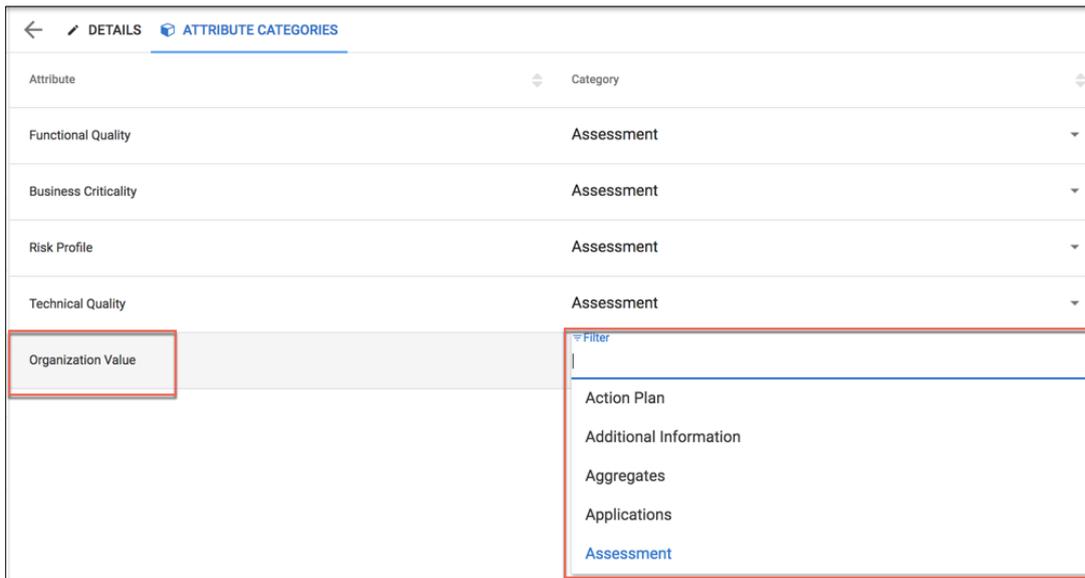
**i** | Once the Attribute Type has been selected it cannot be changed. The attribute Name can be modified.

Each Attribute should have a Category assigned to it, particularly if many attributes are to be created.



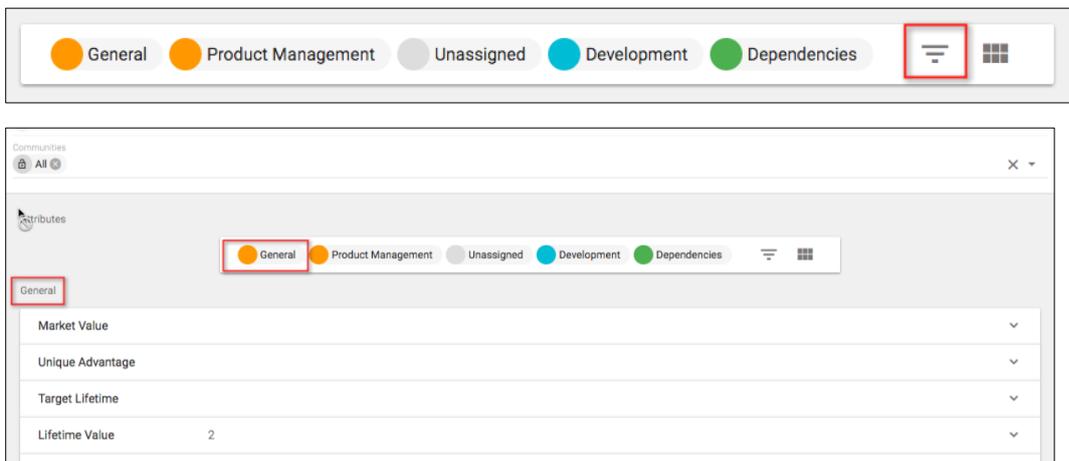
Select the **ATTRIBUTE CATEGORIES** tab from with the object type

Find the attribute within the list, select the drop down next to the entry and choose the attribute category that you would like it to appear in.



Within an object instance, the attribute can either be searched for, by name, or the attribute category can be selected from a list of existing categories. This will filter the attribute list to the categories chosen and make it easier to find an attribute in a long list.

Users can also select the coloured circle and drag the category to where they would like to see it within the object, changing the order of how things are displayed.



Attribute Categories can be edited by Client Administrators, by accessing the **User** menu > **System** > **Attribute Categories**.

**i** Any existing Attribute Groups will be converted to Attribute Categories and will be found in the list.

## Attribute Types

Attribute Type	Description	Applies to	Example
<b>Array (Single and Multiple)</b>	Supports a list of 'key-value pairs'.	A need to store values associated with descriptive labels. Colours and icons can be set for these values, so they can be represented in table and other visualizations.	Impact: High: 50, Medium: 30, Low: 10
<b>Boolean</b>	Supports a "true" or "false" value.	Used when a simple on/off 'switch' is required.	Strategic?: true
<b>Date Time</b>	Displays a date (and time).	Used when a 'point-in-time' is to be specified.	Installation Date: 01/01/2020 15:15 PM
<b>Date Time Range</b>	Supports a 'start date' and an 'end date'.	Used to describe a 'region-of-time', or range of dates.	In Production: 01/01/2017 09:00 AM, 01/01/2020 17:30 PM
<b>Date Time Rollup</b>	Aggregates dates from related objects.	An automatic update of a Date or Date-Range value, calculating the first (Min), or last (Max) date value from related objects.	Duration: 01/01/2016 12:00 AM, 01/01/2020 17:30 PM
<b>Fraction</b>	Allows for 'non-integer' numbers	Values that require a fractional part, or non-whole number.	Production Cost of Item: 4.82
<b>Integer</b>	Supports 'whole' numbers	Values that require whole numbers.	Resource Required: 4
<b>List (Single and Multiple)</b>	A number of values.	Providing a list of values, from which the user can choose. Colours and icons can be set for these values, so they can be represented in table and other visualizations.	IT Alignment: High, Medium, Low

Attribute Type	Description	Applies to	Example
<b>Numeric Rollup</b>	Aggregates numeric values from related objects.	An automatic update of an Integer or Fraction value, calculating the total (Sum), lowest (Min), or highest (Max) numeric value from related objects.	Overall Resource Requirements: 37
<b>Relationship-List</b>	Provides for a selection of related objects.	The need to identify a subset of related objects, from a larger group of related objects.	Primary Service: Claim Registration
<b>Scorecard</b>	Calculates a numeric value based on input from numeric values and/or the object's numeric attributes.	A need to dynamically calculate a number that could be used to possibly indicate a priority, or level of importance, by employing a simple formula that uses other values from within the object.	Marketing Score: 77
<b>String</b>	A number of characters.	Descriptive content.	Business Impact: Impacts the majority of our customer facing processes.
<b>URL</b>	A web browser reference to a web page.	Support a 'navigateable' link to web content, which could be another object, a resource or webpage.	Webpage: <a href="https://www.erwin.com">https://www.erwin.com</a>

Attributes can be set to **Mandatory** if a user wishes them to have to be filled in.

---

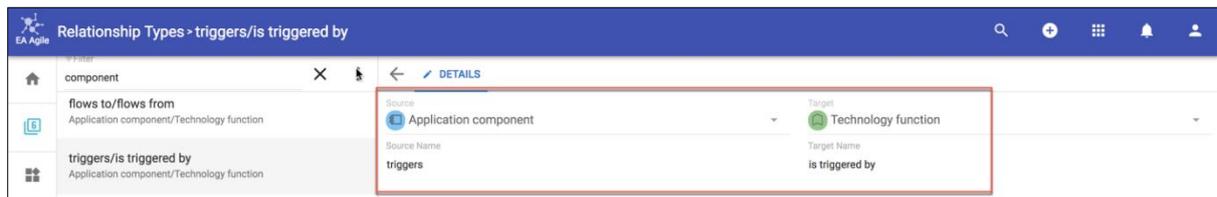
## Relationship Types

Relationship Types are used to link object types together. As standard, we use the relationship types provided by ArchiMate from the Open Group.

However, you may use and define your own relationship types.

To view a list of existing Relationship Types; access the **User** menu > **System** > **Relationship Types**

Relationship Types can be created by Client Administrators, by clicking on the **+** button in the bottom right of the list.



They need to be populated with Source / Target Object Types and Source Name / Target Name relationships

The relationship type currently needs to be created for each object type pairing.

---

# Visio Import

**Role Applicable to:** Client Administrator, Professional

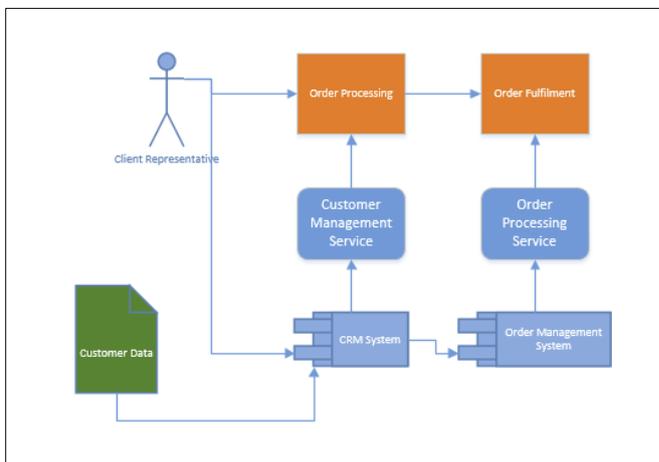
Import Visio 2013 diagrams into the SPP platform

## Step 1 - Create a new diagram

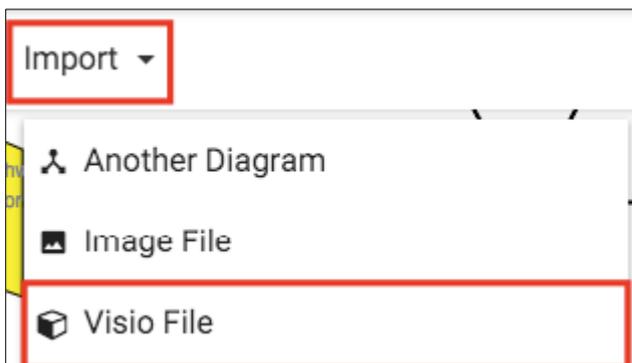
Within the SPP, create a new instance of the diagram type you would like to import the Visio diagram as (this can be an out of the box or a custom diagram type)

Save the diagram.

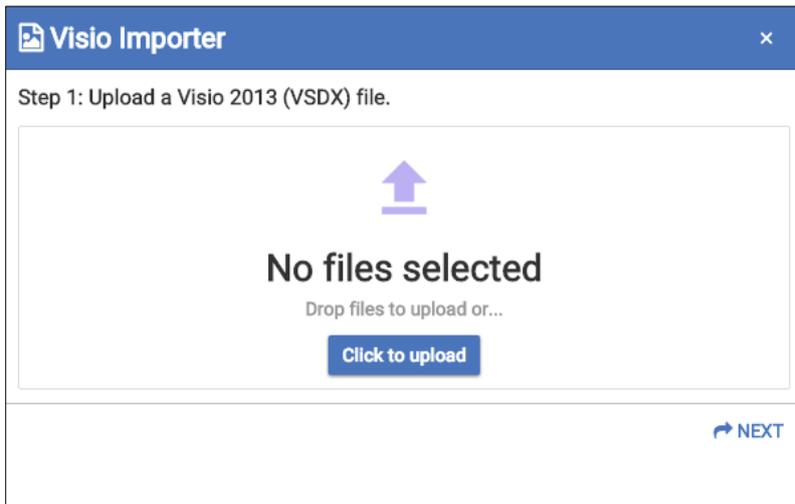
## Step 2 - Import the Visio file



From within the diagram, select the **Import** menu and select the **Visio File** option.

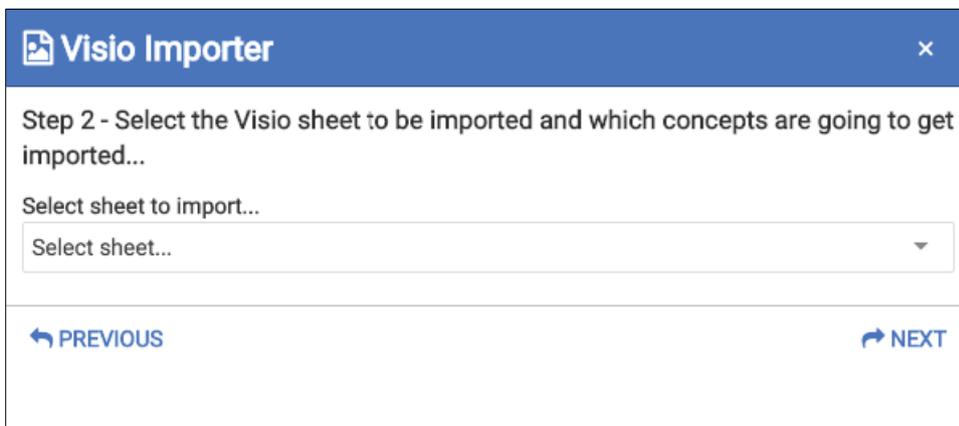


When asked, select 'Click to upload' to upload a VSDX file (containing your diagram) from your machine, for mapping



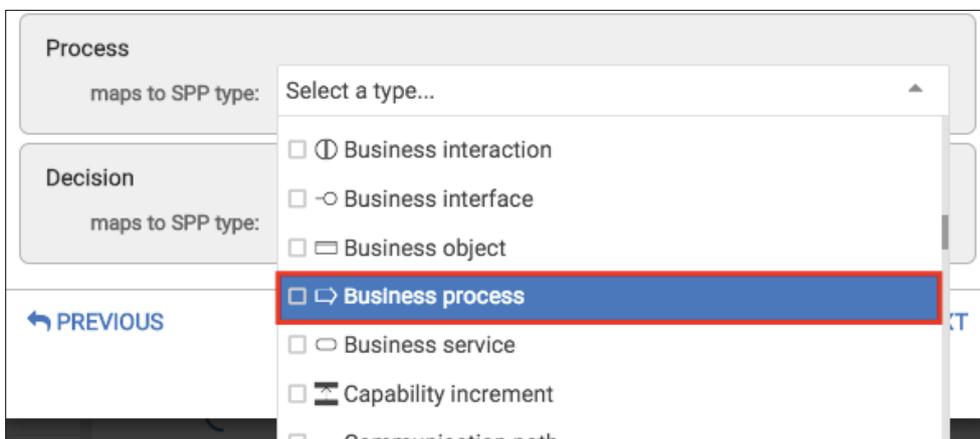
When the file is loaded, select the Next option

Select the sheet you would like to use from the VSDX file, using the drop down arrow



Once the sheet has been chosen, the shapes identified from the sheet will be presented for mapping

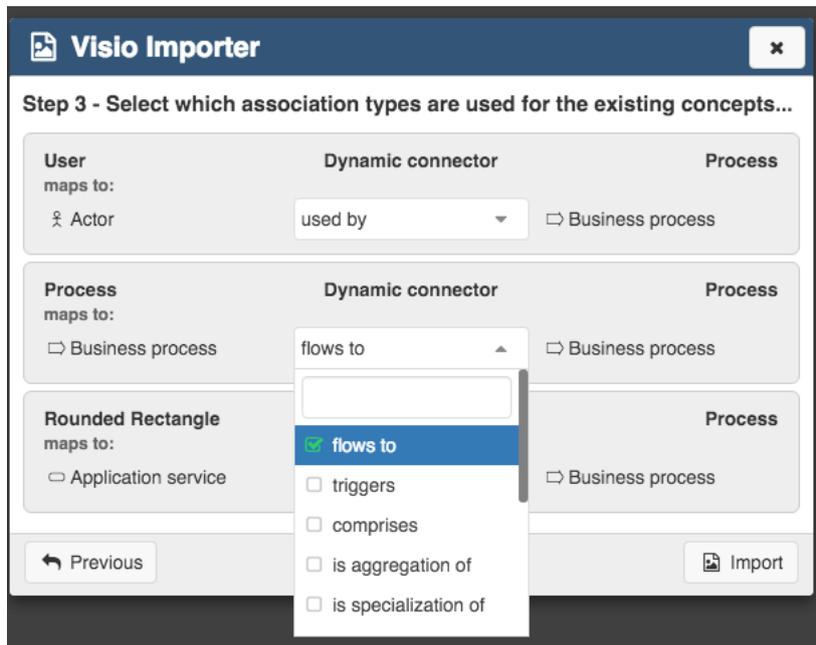
Select the object type you would like to map to each identified shape, using the drop downs for each shape presented (the object type offered are the ones allowed on the diagram type you chose)



Once all of the shapes have been mapped, select the Next option

---

Map all of the associations types that are used by existing concepts on the diagram



Once the associations have been mapped, select the next option

The diagram is now imported into your SPP diagram.

---

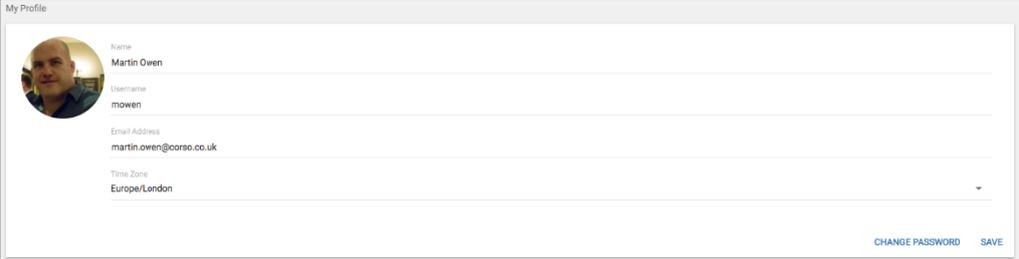
# Settings

## Account profile

**Role Applicable to:** [Client Administrator](#), [User](#), [Contributor](#), [Reviewer](#)

Your account profile allows you to see and amend your username, avatar and a description of yourself, it can be accessed via the Admin button in the user menu.

There is also a button to change your password.



My Profile

Name  
Martin Owen

Username  
mowen

Email Address  
martin.owen@corso.co.uk

Time Zone  
Europe/London

CHANGE PASSWORD SAVE

If you feel that you require further privileges, please contact your client administrator.

If you are unsure who your client administrator is, please contact [support](#).

---

# Tags

Tags are a mechanism for labelling any object in the platform. The platform allows users to create and manage labels (or “tags”) that categorize content using simple keywords.

The users add the tags. The tags are clearly visible and can have a defined color, and are themselves links to other items that share that keyword tag. The colors of tags can be managed on the platform.

Tags can also be removed from an object. Tags are visible across an entire project regardless of user.

To manage your existing tags and create new ones you should navigate to **User Menu > Account > Tags**. From here you will be shown all of the currently available tags and will have the option to add new tags.

**i** | If you want to add special characters as tags, why not copy special characters into a tag? Most modern browsers support special characters. An example web site is here: <http://copypastecharacter.com/>

---

# Licensing

**Role Applicable to:** [Client Administrator](#),

Licenses are managed by the administrator. When a client first signs up for licenses on the platform, they may purchase either full licenses, contributor licenses, or reviewer licenses. The administrator may allocate the licenses with the User Privileges capability.

See the [User Management](#) section for more details.

---

# Metamodel

## What is a meta model?

A meta model is a description of a classification of a set of things.

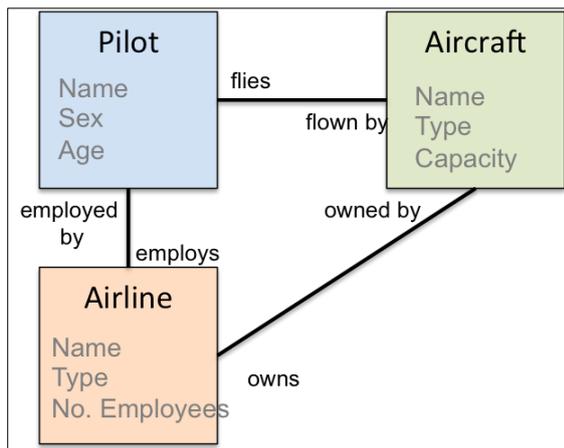
It describes the types of thing (object types), the rules between them (relationships) and the attributes that describe them.

Meta models can usually be built to describe any type of problem or scenario.

They are often graphical and are represented by boxes and lines. Boxes indicate the object types and lines represent the relationships.

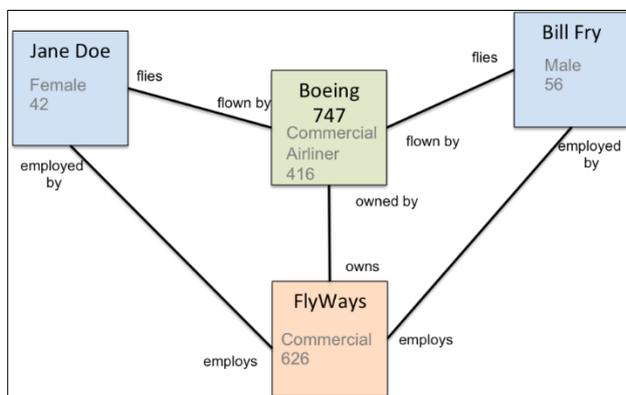
Meta models are used by end users to express models.

In the meta model example below, we could describe the relationship between a Pilot, Airline and Aircraft.



In the model below, we use the metamodel to guide the construction of the model. The meta model provides the framework within which the model can be constructed.

We call the instances of the object types - Objects.



---

## Meta models and the SPP

Each package in the erwin Data Foundation Platform has a meta model. The meta model can be changed. Further guidance is provided [here](#).

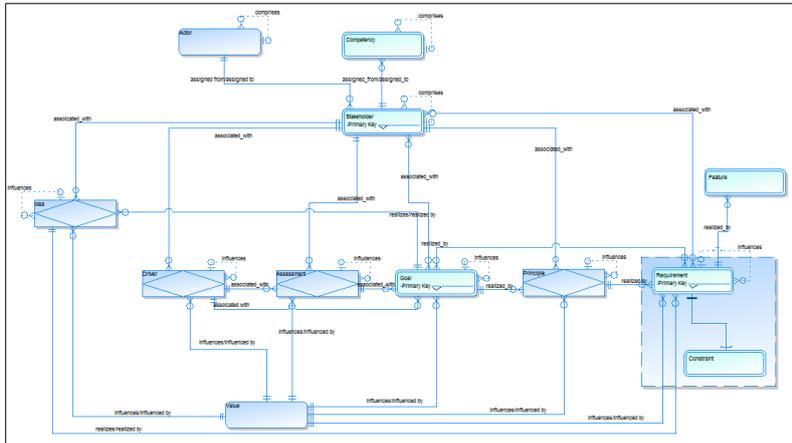
The entire erwin Data Foundation Platform is a meta model in its own right. Each package is a subset of the entire meta model and some object types and relationships overlap.

---

# Innovation management metamodel

**Package:** Innovation Management

**Role Applicable to:** Client Administrator, User, Contributor, Reviewer



Some of the objects and metamodel are based on the ArchiMate specification from the Open Group. However, this has been significantly extended to support innovation management.

erwin licenses the ArchiMate specification from the open group. The terms of the license are [here](#).

# Enterprise architecture metamodel

**Package:** Agile Enterprise Architecture

**Role Applicable to:** Client Administrator, User, Contributor, Reviewer

The enterprise architecture metamodel is based on ArchiMate and TOGAF 9 attributes. The metamodel includes the following:

- ArchiMate 2.1 Core
- ArchiMate 2.1 Implementation and Migration Extension
- ArchiMate 2.1 Motivation Extension
- **Capability** Extension

The metamodel has been extended from ArchiMate 2.1 to reflect the Open Groups' latest thinking around **capabilities**. A working whitepaper is available through the open group.

## Attributes

The ArchiMate language does not define any explicit attributes for object types. Therefore, all attributes for the metamodel are mapped from TOGAF 9.1 attributes. Where object types are specializations of other types, then we have mapped the super type attributes to the sub type. For example, a Metric will inherit all of the attributes of Driver as it is a specialism of a Driver.

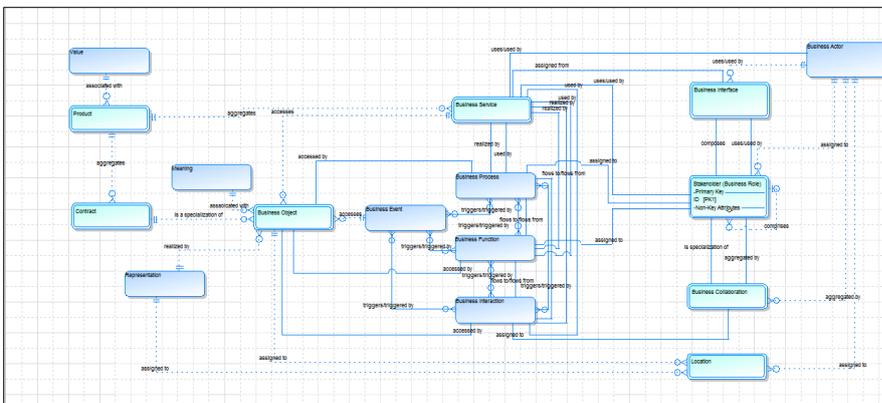
The platform is extensible and you may add your own attributes to any object type. See the **customization** section for more details.

A link to the specification exists [here](#).

The mappings are based on the harmonization work carried out at the open group between TOGAF and ArchiMate. The link to this work is [here](#). However, you must be an Open Group member to access these documents.

The following table shows the ArchiMate to TOGAF object type mappings and the attributes that have been mapped:

## Business Domain Metamodel



ArchiMate Element	TOGAF Element	Attributes										
Business Actor	Actor	<table border="1"> <tr> <td># FTEs</td> <td>Estimated number of FTEs that operate as this Actor.</td> </tr> <tr> <td>Actor goal</td> <td>Objectives that this actor has, in general terms.</td> </tr> <tr> <td>Actor tasks</td> <td>Tasks that this actor performs, in general terms.</td> </tr> </table>	# FTEs	Estimated number of FTEs that operate as this Actor.	Actor goal	Objectives that this actor has, in general terms.	Actor tasks	Tasks that this actor performs, in general terms.				
# FTEs	Estimated number of FTEs that operate as this Actor.											
Actor goal	Objectives that this actor has, in general terms.											
Actor tasks	Tasks that this actor performs, in general terms.											
	Organization Unit	<table border="1"> <tr> <td>Headcount</td> <td>Number of FTEs working within the organization. (Note that this attribute is merged with # FTEs)</td> </tr> </table>	Headcount	Number of FTEs working within the organization. (Note that this attribute is merged with # FTEs)								
Headcount	Number of FTEs working within the organization. (Note that this attribute is merged with # FTEs)											
Business Collaboration	N/A	This object type has the default description attribute.										
Business Function	Function	<table border="1"> <tr> <td>Standard Class</td> <td>Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, RetiredStandard</td> </tr> <tr> <td>Standard creation date</td> <td>If the product is a standard, when the standard was created.</td> </tr> <tr> <td>Last standard review date</td> <td>Last date that the standard was reviewed.</td> </tr> <tr> <td>Next standard review date</td> <td>Next date for the standard to be reviewed.</td> </tr> <tr> <td>Retire date</td> <td>Date when the standard was/will be retired.</td> </tr> </table>	Standard Class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, RetiredStandard	Standard creation date	If the product is a standard, when the standard was created.	Last standard review date	Last date that the standard was reviewed.	Next standard review date	Next date for the standard to be reviewed.	Retire date	Date when the standard was/will be retired.
Standard Class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, RetiredStandard											
Standard creation date	If the product is a standard, when the standard was created.											
Last standard review date	Last date that the standard was reviewed.											
Next standard review date	Next date for the standard to be reviewed.											
Retire date	Date when the standard was/will be retired.											
Business Interface	N/A	This object type has the default description attribute.										

Business Object	Data Entity	<table border="1"> <tr> <td data-bbox="778 297 991 432">Category</td> <td data-bbox="991 297 1401 432">The following categories of data entity apply: Message, Internally Stored Entity.</td> </tr> <tr> <td data-bbox="778 432 991 533">Privacy classification</td> <td data-bbox="991 432 1401 533">Level of restriction placed on access to the data.</td> </tr> <tr> <td data-bbox="778 533 991 633">Retention classification</td> <td data-bbox="991 533 1401 633">Level of retention to be placed on the data.</td> </tr> </table>		Category	The following categories of data entity apply: Message, Internally Stored Entity.	Privacy classification	Level of restriction placed on access to the data.	Retention classification	Level of retention to be placed on the data.										
Category	The following categories of data entity apply: Message, Internally Stored Entity.																		
Privacy classification	Level of restriction placed on access to the data.																		
Retention classification	Level of retention to be placed on the data.																		
Business Process	Process	<table border="1"> <tr> <td data-bbox="778 689 991 869">Standards class</td> <td data-bbox="991 689 1401 869">Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard</td> </tr> <tr> <td data-bbox="778 869 991 969">Standard creation date</td> <td data-bbox="991 869 1401 969">If the product is a standard when the standard was created.</td> </tr> <tr> <td data-bbox="778 969 991 1070">Last standard review date</td> <td data-bbox="991 969 1401 1070">Last date that the standard was reviewed.</td> </tr> <tr> <td data-bbox="778 1070 991 1171">Next standard review date</td> <td data-bbox="991 1070 1401 1171">Next date for the standard to be reviewed.</td> </tr> <tr> <td data-bbox="778 1171 991 1272">Retire date</td> <td data-bbox="991 1171 1401 1272">Date when the standard was/will be retired.</td> </tr> <tr> <td data-bbox="778 1272 991 1373">Process criticality</td> <td data-bbox="991 1272 1401 1373">Criticality of this process to business operations.</td> </tr> <tr> <td data-bbox="778 1373 991 1473">Manual or automated</td> <td data-bbox="991 1373 1401 1473">Whether this process is supported by IT or is a manual process.</td> </tr> <tr> <td data-bbox="778 1473 991 1570">Process volumetrics</td> <td data-bbox="991 1473 1401 1570">Data on frequency of process execution.</td> </tr> </table>		Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard	Standard creation date	If the product is a standard when the standard was created.	Last standard review date	Last date that the standard was reviewed.	Next standard review date	Next date for the standard to be reviewed.	Retire date	Date when the standard was/will be retired.	Process criticality	Criticality of this process to business operations.	Manual or automated	Whether this process is supported by IT or is a manual process.	Process volumetrics	Data on frequency of process execution.
Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard																		
Standard creation date	If the product is a standard when the standard was created.																		
Last standard review date	Last date that the standard was reviewed.																		
Next standard review date	Next date for the standard to be reviewed.																		
Retire date	Date when the standard was/will be retired.																		
Process criticality	Criticality of this process to business operations.																		
Manual or automated	Whether this process is supported by IT or is a manual process.																		
Process volumetrics	Data on frequency of process execution.																		
Role	Role	<table border="1"> <tr> <td data-bbox="778 1615 895 1715"># FTEs</td> <td data-bbox="895 1615 1401 1715">Estimated number of FTEs that operate as this Role.</td> </tr> </table>		# FTEs	Estimated number of FTEs that operate as this Role.														
# FTEs	Estimated number of FTEs that operate as this Role.																		

Business Service	Service	<table border="1"> <tr> <td data-bbox="778 297 1082 555">Standards class</td> <td data-bbox="1082 297 1394 555">Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard</td> </tr> <tr> <td data-bbox="778 555 1082 689">Standard creation date</td> <td data-bbox="1082 555 1394 689">If the product is a standard when the standard was created.</td> </tr> <tr> <td data-bbox="778 689 1082 790">Last standard review date</td> <td data-bbox="1082 689 1394 790">Last date that the standard was reviewed.</td> </tr> <tr> <td data-bbox="778 790 1082 891">Next standard review date</td> <td data-bbox="1082 790 1394 891">Next date for the standard to be reviewed.</td> </tr> <tr> <td data-bbox="778 891 1082 981">Retire Date</td> <td data-bbox="1082 891 1394 981">Date when the standard was/will be re</td> </tr> </table>	Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard	Standard creation date	If the product is a standard when the standard was created.	Last standard review date	Last date that the standard was reviewed.	Next standard review date	Next date for the standard to be reviewed.	Retire Date	Date when the standard was/will be re				
Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard															
Standard creation date	If the product is a standard when the standard was created.															
Last standard review date	Last date that the standard was reviewed.															
Next standard review date	Next date for the standard to be reviewed.															
Retire Date	Date when the standard was/will be re															
Contract	Contract	<table border="1"> <tr> <td data-bbox="778 1081 1082 1216">Behavior characteristics</td> <td data-bbox="1082 1081 1394 1216">Functional behavior to be supported within the scope of the contract.</td> </tr> <tr> <td data-bbox="778 1216 1082 1272">Service name "caller"</td> <td data-bbox="1082 1216 1394 1272">Consuming service.</td> </tr> <tr> <td data-bbox="778 1272 1082 1328">Service name "called"</td> <td data-bbox="1082 1272 1394 1328">Providing service.</td> </tr> <tr> <td data-bbox="778 1328 1082 1507">Service quality characteristics</td> <td data-bbox="1082 1328 1394 1507">Non-functional behavior to be supported within the scope of the contract.</td> </tr> <tr> <td data-bbox="778 1507 1082 1641">Availability characteristics</td> <td data-bbox="1082 1507 1394 1641">Degree to which something is available for use</td> </tr> <tr> <td data-bbox="778 1641 1082 1776">Service times</td> <td data-bbox="1082 1641 1394 1776">Hours during which the service must be available</td> </tr> <tr> <td data-bbox="778 1776 1082 1955">Manageability characteristics</td> <td data-bbox="1082 1776 1394 1955">Ability to gather information about the state of something and control it</td> </tr> </table>	Behavior characteristics	Functional behavior to be supported within the scope of the contract.	Service name "caller"	Consuming service.	Service name "called"	Providing service.	Service quality characteristics	Non-functional behavior to be supported within the scope of the contract.	Availability characteristics	Degree to which something is available for use	Service times	Hours during which the service must be available	Manageability characteristics	Ability to gather information about the state of something and control it
Behavior characteristics	Functional behavior to be supported within the scope of the contract.															
Service name "caller"	Consuming service.															
Service name "called"	Providing service.															
Service quality characteristics	Non-functional behavior to be supported within the scope of the contract.															
Availability characteristics	Degree to which something is available for use															
Service times	Hours during which the service must be available															
Manageability characteristics	Ability to gather information about the state of something and control it															

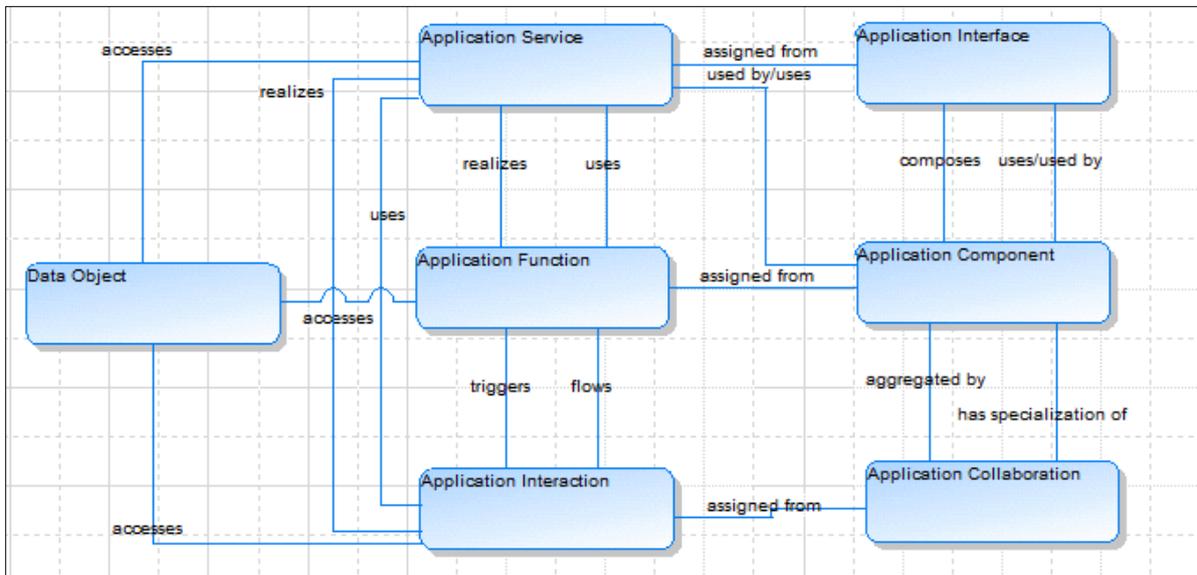
		Serviceability characteristics	Ability to identify problems and take corrective action, such as to repair or upgrade a component in a running system
		Performance characteristics	Ability of a component to perform its tasks in an appropriate time.
		Response requirement	Response times that the service provider must meet for particular operations.
		Reliability characteristics	Resistance to failure.
		Quality of information required	Contracted requirements on accuracy and completeness of information
		Contract control requirements	Level of governance and enforcement applied to the contractual parameters for overall service.
		Result control requirements	Measures in place to ensure that each service request meets contracted criteria
		Recoverability characteristics	Ability to restore a system to a working state after an interruption
		Locatability characteristics	Ability of a system to be found when needed
		Security characteristics	Ability of a system to prevent unauthorized access to functions and data

		Privacy characteristics	Protection of data from unauthorized access
		Integrity characteristics	Ability of a system to ensure that data has not been corrupted
		Credibility characteristics	Ability of a system to ensure that the service request originates from an authorized source
		Localization characteristics	Ability of a service to support localized variants for different consumer groups
		Internationalization characteristics	Ability of a service to support international variations in business logic and data representation (such as character set)
		Interoperability characteristics	Ability of the service to interoperate with different technical environments, inside and outside of the organization.
		Scalability characteristics	Ability of the service to grow or shrink its performance or capacity appropriately to the demands of the environment in which it operates
		Portability characteristics	Of data, people, applications, and components.
		Extensibility characteristics	Ability to accept new functionality.

		Capacity characteristics	Contracted capacity of the service provider to meet requests
		Throughput	Required throughput capacity.
		Throughput period	Time period needed to deliver throughput capacity
		Growth period	Expected future growth rate of service request
		Peak profile short term	Short-term profile of peak service traffic
		Peak profile long term	Long-term profile of peak service traffic
Business Event	Event	This object type has the default description attribute.	
Business Interaction	Process or Function	Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard
		Standard creation date	If the product is a standard when the standard was created.
		Last standard review date	Last date that the standard was reviewed.
		Next standard review date	Next date for the standard to be reviewed.
		Retire Date	Date when the standard was/will be retired,
Location	Location	Category	The following categories of Location apply: Region (applies to a grouping of countries or territory, e.g., South East Asia, UK, and Ireland), Country (applies to a single country; e.g., US), Building (applies to a site of operation; where

		<p>several offices are collected in a single city, this category may represent a city), and Specific Location (applies to any specific location within a building, such as a server room). The nature of the business may introduce other Locations: Ship or Port for a ferry company, Mine for a gold company, Car for a police force, Hotel for any firm's traveling workers, and so on.</p>
Meaning	N/A	This Object type has the default description attribute.
Product	Product	This Object type has the default description attribute.
Representation	N/A	This Object type has the default description attribute.
Value	N/A	This Object type has the default description attribute.

### Application Domain Metamodel



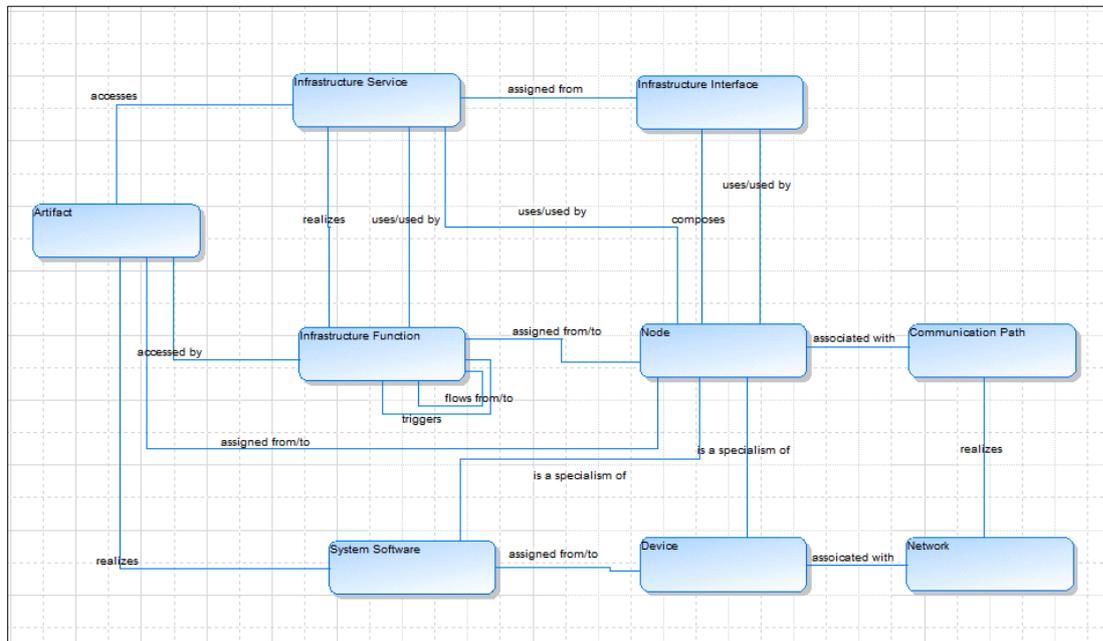
ArchiMate Element	TOGAF Element	Attributes										
Application Service	Information System Service	<table border="1"> <tr> <td>Standards class</td> <td>Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard</td> </tr> <tr> <td>Standard creation date</td> <td>If the product is a standard when the standard was created.</td> </tr> <tr> <td>Last standard review date</td> <td>Last date that the standard was reviewed.</td> </tr> <tr> <td>Next standard review date</td> <td>Next date for the standard to be reviewed.</td> </tr> <tr> <td>Retire Date</td> <td>Date when the standard was/will be retired,</td> </tr> </table>	Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard	Standard creation date	If the product is a standard when the standard was created.	Last standard review date	Last date that the standard was reviewed.	Next standard review date	Next date for the standard to be reviewed.	Retire Date	Date when the standard was/will be retired,
Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard											
Standard creation date	If the product is a standard when the standard was created.											
Last standard review date	Last date that the standard was reviewed.											
Next standard review date	Next date for the standard to be reviewed.											
Retire Date	Date when the standard was/will be retired,											
Data Object	Data Entity	<table border="1"> <tr> <td>Category</td> <td>The following categories of data entity apply: Message, Internally Stored Entity</td> </tr> <tr> <td>Privacy classification</td> <td>Level of restriction placed on access to the data</td> </tr> <tr> <td>Retention classification</td> <td>Level of retention to be placed on the data</td> </tr> </table>	Category	The following categories of data entity apply: Message, Internally Stored Entity	Privacy classification	Level of restriction placed on access to the data	Retention classification	Level of retention to be placed on the data				
Category	The following categories of data entity apply: Message, Internally Stored Entity											
Privacy classification	Level of restriction placed on access to the data											
Retention classification	Level of retention to be placed on the data											
	Logical Data Component	<table border="1"> <tr> <td>Standards class</td> <td>Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard</td> </tr> <tr> <td>Standard creation date</td> <td>If the product is a standard when the standard was created.</td> </tr> <tr> <td>Last standard review date</td> <td>Last date that the standard was reviewed.</td> </tr> <tr> <td>Next standard review date</td> <td>Next date for the standard to be reviewed.</td> </tr> <tr> <td>Retire Date</td> <td>Date when the standard was/will be retired,</td> </tr> </table>	Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard	Standard creation date	If the product is a standard when the standard was created.	Last standard review date	Last date that the standard was reviewed.	Next standard review date	Next date for the standard to be reviewed.	Retire Date	Date when the standard was/will be retired,
Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard											
Standard creation date	If the product is a standard when the standard was created.											
Last standard review date	Last date that the standard was reviewed.											
Next standard review date	Next date for the standard to be reviewed.											
Retire Date	Date when the standard was/will be retired,											
Application Interface	N/A	This Object type has the default description attribute.										

Application Component	Logical Application Component		
	Physical Application Component		
		Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard
		Standard creation date	If the product is a standard when the standard was created.
		Last standard review date	Last date that the standard was reviewed.
		Next standard review date	Next date for the standard to be reviewed.
		Retire Date	Date when the standard was/will be retired,
		Lifecycle status	Proposed, In Development, Live, Phasing Out, Retired
		Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard
		Standard creation date	If the product is a standard, when the standard was created
		Last standard review date	Last date that the standard was reviewed
		Next standard review date	Next date for the standard to be reviewed
		Retire date	Date when the standard was/will be retired
		Initial live date	Date when the first release of the application was/will be released into production
		Date of last release	Date when the last release of the application was released into production
		Date of next release	Date when the next release of the application will be released into production
		Retirement date	Date when the application was/will be retired

		Availability characteristics	Degree to which something is available for use.
		Service times	Hours during which the application must be available
		Manageability characteristics	Ability to gather information about the state of something and control it
		Serviceability characteristics	Ability to identify problems and take corrective action, such as to repair or upgrade a component in a running system
		Performance characteristics	Ability of a component to perform its tasks in an appropriate time
		Reliability characteristics	Resistance to failure
		Recoverability characteristics	Ability to restore a system to a working state after an interruption
		Locatability characteristics	Ability of a system to be found when needed
		Security characteristics	Ability of a system to prevent unauthorized access to functions and data
		Privacy characteristics	Protection of data from unauthorized access
		Integrity characteristics	Ability of a system to ensure that data has not been corrupted
		Credibility characteristics	Ability of a system to ensure that the service request originates from an authorized source
		Localization characteristics	Ability of a service to support localized variants for different consumer groups
		Internationalization characteristics	Ability of a service to support international variations in business logic and data representation (such as character set)
		Interoperability characteristics	Ability of the service to interoperate with different technical environments, inside and outside of the organization

		Scalability characteristics	Ability of the service to grow or shrink its performance or capacity appropriately to the demands of the environment in which it operates
		Portability characteristics	Of data, people, applications, and components
		Extensibility characteristics	Ability to accept new functionality
		Capacity characteristics	Contracted capacity of the service provider to meet requests
		Throughput	Required throughput capacity
		Throughput period	Time period needed to deliver throughput capacity
		Growth	Expected future growth rate of service request
		Growth period	Time period needed to reach the expected growth rate
		Peak profile short term	Short-term profile of peak service traffic
		Peak profile long term	Long-term profile of peak service traffic
Application Collaboration	N/A	This Object type has the default description attribute.	
Application Function	Logical Application Component	Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard
		Standard creation date	If the product is a standard when the standard was created.
		Last standard review date	Last date that the standard was reviewed.
		Next standard review date	Next date for the standard to be reviewed.
		Retire Date	Date when the standard was/will be retired,

## Technology Domain Metamodel



ArchiMate Element	TOGAF Element	Attributes										
Infrastructure Service	Platform Service	<table border="1"> <tr> <td>Standard Class</td> <td>Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard.</td> </tr> <tr> <td>Category</td> <td>Platform Services are categorized according to the TOGAF TRM, which may be extended to meet the needs of an individual organization</td> </tr> </table>	Standard Class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard.	Category	Platform Services are categorized according to the TOGAF TRM, which may be extended to meet the needs of an individual organization						
Standard Class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard.											
Category	Platform Services are categorized according to the TOGAF TRM, which may be extended to meet the needs of an individual organization											
Infrastructure Function	Logical Technology Component	<table border="1"> <tr> <td>Standards class</td> <td>Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard</td> </tr> <tr> <td>Standard creation date</td> <td>If the product is a standard when the standard was created.</td> </tr> <tr> <td>Last standard review date</td> <td>Last date that the standard was reviewed.</td> </tr> <tr> <td>Next standard review date</td> <td>Next date for the standard to be reviewed.</td> </tr> <tr> <td>Retire Date</td> <td>Date when the standard was/will be retired,</td> </tr> </table>	Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard	Standard creation date	If the product is a standard when the standard was created.	Last standard review date	Last date that the standard was reviewed.	Next standard review date	Next date for the standard to be reviewed.	Retire Date	Date when the standard was/will be retired,
Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard											
Standard creation date	If the product is a standard when the standard was created.											
Last standard review date	Last date that the standard was reviewed.											
Next standard review date	Next date for the standard to be reviewed.											
Retire Date	Date when the standard was/will be retired,											

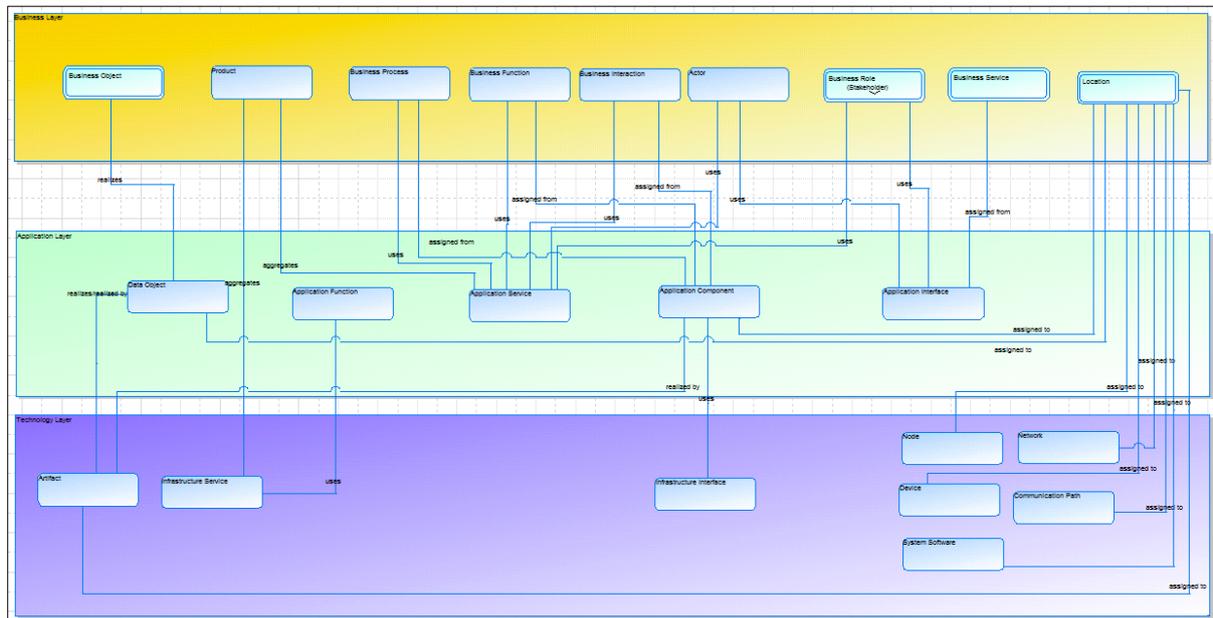
Artifact	Physical Data Component	Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard
		Standard creation date	If the product is a standard when the standard was created.
		Last standard review date	Last date that the standard was reviewed.
		Next standard review date	Next date for the standard to be reviewed.
		Retire Date	Date when the standard was/will be retired,
System Software	Physical Technology Component	Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard
		Standard creation date	If the product is a standard when the standard was created.
		Last standard review date	Last date that the standard was reviewed.
		Next standard review date	Next date for the standard to be reviewed.
		Retire Date	Date when the standard was/will be retired,
		Category	Physical Technology Components are categorized according to the TOGAF TRM, which may be extended to meet the needs of an individual organization.
		Product Name	Name of the product making up the technology component.
		Module Name	Module, or other sub-product, name making up the technology component.
		Vendor	Vendor providing the technology component.
		Version	Version of the product making up the technology component

Device	Physical Technology Component	Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard
		Standard creation date	If the product is a standard when the standard was created.
		Last standard review date	Last date that the standard was reviewed.
		Next standard review date	Next date for the standard to be reviewed.
		Retire Date	Date when the standard was/will be retired,
Node	Logical Technology Component	Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard
		Standard creation date	If the product is a standard when the standard was created.
		Last standard review date	Last date that the standard was reviewed.
		Next standard review date	Next date for the standard to be reviewed.
		Retire Date	Date when the standard was/will be retired,
Infrastructure Interface	Logical Technology Component	Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard
		Standard creation date	If the product is a standard when the standard was created.
		Last standard review date	Last date that the standard was reviewed.
		Next standard review date	Next date for the standard to be reviewed.
		Retire Date	Date when the standard was/will be retired,

	Physical Technology Component	Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard
		Standard creation date	If the product is a standard when the standard was created.
		Last standard review date	Last date that the standard was reviewed.
		Next standard review date	Next date for the standard to be reviewed.
		Retire Date	Date when the standard was/will be retired,
		Category	Physical Technology Components are categorized according to the TOGAF TRM, which may be extended to meet the needs of an individual organization.
		Product Name	Name of the product making up the technology component.
		Module Name	Module, or other sub-product, name making up the technology component.
		Vendor	Vendor providing the technology component.
		Version	Version of the product making up the technology component

Network	Physical Technology Component	<table border="1"> <tr> <td data-bbox="644 309 831 450">Standards class</td> <td data-bbox="831 309 1385 450">Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard</td> </tr> <tr> <td data-bbox="644 450 831 544">Standard creation date</td> <td data-bbox="831 450 1385 544">If the product is a standard when the standard was created.</td> </tr> <tr> <td data-bbox="644 544 831 638">Last standard review date</td> <td data-bbox="831 544 1385 638">Last date that the standard was reviewed.</td> </tr> <tr> <td data-bbox="644 638 831 732">Next standard review date</td> <td data-bbox="831 638 1385 732">Next date for the standard to be reviewed.</td> </tr> <tr> <td data-bbox="644 732 831 792">Retire Date</td> <td data-bbox="831 732 1385 792">Date when the standard was/will be retired,</td> </tr> <tr> <td data-bbox="644 792 831 974">Category</td> <td data-bbox="831 792 1385 974">Physical Technology Components are categorized according to the TOGAF TRM, which may be extended to meet the needs of an individual organization.</td> </tr> <tr> <td data-bbox="644 974 831 1068">Product Name</td> <td data-bbox="831 974 1385 1068">Name of the product making up the technology component.</td> </tr> <tr> <td data-bbox="644 1068 831 1162">Module Name</td> <td data-bbox="831 1068 1385 1162">Module, or other sub-product, name making up the technology component.</td> </tr> <tr> <td data-bbox="644 1162 831 1223">Vendor</td> <td data-bbox="831 1162 1385 1223">Vendor providing the technology component.</td> </tr> <tr> <td data-bbox="644 1223 831 1317">Version</td> <td data-bbox="831 1223 1385 1317">Version of the product making up the technology component</td> </tr> </table>	Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard	Standard creation date	If the product is a standard when the standard was created.	Last standard review date	Last date that the standard was reviewed.	Next standard review date	Next date for the standard to be reviewed.	Retire Date	Date when the standard was/will be retired,	Category	Physical Technology Components are categorized according to the TOGAF TRM, which may be extended to meet the needs of an individual organization.	Product Name	Name of the product making up the technology component.	Module Name	Module, or other sub-product, name making up the technology component.	Vendor	Vendor providing the technology component.	Version	Version of the product making up the technology component
Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard																					
Standard creation date	If the product is a standard when the standard was created.																					
Last standard review date	Last date that the standard was reviewed.																					
Next standard review date	Next date for the standard to be reviewed.																					
Retire Date	Date when the standard was/will be retired,																					
Category	Physical Technology Components are categorized according to the TOGAF TRM, which may be extended to meet the needs of an individual organization.																					
Product Name	Name of the product making up the technology component.																					
Module Name	Module, or other sub-product, name making up the technology component.																					
Vendor	Vendor providing the technology component.																					
Version	Version of the product making up the technology component																					
Communication Path	Logical Technology Component	<table border="1"> <tr> <td data-bbox="644 1402 831 1543">Standards class</td> <td data-bbox="831 1402 1385 1543">Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard</td> </tr> <tr> <td data-bbox="644 1543 831 1637">Standard creation date</td> <td data-bbox="831 1543 1385 1637">If the product is a standard when the standard was created.</td> </tr> <tr> <td data-bbox="644 1637 831 1731">Last standard review date</td> <td data-bbox="831 1637 1385 1731">Last date that the standard was reviewed.</td> </tr> <tr> <td data-bbox="644 1731 831 1825">Next standard review date</td> <td data-bbox="831 1731 1385 1825">Next date for the standard to be reviewed.</td> </tr> <tr> <td data-bbox="644 1825 831 1886">Retire Date</td> <td data-bbox="831 1825 1385 1886">Date when the standard was/will be retired,</td> </tr> </table>	Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard	Standard creation date	If the product is a standard when the standard was created.	Last standard review date	Last date that the standard was reviewed.	Next standard review date	Next date for the standard to be reviewed.	Retire Date	Date when the standard was/will be retired,										
Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard																					
Standard creation date	If the product is a standard when the standard was created.																					
Last standard review date	Last date that the standard was reviewed.																					
Next standard review date	Next date for the standard to be reviewed.																					
Retire Date	Date when the standard was/will be retired,																					

## Cross Layer Metamodel



## Implementation and Migration Extension Metamodel

Plateau	N/A	This object type has the default description attribute.				
Gap	Gap	This object type has the default description attribute.				
Deliverable	N/A	This object type has the default description attribute.				
Work Package	Work Package	<table border="1"> <tr> <td>Category</td> <td>The following categories of work package apply: Work Package, Work Stream, Project, Program, Portfolio.</td> </tr> <tr> <td>Capability delivered</td> <td>Describes the contribution this work package makes to capability delivery.</td> </tr> </table>	Category	The following categories of work package apply: Work Package, Work Stream, Project, Program, Portfolio.	Capability delivered	Describes the contribution this work package makes to capability delivery.
Category	The following categories of work package apply: Work Package, Work Stream, Project, Program, Portfolio.					
Capability delivered	Describes the contribution this work package makes to capability delivery.					

## Motivation Extension Metamodel

Stakeholder (specialism of Role)	Role	<table border="1"> <tr> <td># FTEs</td> <td>Estimated number of FTEs that operate as this Role</td> </tr> </table>	# FTEs	Estimated number of FTEs that operate as this Role										
# FTEs	Estimated number of FTEs that operate as this Role													
	Organization Unit	<table border="1"> <tr> <td>Headcount</td> <td>Number of FTEs working within the organization. (Note that this attribute is merged with # FTEs)</td> </tr> </table>	Headcount	Number of FTEs working within the organization. (Note that this attribute is merged with # FTEs)										
Headcount	Number of FTEs working within the organization. (Note that this attribute is merged with # FTEs)													
Driver	Driver	This object type has the default description attribute.												
Assessment	Measure	This object type has the default description attribute.												
Goal	Goal	This object type has the default description attribute.												
	Objective	This object type has the default description attribute.												
Principle	Principle	<table border="1"> <tr> <td>Category</td> <td>The following categories of principle apply: Guiding Principle, Business Principle, Data Principle, Application Principle, Integration Principle, Technology Principle.</td> </tr> <tr> <td>Priority</td> <td>Priority of this principle relative to other principles</td> </tr> <tr> <td>Statement of Principle</td> <td>Statement of what the principle is</td> </tr> <tr> <td>Rationale</td> <td>Statement of why the principle is required and the outcome to be reached</td> </tr> <tr> <td>Implication</td> <td>Statement of what the principle means in practical terms</td> </tr> <tr> <td>Metric</td> <td>Identifies mechanisms that will be used to measure whether the principle has been met or not</td> </tr> </table>	Category	The following categories of principle apply: Guiding Principle, Business Principle, Data Principle, Application Principle, Integration Principle, Technology Principle.	Priority	Priority of this principle relative to other principles	Statement of Principle	Statement of what the principle is	Rationale	Statement of why the principle is required and the outcome to be reached	Implication	Statement of what the principle means in practical terms	Metric	Identifies mechanisms that will be used to measure whether the principle has been met or not
Category	The following categories of principle apply: Guiding Principle, Business Principle, Data Principle, Application Principle, Integration Principle, Technology Principle.													
Priority	Priority of this principle relative to other principles													
Statement of Principle	Statement of what the principle is													
Rationale	Statement of why the principle is required and the outcome to be reached													
Implication	Statement of what the principle means in practical terms													
Metric	Identifies mechanisms that will be used to measure whether the principle has been met or not													

Requirement	Requirement	Statement of requirement	Statement of what the requirement is, including a definition of whether the requirement shall be met, should be met, or may be met
		Rationale	Statement of why the requirement exists
		Acceptance criteria	Statement of what tests will be carried out to ensure that the requirement will be met
Constraint	Constraint	This object type has the default description attribute.	

### Capability Based Planning

Business Capability	Capability	Business Value	Describes how this capability provides value to the enterprise
		Increments	Lists possible maturity/quality levels for the capability
Capability Increment	N/A	This object type has the default description attribute.	
Metric	N/A	This object type has the default description attribute.	

# Enterprise architecture notation

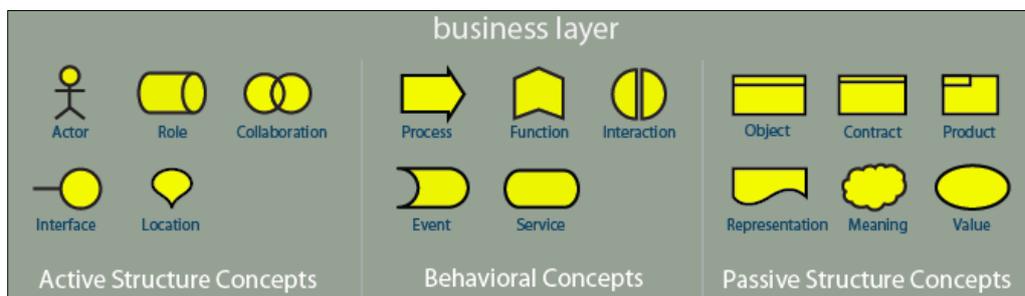
**Package:** Agile Enterprise Architecture

**Role Applicable to:** Client Administrator, User, Contributor, Reviewer

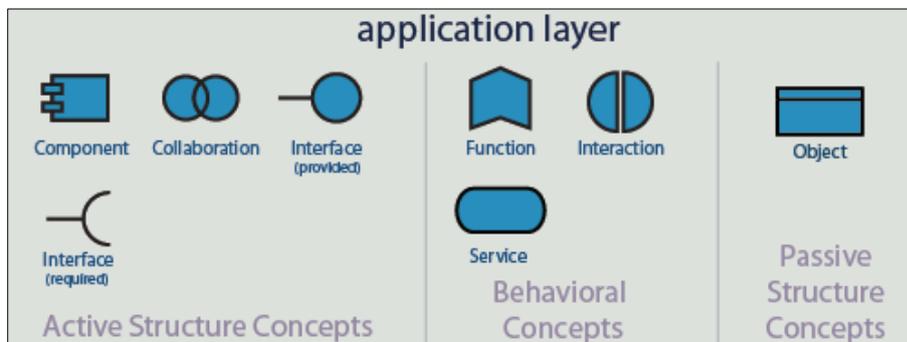
The agile enterprise architecture notation uses the standard ArchiMate 2.1 notation.

Corso supply a [poster](#) of the ArchiMate notation that you may find useful.

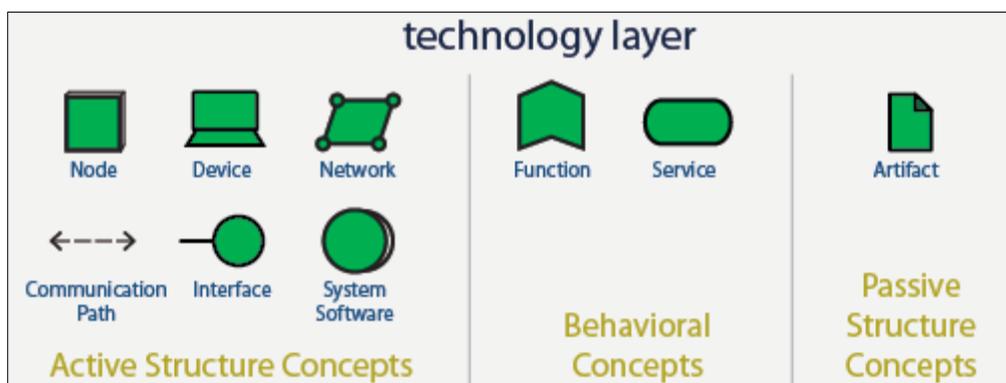
## Business Layer



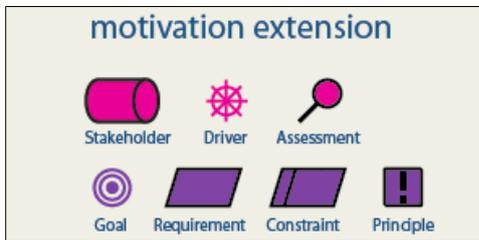
## Application Layer



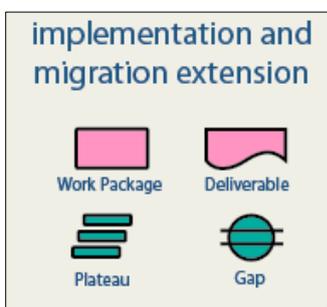
## Technology Layer



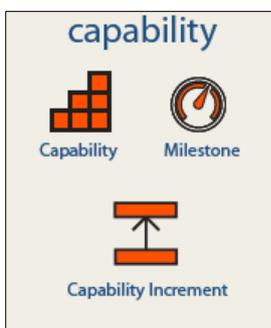
## Motivation Extension



## Implementation and Migration Extension



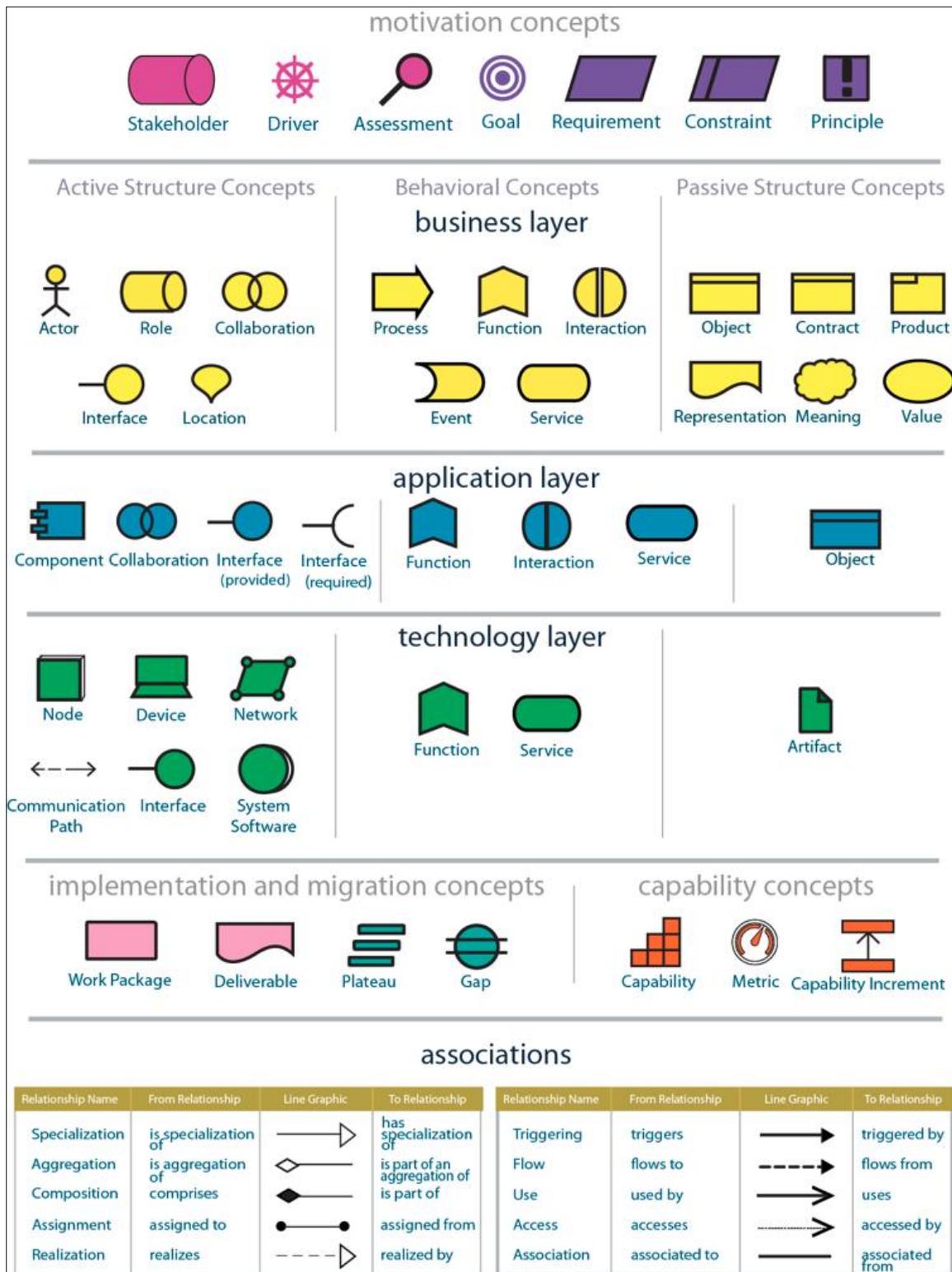
## Capability Extension



## Relationship types

ArchiMate relationships							
Relationship Name	From Relationship	Line Graphic	To Relationship	Relationship Name	From Relationship	Line Graphic	To Relationship
Specialization	is specialization of		has specialization of	Triggering	triggers		triggered by
Aggregation	is aggregation of		is part of an aggregation of	Flow	flows to		flows from
Composition	comprises		is part of	Use	used by		uses
Assignment	assigned to		assigned from	Access	accesses		accessed by
Realization	realizes		realized by	Association	associated to		associated from

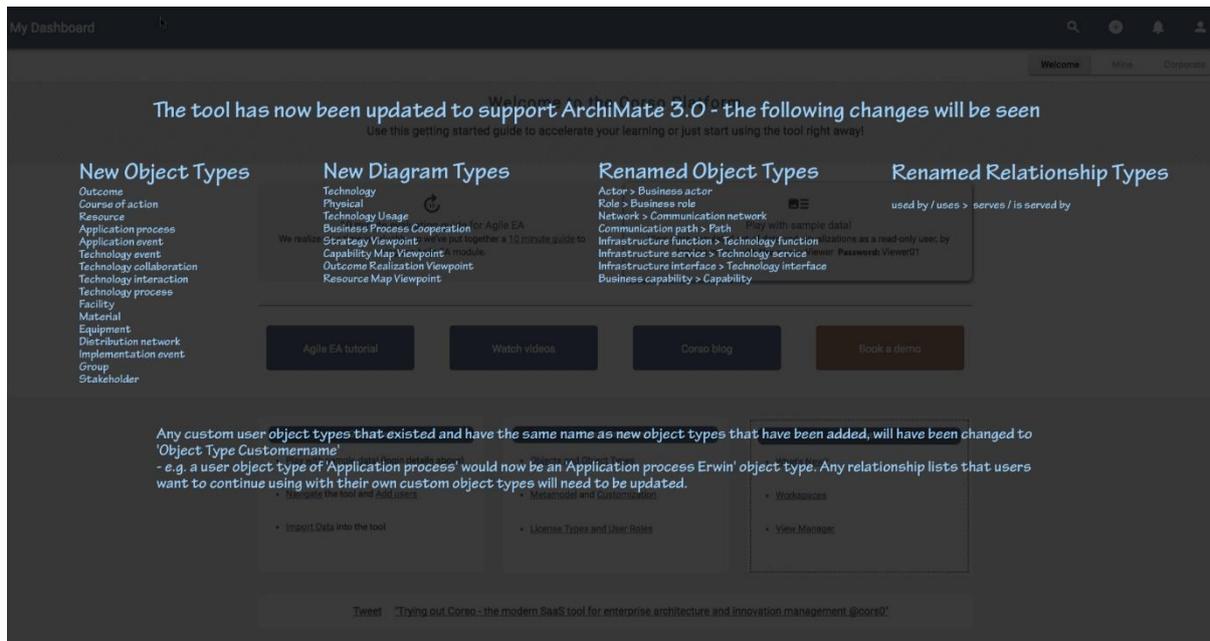
# ArchiMate Concepts



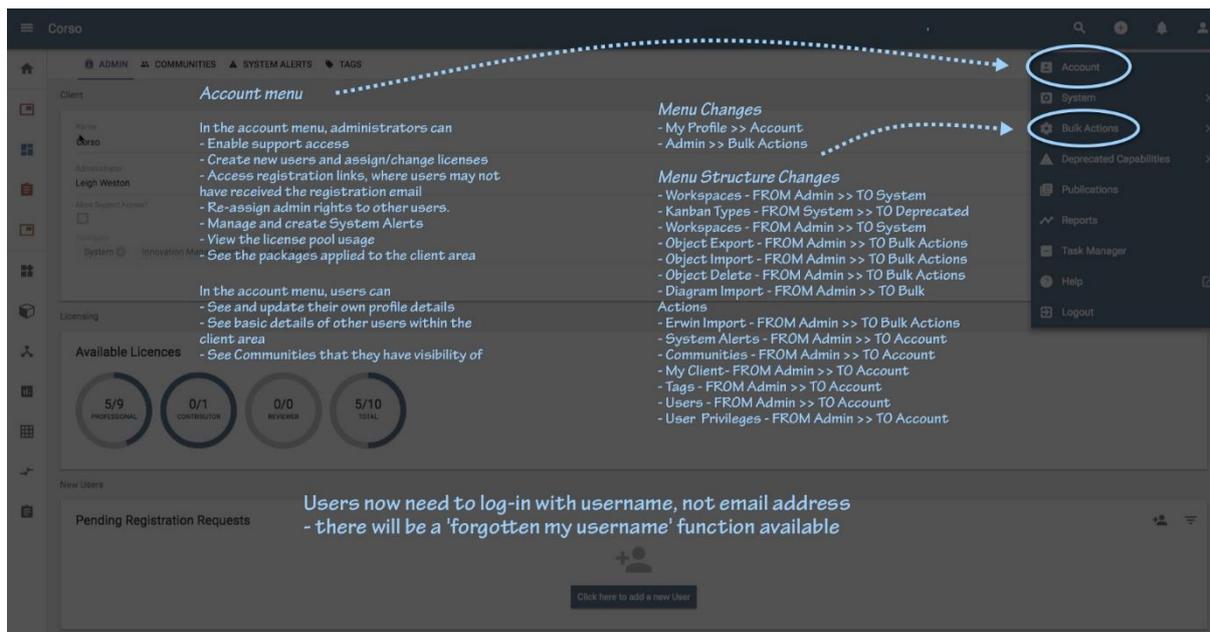
# Tool Change Overlays

In this page, you can find the in-tool overlays that are shown when significant changes are made to the tool - Just in case you clicked and dismissed them before you had chance to read and take in the information

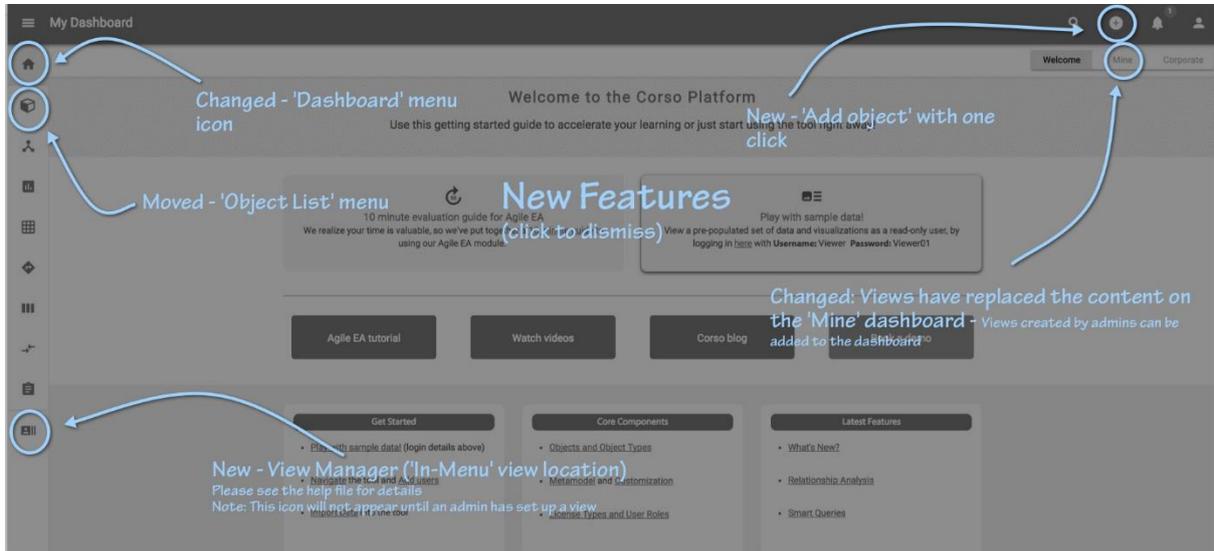
## ArchiMate 3



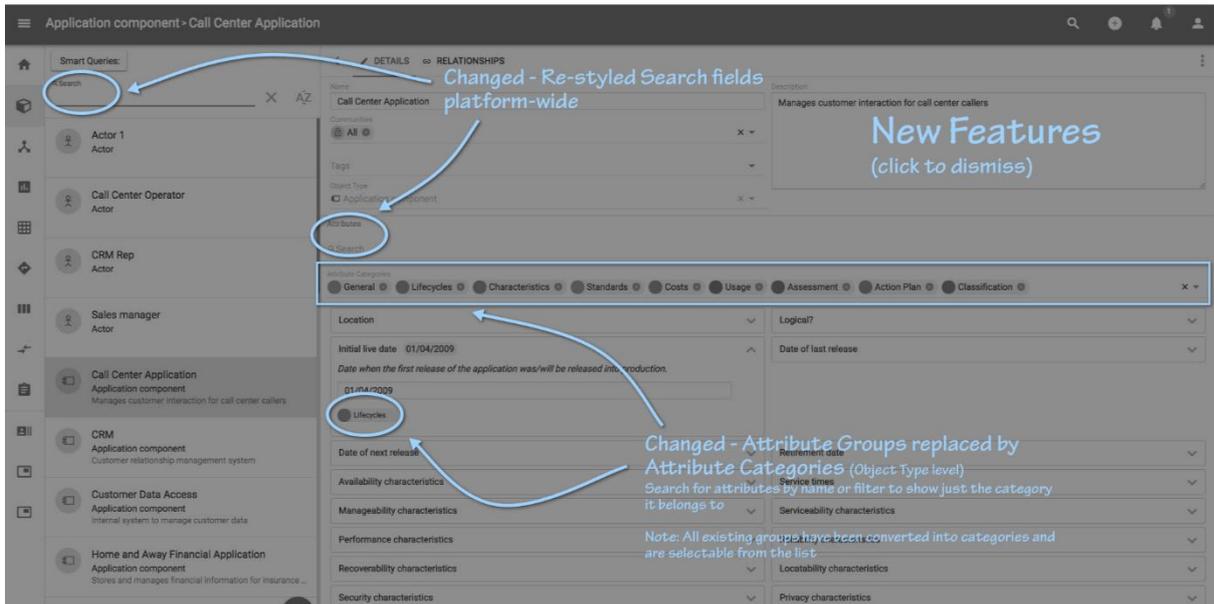
## Menu changes and Account menu



## Menu changes and dashboard



## Object UI changes



## View Manager and Relationship UI Changes

Current > Views > Claim Management Applications

CLAIM MANAGEMENT APPLICATION PORTFOLIO LIFECYCLES LIVE ROADMAP RETIREMENT ROADMAP ACTION PLAN DATA DISTRIBUTION

Claim Management Applications

High Risk Applications

Underperforming Capabilities

Infrastructure with security implication

Workpackage Approvals

Claim Application Assessment

Critical Processes

Capability Cost Savings

CONFIGURE

Smart Queries

Claim Management Application Portfolio

Name	Standards class	Related entities
Bank Application	Proposed Standard	
Bank System	Standard	
Call Center Manager	Phasing Out Standard	
Claim Data Management	Retired Standard	Claim
Claim Handling	Standard	Claim

Changed - View Configuration

- Views are now configurable from within either the 'Views' menu item, or from within their own bookmarked menu item
- All User types can now see and play with View configurations, but only 'Admins' and 'Users' can save them

Changed - 'View Manager' moved into left side menu, as 'Views'

Changed - Relationships UI

UI changed for:

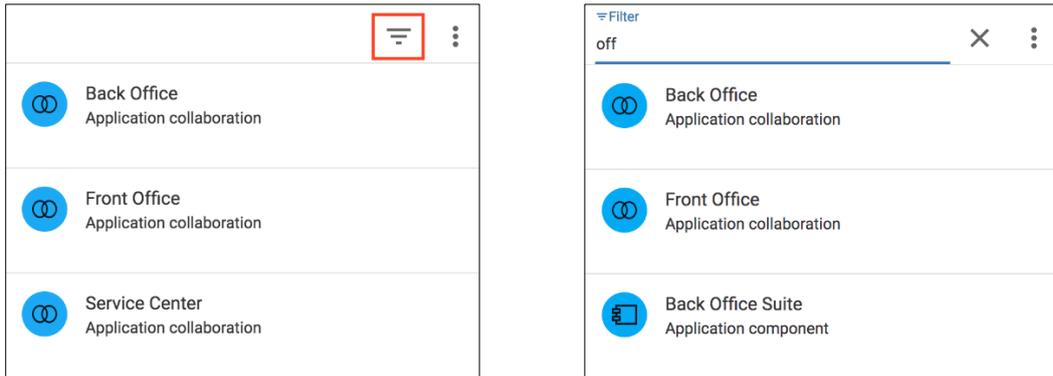
- Object Relationships
- Object Type Relationships
- Diagram/drawing relationships

Simplified what is offered to the user

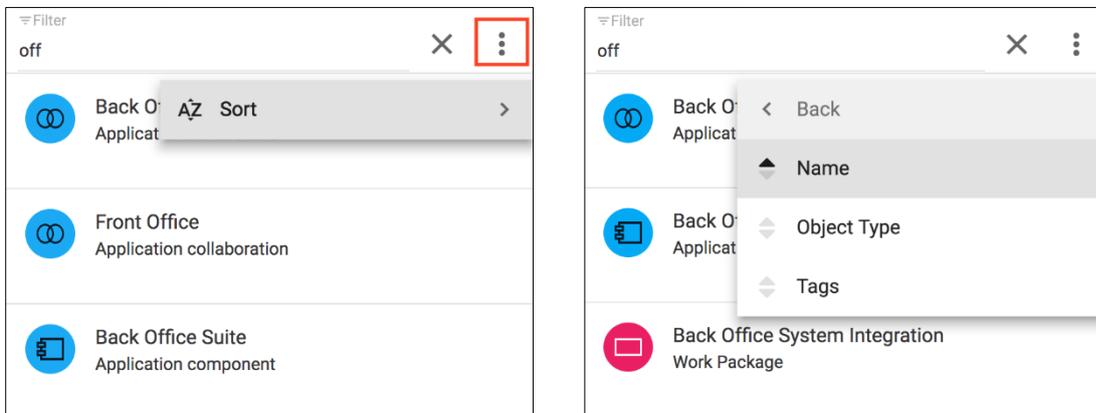
# In-Module Search

**Role Applicable to:** Client Administrator, User, Contributor, Reviewer

Each module in the platform provides 'type ahead' capability. This means that as you type, the table in the module is filtered by whatever is typed into the Filter.



The list of objects, filtered or not, can be alphabetically sorted. A combination of object Name, Object Type name, and Tag names, can be used when sorting. Each sort can be in ascending or descending order.



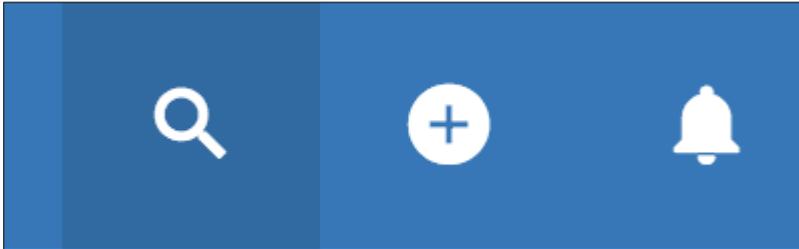
**i** Type ahead capability is prevalent throughout the platform.

---

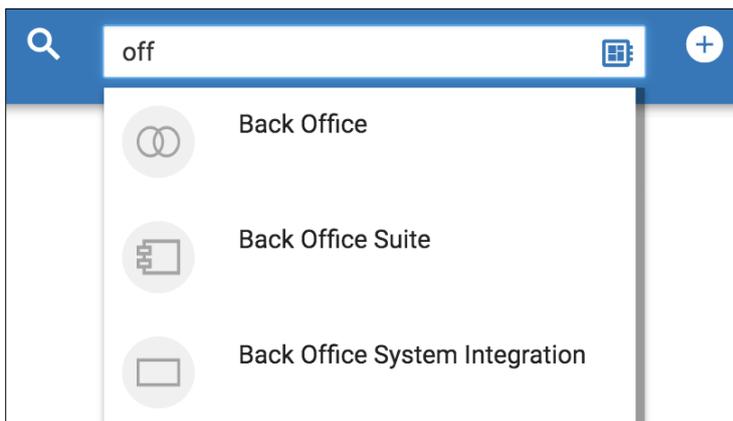
# Global Search

**Role Applicable to:** Client Administrator, User, Contributor, Reviewer

Users can search for all platform contents (objects or visualizations), using the type-ahead capability, within the search box in the top toolbar of the tool, click the magnifying glass to expand the search.

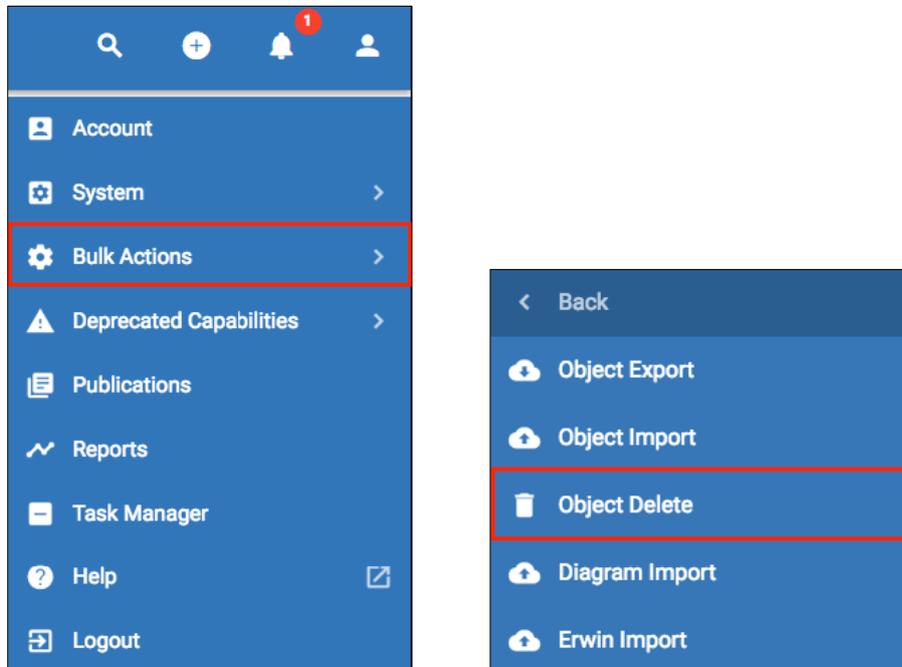


The results are shown in a list below the global search box - selecting the option from the list, will open the object or visualization.

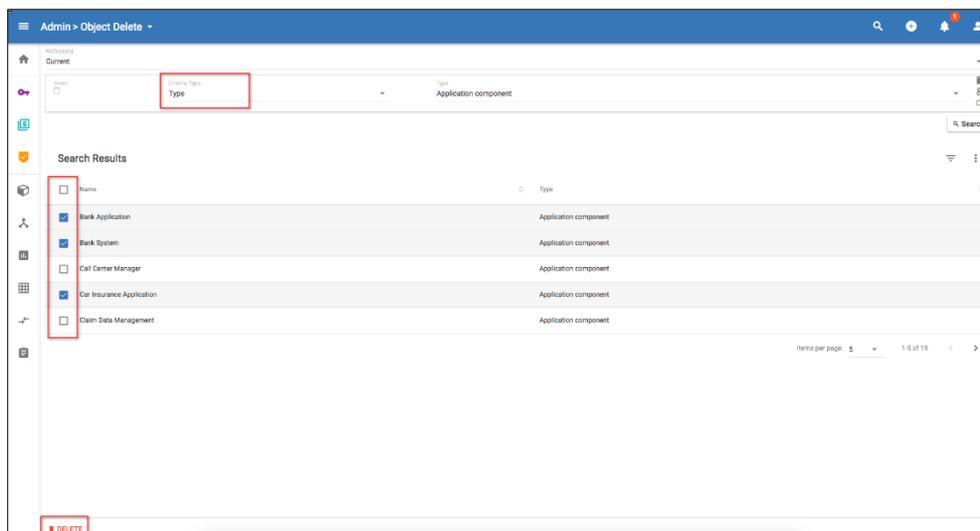


# Object Deletion

Deleting a selection of objects can now be performed in the **User Menu > Bulk Actions > Object Delete**



Users can search for a subset of objects, using criteria to build queries, and then select objects for deletion from the results.



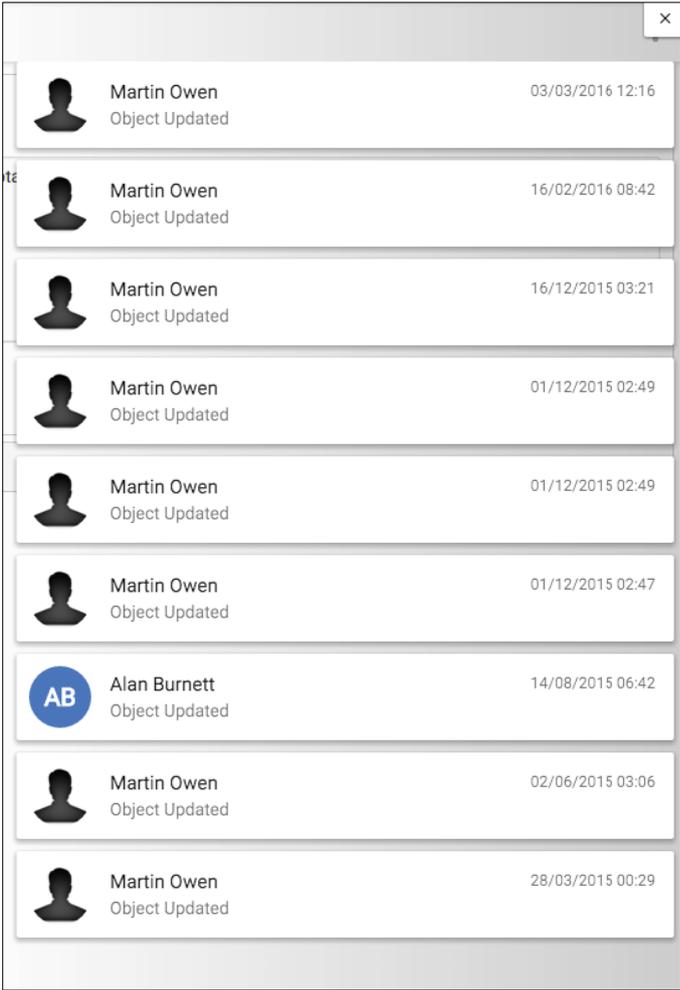
---

# Activity log

**Role Applicable to:** [Client Administrator](#), [User](#), [Contributor](#)

The activity log includes a list of changes made to the object including details of who made the changes and when.

The activity log is available by clicking on an object and navigating to activity using the options menu (the 3 vertical dots).



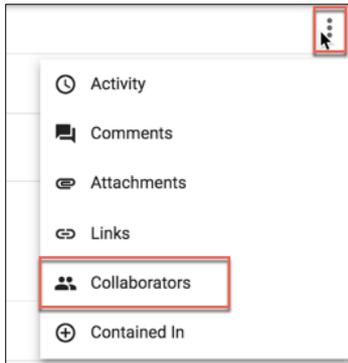
User	Action	Timestamp
Martin Owen	Object Updated	03/03/2016 12:16
Martin Owen	Object Updated	16/02/2016 08:42
Martin Owen	Object Updated	16/12/2015 03:21
Martin Owen	Object Updated	01/12/2015 02:49
Martin Owen	Object Updated	01/12/2015 02:49
Martin Owen	Object Updated	01/12/2015 02:47
Alan Burnett	Object Updated	14/08/2015 06:42
Martin Owen	Object Updated	02/06/2015 03:06
Martin Owen	Object Updated	28/03/2015 00:29

---

# Task management

**Role Applicable to:** [Client Administrator](#), [Professional](#), [Contributor](#), [Reviewer](#)

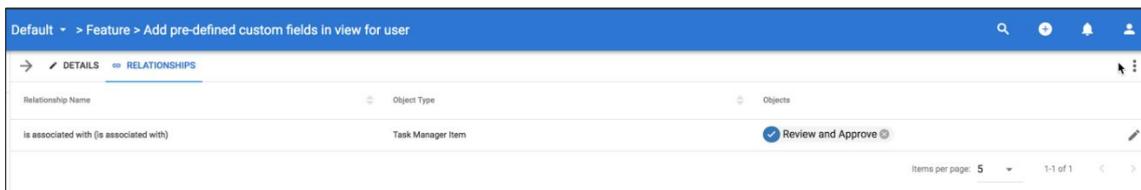
Tasks can be created for users and managed within a kanban board. Task items are created when users are added as Collaborators within an object



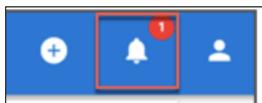
Fill in the task details and save

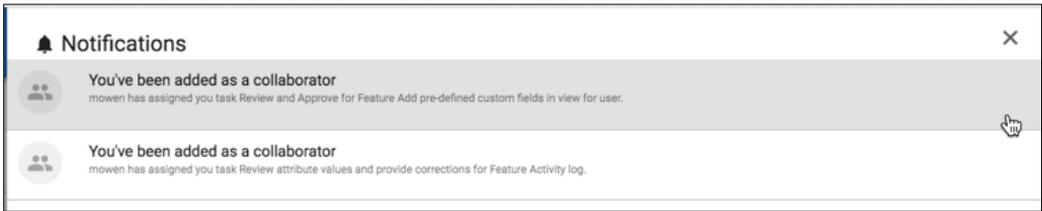
A screenshot of a task details form. The 'Task Name' field contains 'Review attribute values and provide corrections'. The 'Target Date' field contains '05/19/2017 10:19 AM'. The 'Collaborators' field contains 'Leigh Weston'. At the bottom, there are 'DELETE' and 'SAVE' buttons.

When a user is added as a collaborator, an object of type 'Task Manager Item' is created and linked to the source object item, with an 'associated with' relationship



The user will receive an email and a notification in-tool, telling them who added them, what the task is and what object it relates too.



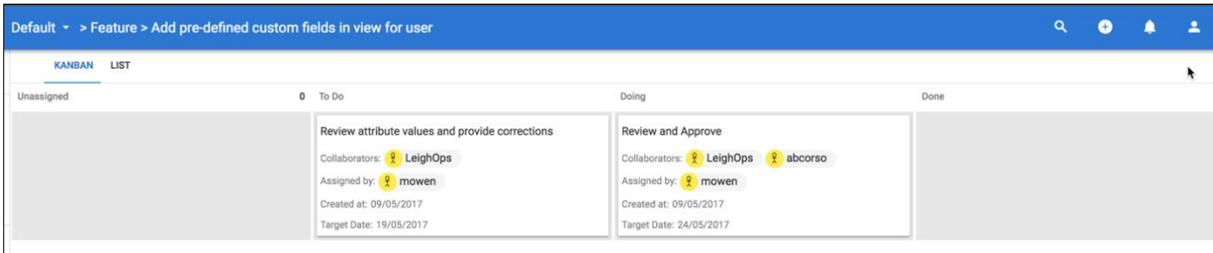


Clicking the notification will take you to the source object of the task, within the same browser

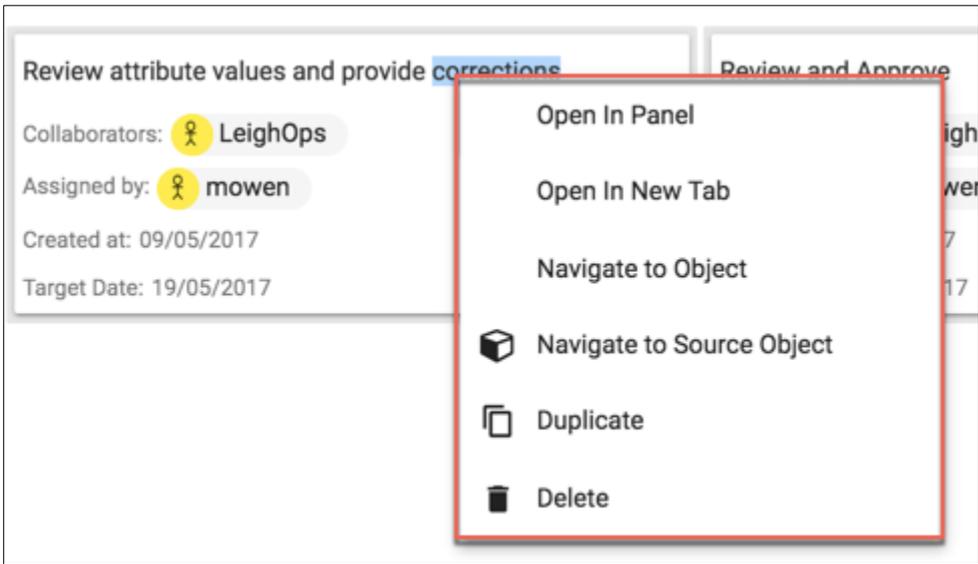
The Task Manager allows users to manage all of their own 'Task Manager Items', and can be accessed from the **User** menu > **Task Manager**.

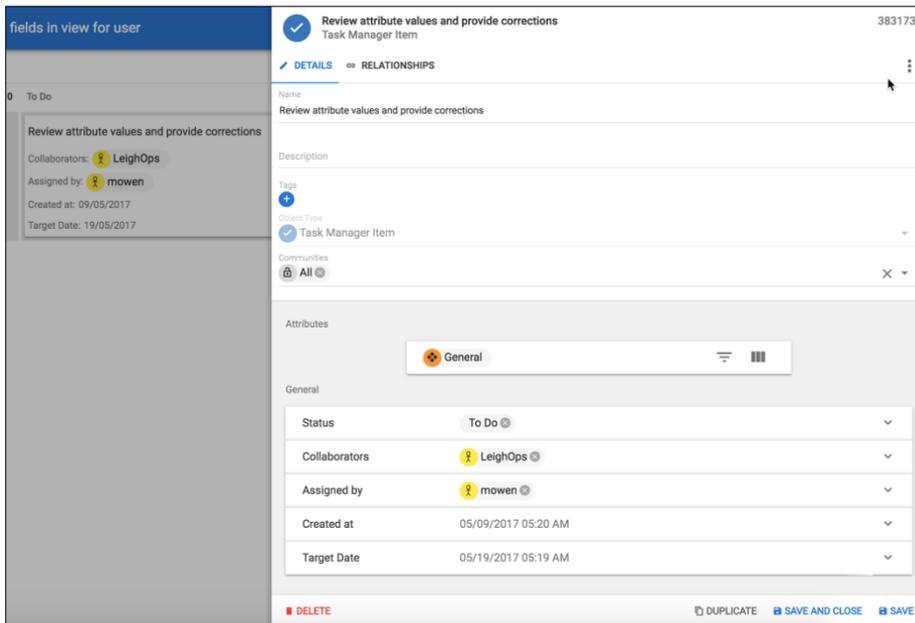
Task cards will show, presenting the assigned collaborators, who assigned the task, when the task was created and what the target date is.

Dragging the tasks into separate stages will update the status attribute.



Right clicking in the kanban will allow users to open the task, or open the linked object for more context and to complete the requested task





Task Manager Item objects can be added to an overall kanban views, if team leaders want to see all tasks their team members are working on.

- Reviewers can move the cards on the kanban stage - to update the status of the tasks they are working on.
- They cannot edit the attributes of a task manager item

---

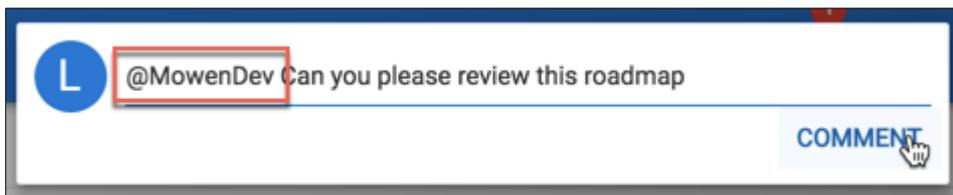
# Mentions

The platform provides a mechanism for 'mentions'. You may be familiar with mentions in other platforms such as facebook.

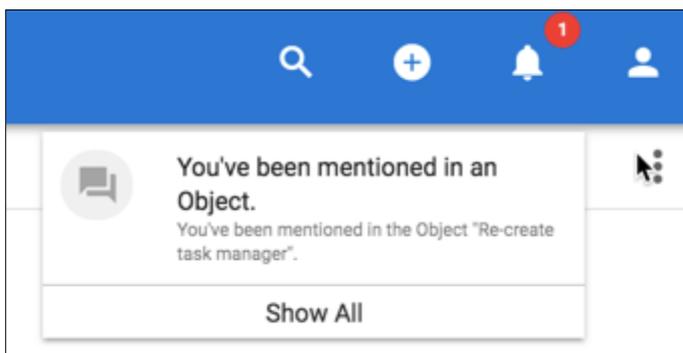
Mentioning uses the @ symbol within Comments on Objects, Diagrams and Views.

When you are in the comments section, you can use the @ symbol to bring up a list of users on the platform.

Selecting the user will place their username after the @ and you can continue with the message.



This is useful as it alerts and sends the user an email message that they were mentioned in a page, as well as showing an in-tool notification.



Clicking the notification will take the user directly to the object they have been mentioned on.

## Domains

The following domains are available on the erwin platform:

### [myerwin.io](#)

This is the live environment in the US. Both live and evaluation customers run in this environment.

There is no sample data within it.

### [uk.myerwin.io](#)

This is the live environment in the UK. Both live and evaluation customers run in this environment.

There is no sample data within it.

### [beta.myerwin.io](#)

This environment is for beta customers only. It is a formal environment for customers to test and provide feedback on our software.

It may have sample data within it.

Customers should not place production data within this.

We will place new capabilities, tools and features into this database for customers to play with on a request basis.

---

# Platforms

The strategic planning platform is designed to work on mobile platforms and on the desktop.

## Browsers

The following browsers have been tested as working. However, other versions of browser may work but results may be unpredictable.

- Chrome 30 and upwards (Recommended)
- Safari 6.0 and upwards
- IE 11 and above

## Mobile

The following platforms have been tested as working:

- iPad
- iPad mini

---

# Installation

The erwin Data Foundation Platform is normally licensed on a SaaS platform. However, for on-premise installations, please contact [erwin](#).

# User Roles

## User Role/Capability Matrix

Capability	License/Role			
		License - Professional	License - Contributor	License - Reviewer
	Role - Administrator	Role - Professional	Role - Contributor	Role - Reviewer
Account	CRUD	RU	RU	RU
Dashboard (Personal)	The admin role is something that can be applied to any user in the client that is using any license type. It gives further capabilities in the Account section to create users and modify client details.	CRUD	CRUD	CRUD
Object		CRUD	CRUD	CR
Publication		CRUD	R	R
Task Manager		CRUD	CRUD	CRUD*
Pivot Table		CRUD	R	R
Chart		CRUD	R	R
Campaign		CRUD	R	R
Community	CRUD	RU	R	R
Comparison		CRUD	R	R

---

Diagrams		CRUD	CRUD	R
Diagrams Analysis		CRUD	R	R
Scorecards		CRUD	R	R
Roadmap		CRUD	R	R
Kanban		CRUD	R	R
Smart Queries		CRUD	CRUD	R
Object Types		CRUD		
Views		CRUD	R	R
Workspaces		CRUD	RU	R

\* = Cannot add collaborators

C = Create, R = Read, U = Update, D = Delete

An end user may be one of four roles in the platform. We categorize the roles by the license types required to activate them.

## **Licenses**

### **Administrator**

The administrator is responsible for

- Managing the users of the repository and licensing of those users.
- Configuration of the communities.

### **Professional**

The professional user is responsible for

- Managing the the repository structure including projects and workspaces.
- The meta-model extensions/customizations.

---

The aim of a professional license is to allow stakeholders to perform a complete set of end user tasks and full analysis on objects.

### **Contributor Licenses**

The aim of a contributor license is to allow stakeholders to create and contribute concepts onto the platform. The contributor is primarily a front end user of the platform.

A Contributor is a user who can, create objects, see data about objects that they have been given permission to see, and add comments. A contributor may also create diagrams and publications. A contributor may participate in comparisons. It is a perfect way to increase visibility across the organization and engage CXOs, IT, engineering and other key teams. A contributor will need a contributor license.

### **Reviewer Licenses**

The aim of a reviewer license is to provide the ability for stakeholders to review and comment upon concepts on the platform.

A Reviewer user does not have entire Create, Delete or Update access to the repository. A reviewer can configure their own dashboard, profile settings and create their own tasks and review and comment on other users tasks. An administrator may create as many reviewers as they wish according to their commercial license agreement.

---

# User Management

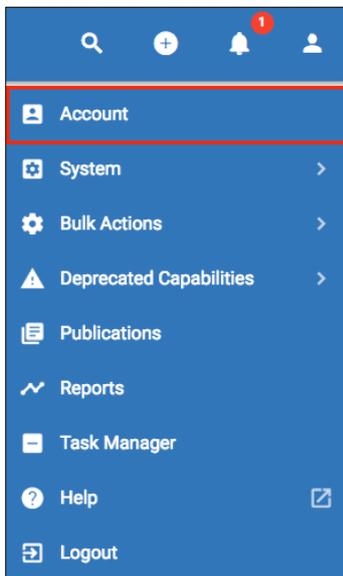
## Change the assigned Administrator

**Role needed to use:** [Client Administrator](#)

The administrator function gives the user the ability to create users and assign/remove licenses from users.

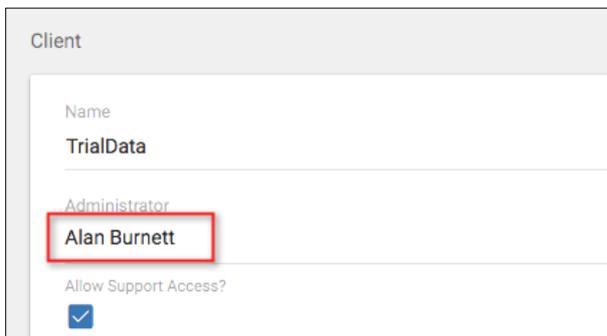
Administrators can transfer this function to another user.

To do this, whilst logged in as an Administrator, select the **User** menu > **Account**.



In the **Client** section, you will see an **Administrator** field, select the drop down list and choose the user you would like to have the Administrator function. They can be of any license type.

Click **Save**.



You, as a user will now no longer be able to see these fields, but the new designated administrator will be able to see them when they log in. They can then manage users.

---

## User Creation (for Administrators)

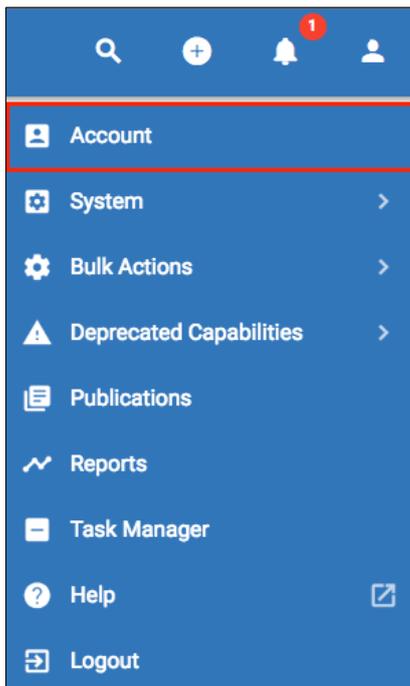
Client Administrators will be created by erwin and assigned the purchased license types within the tool by erwin. Client Administrators can then assign license for other users within their organization.

Users that sign up via the website for a trial will be given access to the environment. This will provide users with 3 User licenses, 3 Contributor licenses and 3 Reviewer licenses for the trial period. The administrator will consume the first User license by default. Client Administrators can then assign license for other users within their organization.

### ***Steps to create further users are below;***

For a list of the types of user and their capabilities click [here](#).

1. Login to the environment.
2. To create new users within the tool, we simply need to assign them a license from the license pool; select the **User** menu > **Account**.



3. Scroll down to the **Registration New Users** section and press the **+** button in the top right.



4. In the window that appears enter the email address of the new user, then press **SAVE**.

Email Address

CANCEL SAVE

- They will then appear in the register new users box. Users will receive an email notifying them that they have been assigned a license, and asking them to register and create a username and password. Once the registration has been completed, they will be re-directed to the login page and will appear within the platform. You can also click the link symbol at the right hand side of their row to follow to the registration link for them.

Email Address	Created
akelly@erwin.com	04/09/2017 12:20 PM

- Then expand the application that you wish to add them into, depending on your licenses, you may only see once application there.

Enterprise Architecture Agile	[Available Licences: 2/4]
Data Modeler NoSQL	[Available Licences: 0/1]

- On the license row, click in the user field and choose the user from the list that appears. The users will only appear as email addresses, once they have completed registration they will display as usernames.

License Type	Total	Expiration	Created	Last Login
Reviewer	1	31/05/2026 08:53 AM	05/05/2017 08:54 AM	
Contributor	0	31/05/2026 08:53 AM	05/05/2017 08:54 AM	05/06/2017 01:25 PM
Professional	1	31/05/2026 08:53 AM	05/05/2017 08:54 AM	04/09/2017 12:22 PM
Professional	2	31/05/2026 08:53 AM	05/05/2017 08:54 AM	

Registered users appear lower down on the page below your profile with a star signifying the client admin.

**Aidan Contributor**  
AidanContributor  
akelly@erwin.com  
05/06/2017 01:35 PM

---

**EA Agile: Contributor**  
**DM NoSQL: No Licence**

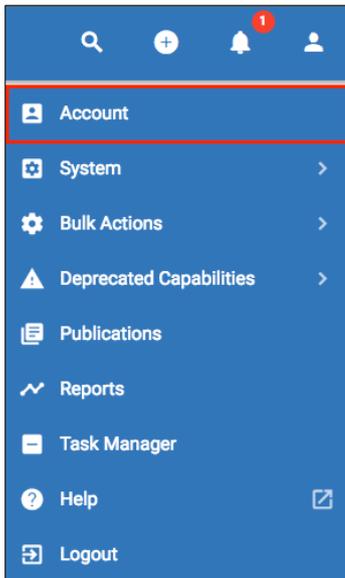
**Aidan Kelly** ★  
AidanTest  
akelly@erwin.com  
04/09/2017 12:22 PM

---

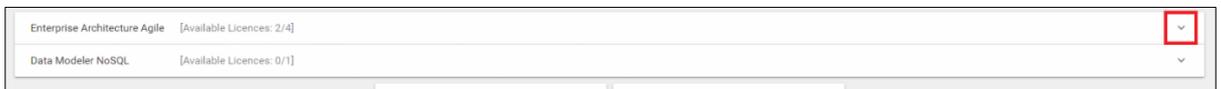
**EA Agile: Professional**  
**DM NoSQL: Professional**

# Change/Assign a license to an existing platform user (for Administrators)

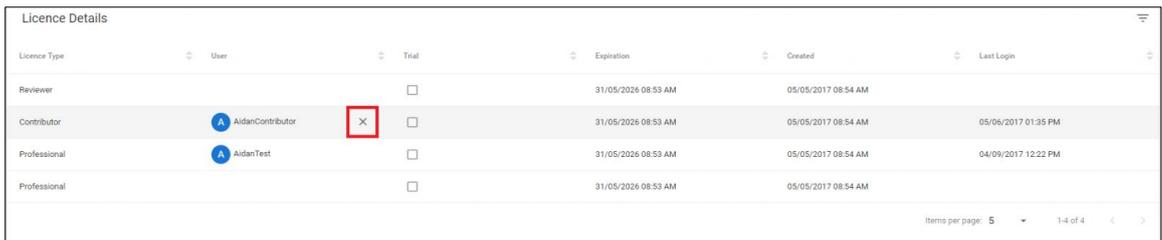
1. To change or assign a license to existing users within the tool. Select the **User** menu > **Account**.



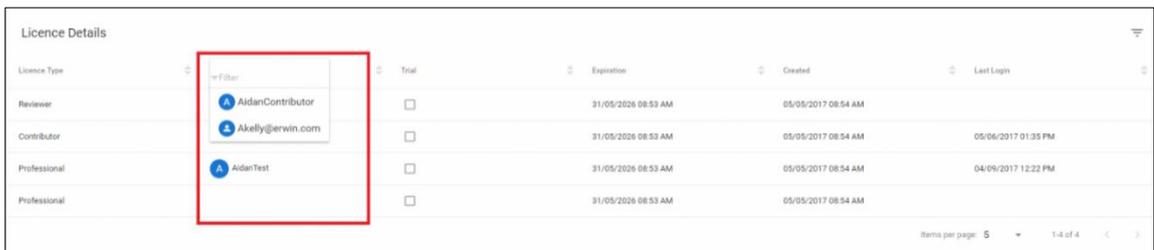
2. Then expand the application that you wish to add them into/modify their licenses for, depending on your licenses, you may only see once application there.



3. Find the license they are currently assigned to and hover over their username, you will an 'X' appear, click this to remove them from the license.

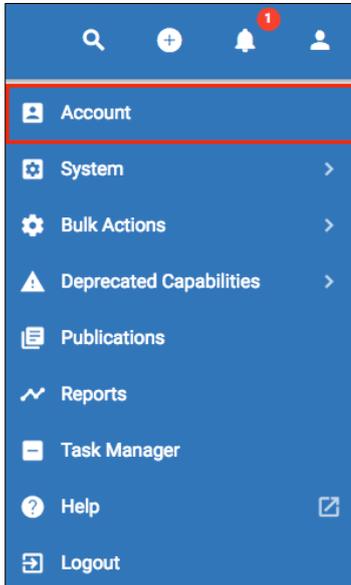


4. On the license row you wish to assign them to, click on the user field and choose the user from the list appears.



# User Removal (for Administrators)

1. To change or assign a license to existing users within the tool. Select the **User** menu > **Account**.



2. Then expand the application that you wish to add them into/modify their licenses for, depending on your licenses, you may only see once application there.



3. Find the license they are currently assigned to and hover over their username, you will an 'X' appear, click this to remove them from the license.

Licence Type	User	Trial	Expiration	Created	Last Login
Reviewer		<input type="checkbox"/>	31/05/2026 08:53 AM	05/05/2017 08:54 AM	
Contributor	AidanContributor	<input checked="" type="checkbox"/>	31/05/2026 08:53 AM	05/05/2017 08:54 AM	05/06/2017 01:35 PM
Professional	AidanTest	<input type="checkbox"/>	31/05/2026 08:53 AM	05/05/2017 08:54 AM	04/09/2017 12:22 PM
Professional		<input type="checkbox"/>	31/05/2026 08:53 AM	05/05/2017 08:54 AM	

4. They will still display in the users list below, but they will not be able to login and will have **No Licence**.

**Aidan Contributor**  
AidanContributor  
akelly@erwin.com  
05/06/2017 01:35 PM

---

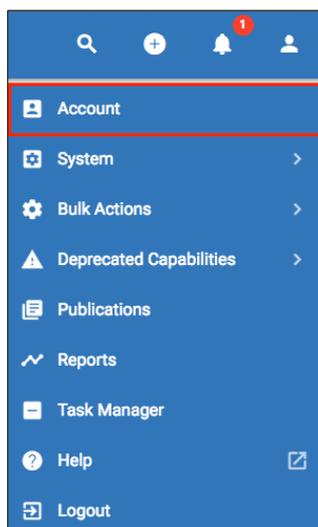
EA Agile: No Licence  
DM NoSQL: No Licence

---

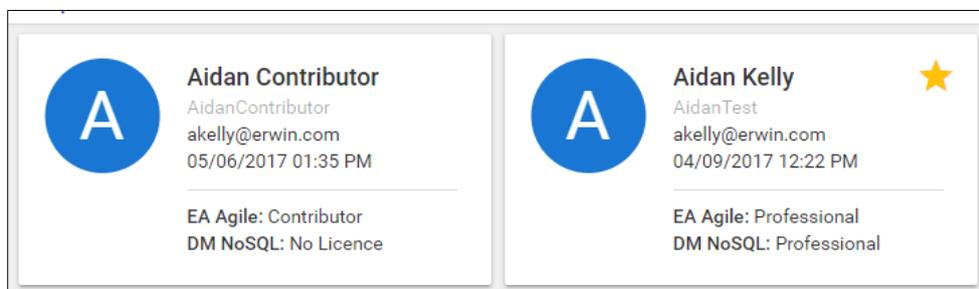
# Requesting More Licenses

To request more licenses, the client administrator should contact [support@erwin.com](mailto:support@erwin.com).

End users can find out who the administrator for their organization is in their profiles. Once logged in, select the **User** menu > **Account**.



Scroll down to the list of users and find the user with the **star** on their tile. This symbolises that they are the client admin. You can then use their details to contact them and request further licenses.



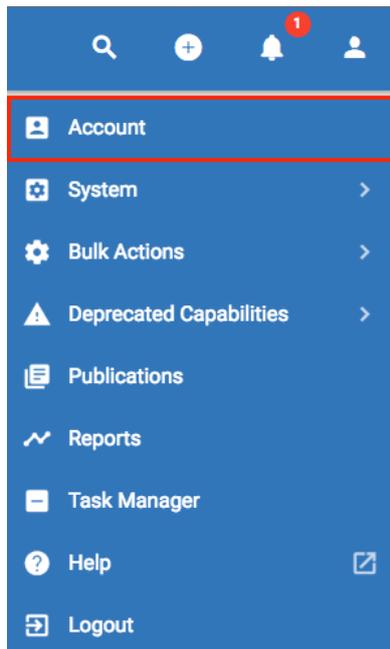
---

# Support Access

Client administrators can now make client areas available to allow our erwin support engineers to log in under their registered users' usernames. This will allow us to;

- See the issues first hand
- Test issues and diagnose problems easily
- Perform difficult functions for users when needed

This function is available under the **User** menu > **Account**.



- Find the **Allow Support Access?** field in the Client section at the top of the page
- Check the box
- **Save** the changes

A screenshot of a settings form. The form has a white background with a thin border. It contains the following fields: 'Name' with the value 'Corso Ltd.', 'Administrator' with the value 'Martin Owen' and a dropdown arrow, and 'Allow Support Access?' with a checked checkbox. A red box highlights the checkbox. In the bottom right corner, there is a 'SAVE' button.

The erwin support engineers can then log into the client area, using any registered username in the client area.

- This access can be taken away by the Client administrator at any time, and re-enabled if/when needed
- This will not affect the client administrator's settings or passwords
- Standard users will need to contact their client administrator to enable this

---

# Billing Options

In order to understand what options you need, please refer to the user license types [here](#).

If you wish to discuss enterprise level agreements (Purchase order), please contact sales: 1-800-78-erwin (1-800-783-7946)

## **Trials**

Trial users will be notified automatically at the end of the trial period to subscribe to the platform.

Users will automatically be disabled on the platform at the end of the trial and will be taken to the billing portal.

---

# Terms and Conditions

Our current list of our terms and conditions and privacy policies:

- <http://erwin.com/terms-of-service/>
- <http://erwin.com/legal/>
- <http://erwin.com/privacy/>

---

# Agile EA Tutorials

## 10-minute evaluation guide for Agile EA

What can you do in 10 minutes? Grab a coffee? Bathroom break? Buy a sandwich?

We realise your time is valuable, so we've put together a 10 minute guide to using our Agile EA module. Hopefully it will give you enough to get you interested and encourage you to complete our full Agile EA tutorial.

To make the steps below easier to follow, you may have to **log in** to your trial first of all. The license you will need to complete this tutorial is **Administrator**.

*To add further users to your client area, please see the [User management](#) page.*

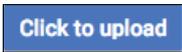
### Step 1: Import Data

*Time to value is essential when starting an enterprise architecture journey.*

*Often data is stored in a spreadsheet and is difficult to maintain, share and collaborate upon.*

*You need to assimilate that data quickly into a tool. CSV files can be produced by any spreadsheet and **imported** into the erwin platform.*

#### How:

1. Click on [this link](#) to download some sample CSV files.
2. Next, navigate to the **User** menu > **Bulk Actions** > **Object Import**.
3. Select  and then choose the file, or drop the file into the box. You do not need to unzip them to do this.
4. Select the workspace you would like to import into - Default is the base workspace - leave this as is if you have not yet created any workspaces.
5. Press  in the bottom right hand corner.

6. Check the imported data by clicking the All Objects button , and review the list of Application component objects.

(If they cannot be seen, use the drop down in the very top left corner , to move into the workspace that the data was imported into.)

---

## Step 2: Visualise in Diagram

*Sometimes textual information isn't the most effective way to communicate.*

**Diagrams** help you convey a clearer picture of your information and how it interacts with other data.

*Many views of enterprise architecture are better expressed as a diagram. Standards such as ArchiMate provide a standard way in which to represent architecture in a diagram.*

### How:

1. Click **Diagrams** from the views menu  and press add .
2. Enter a diagram name of **My applications**.
3. Ensure the diagram type is set to **Impact Diagram** by selecting from the drop down list.
4. In the **Dataset** area, select the **Search** function . Create a search for **Name contains CRM**, and click the **Search** button.
5. Click **ADD** to add the selected objects into the diagram's Dataset.
6. Click **SAVE** at the bottom right of the page, to save the diagram with the CRM objects.
7. Click the **DIAGRAM** tab at the top of the page.
8. Right hand mouse click on the **CRM** symbol, and choose **Show Relationships**.
9. Select **all** relationships by checking the box to the left of the **Relationship Name** header, and click **OK**.
10. Next you can move symbols around to tidy the diagram.
11. Click **SAVE** to ensure that the changes to the diagram are updated.

### Bonus step:

1. Choose **View** in the menu bar > **Layout > Circular**.
2. Click on the **CRM** symbol, and it should become the focus of the circle.
3. Next, Right hand mouse click on **Web Portal** and choose **Show Relationships**.
4. Select **all** relationships by checking the box to the left of the **Relationship Name** header, and click **OK**.
5. Click on any symbol to place it at the center of the Circular diagram.

## Step 3: View Roadmap

*Roadmaps provide key dates about your architecture assets such as lifecycles, availability, and other temporal information.*

*A high level visual presentation of a roadmap can be the key to creating the right impression to other stakeholders.*

*Each time you have to reproduce, re-align or re-do a roadmap, you need to do it effortlessly.*

---

## How:

1. Click the **Views** option  (left hand menu) and press the add button .
2. Enter a view name **My application portfolio**
3. Select an **Icon**, check the **Bookmark** checkbox, and change the **Order** to **1**.
4. In the Data section, set the **Object Type** drop down to **Application component**.
5. Make no changes to the filtering, so that all Application component objects will be included in the view.

Now the dataset has been selected, we can define and configure the visualizations;

6. Enter a Name **Application Live Dates** (where you see the text 'Visualization 1').
7. Select the visualization to be of type **Roadmap**.
8. Choose the color for the roadmap objects, by selecting the **circle** and using the **color picker**.
9. Choose the Attribute to be **Initial live date** from the drop down list. This is the date attribute that will be used by the roadmap.
10. Click **SAVE**.
11. Select the new **View** menu item, that should now be showing at the **top of the left hand menu**.

You can interact with the roadmap by scrolling the timeline and selecting and moving application components.

## **Step 4 Analyze in Pivot Table**

*A pivot table is a special spreadsheet view that allows you to summarize and explore data interactively. It is one of the most powerful features for analyzing and quickly organizing your concepts using attributes, filters and computational functions.*

*This example shows the cost savings for each application, grouped by Business Criticality, Action Priority, and Action.*

## How:

1. Click **Pivot Table** from the views menu  and press the add button .
2. Enter the Name **Application cost savings by importance**
3. Set Object Type as **Application component**
4. In the **Dataset** area, select the **Search** function. Search for **Criteria Type, Type, Application component**, and click the **Search** button.
5. Click **ADD** to add all the Application components to the Dataset.
6. Click **SAVE** at the bottom right of the page.
7. Press the **PIVOT TABLE** tab at the top of the page.
8. Select and drag **Action Priority** to the row above the table to the right of the Count drop down.
9. And do the same with **Business Criticality**.
10. Drag **Action** into the column to the right, below the Count drop down.
11. Next choose **Sum** where the word **Count** is displayed.
12. In the box below choose **Action Cost Savings**.
13. Click **SAVE** to save the changes to the Pivot Table.

- 
14. Access the **Action** menu , and **Refresh**  the pivot table, to see the updated values.

## Step 5 Create Publication

***Publications** represent the output that can be made available from the platform.*

*These can be electronic PDFs, web output, or images.*

### How:

1. Click on the **User** menu in the top right hand corner, and click **Publications**.
2. Click the + button in the bottom right of the list.
3. Type in the Name box **CRM System Findings**
4. In the **HTML** field, Click and type the words **CRM System Findings**, and hit enter.
5. Highlight the words and choose **Formats > Headings > Heading 1**.
6. Next type in the words 'These are my findings on the state of the CRM system.' and hit enter
7. Click **SAVE**.

## Step 6 Pull information into table view

***Tables** allow you to get output from the data you have stored in the repository. One way to report is to use a table view to pull out the information that we want.*

*In this example, we're working on all Application Components and some of their associated Concepts.*

### How:

1. Click the **My application portfolio** view (left hand menu), that we created earlier and press the **CONFIGURE** tab.
2. Select the **Plus** icon  to the right of the 'Application Live Dates' visualization, to create a new visualization.
3. Enter a Name **Application Lifecycles and Locations** (where you see the text 'Visualization 2').
4. Select the visualization to be of type **Table**.
5. Check the Attributes **Locations, Initial live date** and **Lifecycle status** - to be the columns of the table.  
It is recommended that the Filter is used in locating these attributes.
6. Click **SAVE**.
7. Select the new **Visualization** tab to see the table view output.

## Step 7 Add views to Publication

### How:

1. Click **Diagrams** from the views menu and select the **My applications** diagram created earlier.

- 
2. Access the **Action** menu , and click the **Publications** button .
  3. Press **Add to publication** on the Publication screen, and select the **CRM System Findings** publication from the drop down.
  4. Click **ADD**  to append the diagram to the publication.
  5. Repeat these steps for the pivot table and the roadmap that were created earlier, to also include them in the publication.
  6. Access the **Publications** list and select **CRM System Findings** to see that the diagram, pivot table, and roadmap, have been added to the publication.
  7. Add text and edit the publication as you wish.
  8. **Save** and/or **Download** the publication as a PDF, from the **CONFIGURE** tab.

---

# Tutorial for Agile Enterprise Architecture

**Role needed to use:** [Client Administrator](#)

This scenario deals with an insurance company, ArchiSurance, who are currently experiencing issues with their claims processes. The issues are around the manual processes that they have put in place to handle the volume of claims they are receiving, especially around human input of data and the manual checking of claims. Both of these process steps have led to incorrect data being input for the claims and insufficient approvals on signoff practices.

The role you will be playing in this tutorial is as an **Administrator**. To add further users to your client area, please see the [User management](#)

page.

This tutorial is designed to be run on a clean client area with the ArchiMate framework applied to it. It can be used on client areas that are already in use, but please be aware that the screenshots may not then match exactly.

  Please allow 120-180 minutes to complete this tutorial.

Specific initiatives for Enterprise Architecture (EA) can be targeted towards specific business goals. To focus on our issues with a claims handling process that the organization is seeing, we can create a new goal to improve our claims handling process.



- Select the **All Objects** module , and then click the **add** button at the bottom of the pane .
- In the new object dialog, fill in the details as shown below;
  - **Name:** Revise claim handling process
  - **Description:** Revise the claim handling process to be more streamlined.
  - Select Goal as the **Object Type**. (Click in the Object Type area and enter 'go' in the Filter.)  
Selecting an Object Type will automatically display the object type's Attributes.

- Notice the **General** group that appears. Set its **Attributes** as shown below; (clicking on them will open their entry fields.)
  - **Attainable:** true
  - **Target Date:** 12/12/2016 (+ time)
  - **Lifetime Value:** 5 (by moving the sliders as shown)

- **SAVE** the new Goal (using the SAVE button in the bottom right hand corner).

**i** | The goal is now an object in the repository. It is re-usable in other places and can be shown in diagrams.

***We will now create a Requirement object.***



- Click the **add** button that is visible at the bottom of the objects list on the left of the screen.
- In the new object dialog, fill in the details as shown below;
  - **Name:** Provide automatic billing for customers
  - **Description:** Ensure that customers can pay monthly or annually for their premiums.
  - **Object Type:** Requirement



- **SAVE** the new Requirement.

You can now see how quickly and easily users can add objects and add attribute information. These objects can now be used to contribute to the building of a model, and also populated with further information, allowing us to interrogate and analyse later on.

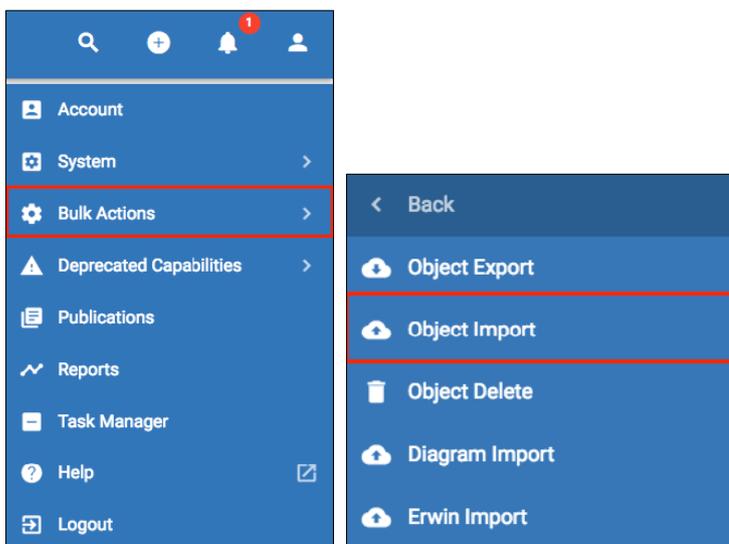
# Importing Data

We will use the import feature to import some more objects.

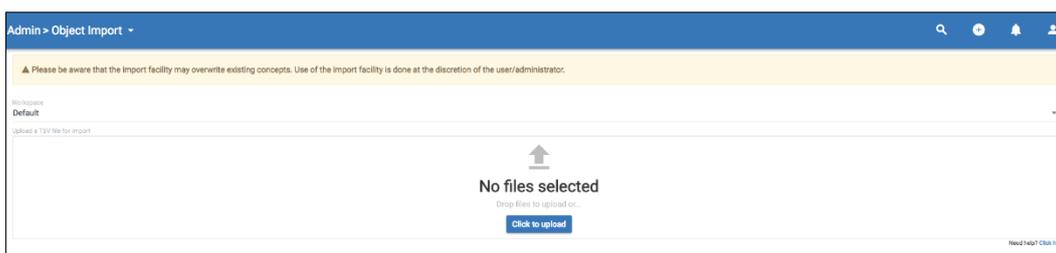
- Click the URL below to download the import file onto your machine

## Download .csv file

- Once you have downloaded the file to your machine, in the Agile EA tool, navigate to the **User** menu > **Bulk Actions** > **Object Import**.



- Select the file that was just downloaded to your machine; **Corso\_Agile\_EA-Tutorial Req Import.zip** and drag it into the upload box, or press the **Click to upload** button.



- Click the **IMPORT** button .

If the import is successful, a **success** message will appear on your screen. A log should also report a successful import of the individual objects and relationships.

Now that you have successfully imported some objects, with attributes, and relationships between them, we can begin to use the analysis tools to investigate the relationships between the model objects.

---

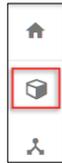
# Managing Objects

The core components of the platform are termed objects.

There are different types of object, and the Agile EA platform conforms to both ArchiMate and TOGAF.

The full EA metamodel defines the available object types.

The objects are visible within the All Objects view module.



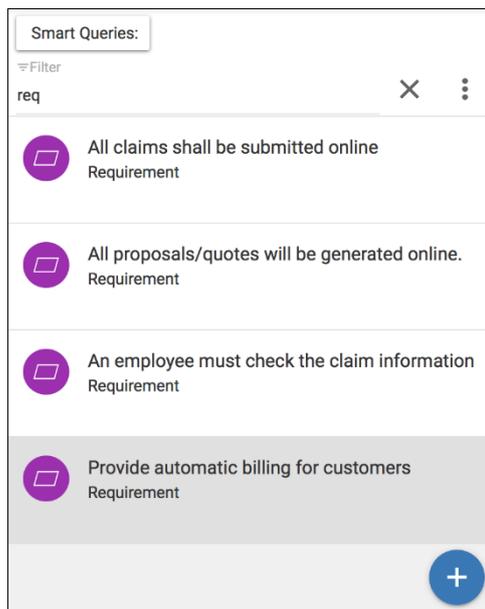
- From the toolbar select **All Objects**

The All Objects list can be dynamically filtered;



Click on the Filter icon, and type **req** to list only object containing the letter 'req', in their Name, Object Type, or Tags.

- Select the requirement **Provide automatic billing for customers** from the list.



- The details will be displayed with the attributes groups below it - look in the **General** category.
- **Set** the attributes as below
  - **Difficulty:** Medium
  - **Cost:** 125000
  - **Status:** Draft
  - **Start Date:** 01/15/2018
  - **MoSCoW:** Must: 100

- **Type:** Functional (look in the **Classification** attribute category)



To change the layout style, press the button next to the filter icon to change between list and grid layout



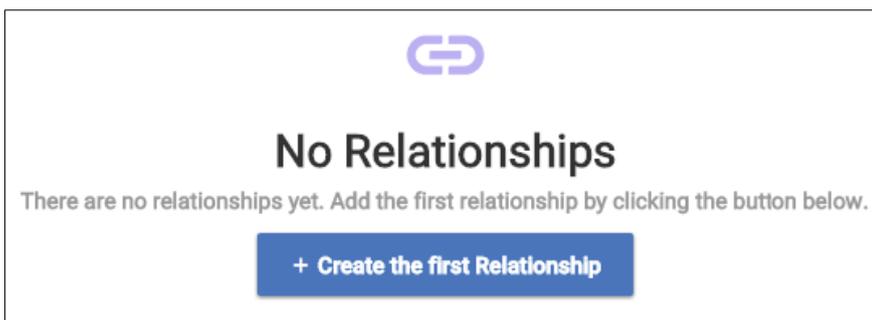
General		Additional Information	
Difficulty	Medium: 50	Effective Date	Effective Date...
Stability	Stability...	Priority	Priority...
Cost	125000	Implication	Implication...
Status	Draft	Metric	Metric...
Start Date	15/01/2018 10:32 AM	Classification	
MoSCoW	Must: 100	Type	Functional
Rationale	Rationale...		
Statement of requirement	Statement of requirement...		
Acceptance criteria	Acceptance criteria...		
Ideas realized	Ideas realized...		
Goals	Goals...		
Actors	Actors...		

- **SAVE** the Requirement, to update the changes.



You can **customize** the objects with your own attributes, if you want to add additional detail.

- Select the **RELATIONSHIPS** tab, at the top of the page, in order to create relationships to other objects.
- Select the **Create the first Relationship** button, to add a new relationship.



- Click the **Relationship Type** field, and enter **real**. Select **realizes (is realized by)** from the list.
- Click the **Object Type** field, and select the **Goal** type.
- In the **Objects** list returned, select the **Revise claim handling process** Goal, by checking its checkbox.

Relationship Type: realizes (is realized by)

Object Type: Goal

Requirement: Provide automatic billing for customers realizes

<input checked="" type="checkbox"/>	Name	Type
<input checked="" type="checkbox"/>	Revise claim handling process	Goal

Items per page: 5 1-1 of 1

CANCEL SAVE

**SAVE** the relationship, and it should now be listed in the relationship table.

← DETAILS RELATIONSHIPS

Relationship Name	Object Type	Objects
realizes (is realized by)	Goal	Revise claim handling process

Items per page: 10 1-1 of 1

---

# Pivot Tables

Many users use spreadsheets for looking at information and analyzing their attributes. The platform provides a pivot table capability that allows this type of behavior.

A pivot table provides a spreadsheet like data summarization tool, through which you can make powerful decisions dynamically.

The pivot table provides the ability to 'slice and dice' your objects by attribute.

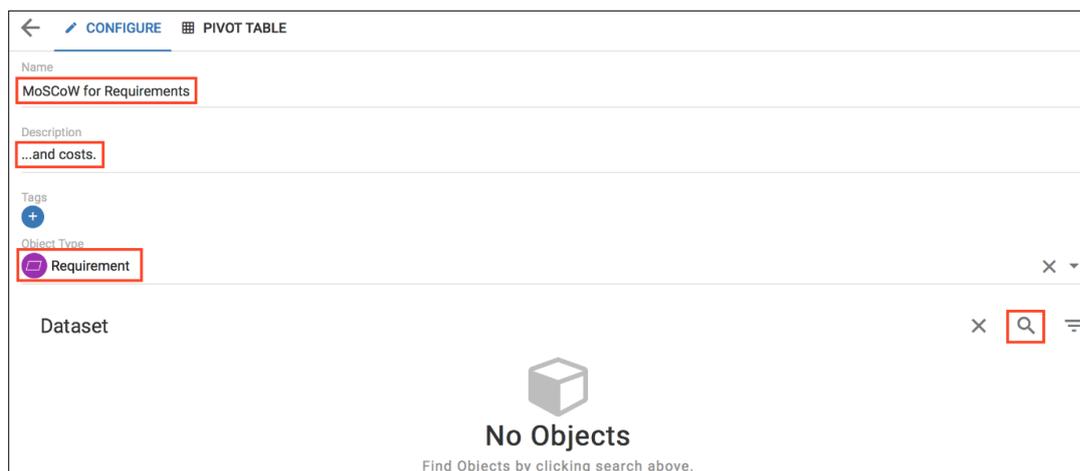
Next, you will look at some further analytics, filter the requirements of value to us and examine objects that have certain attribute values set.

Pivot tables provide a view on the objects and can be created for any type of object.

- Select **Pivot Tables** module , and then click the **add** button .

In this example, you will build a pivot table that, shows the MoSCoW rules for Requirements (Must have, Should have, Could have and Would like to have), and also shows their costs.

- Fill in the fields as shown below;
  - **Name:** MoSCoW for Requirements
  - **Description:** ...and costs.
  - **Object Type:** Requirement
  - In the **Dataset** section, select the **Search** option to begin selecting the Requirements to be added to the pivot table.



- In the **Search for Objects** dialog that opens press **Search**, because you already limited the dataset to requirements, only objects of that type will be returned.



	MoSCoW	Must : 100	Should : 75	Totals
Requirements				
All claims shall be submitted online			1	1
All proposals/quotes will be generated online.			1	1
An employee must check the claim information		1		1
Provide automatic billing for customers		1		1
Totals		2	2	4

A column for each of the values from the MoSCoW attribute, that are used by the listed Requirements, will be added to the pivot table.

- Drag the **Difficulty** attribute into the top panel to re-structure the table. Note that the order of the attributes will determine the 'grouping' of the columns.
- Change **Count** to **Sum**, by clicking Count, and selecting Sum from the list.
- Click the new drop down list that appears underneath, and select **Cost** from the list of numeric attributes.
- Change **Table** to **Heatmap**, to highlight the numbered cells in varying shades of white to red.

Requirements	MoSCoW	Must : 100		Should : 75		Totals
	Difficulty	Low : 100	Medium : 50	High : 10	Medium : 50	
All claims shall be submitted online				50000		50000
All proposals/quotes will be generated online.					300000	300000
An employee must check the claim information		50000				50000
Provide automatic billing for customers			125000			125000
Totals		50000	125000	50000	300000	1020000

This shows an easy way to represent financial information, and draw conclusions quickly, as to how difficult it is, and how much it will cost, to implement the requirements that we have decided are a 'Must'.

- **SAVE** the pivot table.

**i** Clicking on the 'down arrow' to the right of an active attribute name, will allow filtering of columns and rows. A filtered attribute name will be shown in *italic*.

# Charts

Charts provide a graphical view of our objects, and are often useful tools to make decisions. Charts shouldn't be used in isolation but are a tool in your kit bag to make meaningful decisions about your portfolio.

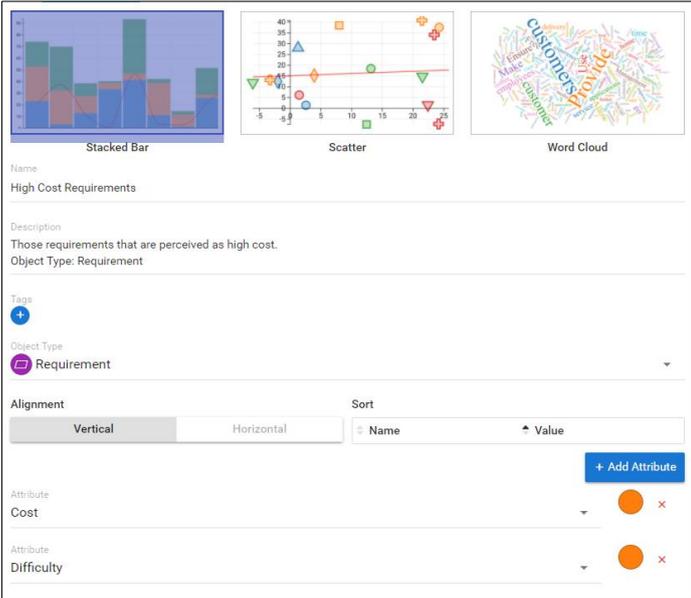
Charts are fully configurable.

The types consist of:

- Bar charts
- Scatter charts
- Word clouds

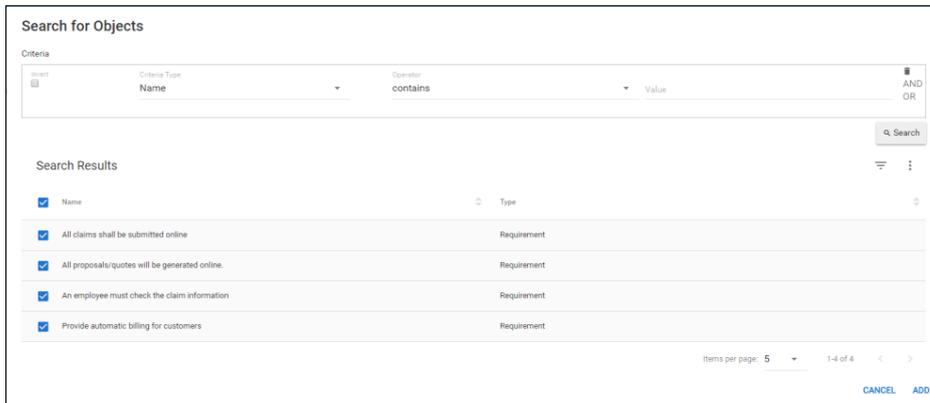
- Select the **Charts** module , and then click the add button .
- Fill in the details as below

- **Chart Type:** Stacked Bar
- **Name:** High-Cost Requirements
- **Description:** Those requirements that are perceived as high cost.
- **Object Type:** Requirement
- **Bar Alignment:** Vertical
- **Sort By:** Value (Descending)
- Press the **Add Attribute** button
- In the first attribute dropdown, select **Cost**
- In the second drop down select **Difficulty**



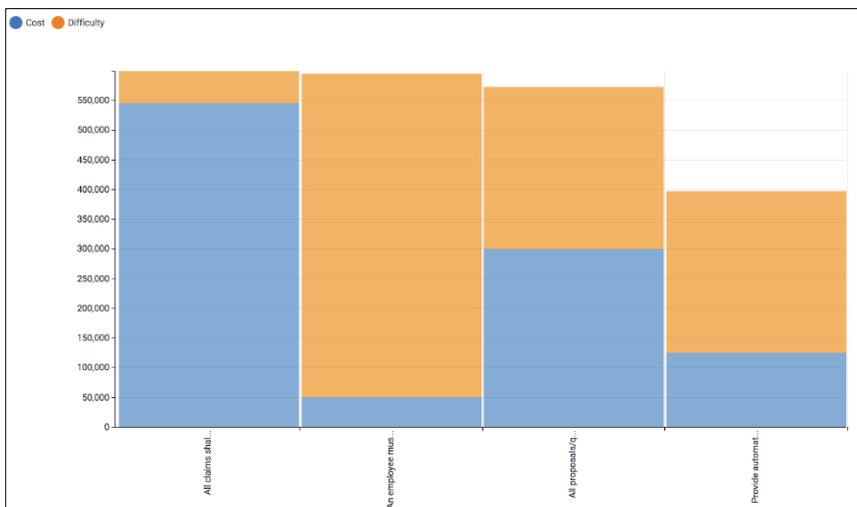
The screenshot displays the chart configuration interface. At the top, three chart types are shown: Stacked Bar, Scatter, and Word Cloud. Below these, the configuration form for 'High Cost Requirements' is visible. The form includes fields for Name, Description, Object Type, Tags, Object Type (Requirement), Alignment (Vertical/Horizontal), Sort (Name/Value), and two attribute dropdowns (Cost and Difficulty). An 'Add Attribute' button is also present.

- In the **Data** section, select the **Search** option to begin selecting the objects to be added to the chart.
- In the section that opens click **Search**, when you selected the object type on the previous screen you already limited the object type and we want all objects of that type.



- Check all existing Requirements, and click **ADD**.
- Click **SAVE**.

Select the **CHART** tab, at the top of the page



The Cost and Difficulty attributes have colors assigned automatically, they can be changed on the **CONFIGURE** tab, if required.

- Select an object from within the chart to open the in-context editing pane – allowing editing of the object and its attributes directly from the chart.

Here we have seen how we can quickly create simple stacked bar charts of objects, displaying some of their numeric attributes.

**i** Note that numeric ranges are scaled to create similar sized bars



---

# Application Components and Analysis

You will now shift your focus to understand the application components, that realize the requirements, to fulfil the goal **Revise claim handling process**.

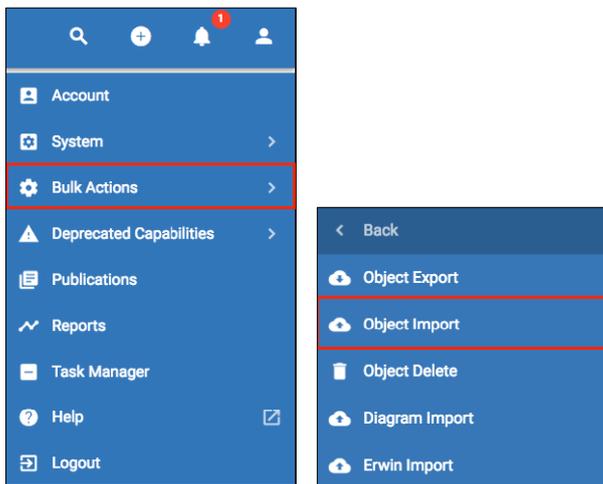
The first task is to create the Application component objects, update their attributes, and establish relationships to other objects in the model. You can then view and analyze the model, using diagrams and roadmaps, and track progress using kanban boards.

To do this we will use the import function again to import some Application components.

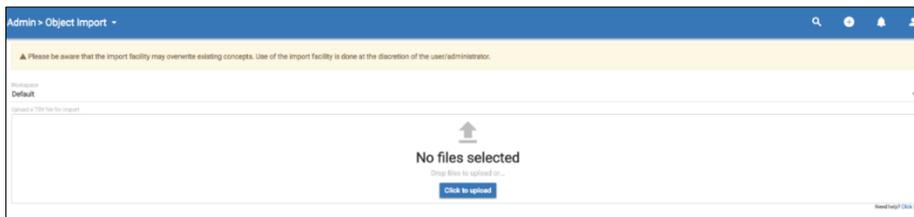
- Click the URL below to download an import file onto your machine;

## Download .csv file

- Once you have downloaded the file, in the Agile EA tool, navigate to the **User** menu > **Bulk Actions** > **Object Import**.



- Select the file that was just downloaded to your machine - **Corso\_Agile\_EA-Application\_Component\_import.zip** and drag it into the upload box, or press the **Click to upload** button.



- Click the **IMPORT** button .

If the import is successful, a **success** message will appear on your screen. A log should also report a successful import of the individual objects and relationships.

---

# Scorecards

Some of the analysis we want to perform is going to make use of Scorecards within our Application components.

A Scorecard calculates a 'score' from attributes of the scorecard. One or more of these attributes can take their value from other attributes of the object. For example the value of **Application Lifetime Cost** is calculated from several attributes; Expected Lifespan, Monthly User Cost, Number of Users, License Cost, and Maintenance Cost.

- Enter the following values into the named Application component objects' **Application Lifetime Cost** attribute, (instructions below table).

## *Application Lifetime Cost Scorecard Values*

Object to edit	Values to enter
Web Portal	Expected lifespan – 6 years License Cost (Perpetual) – 30000 Maintenance Cost (Perpetual) – 3000
Home and Away Financial Application	Expected lifespan – 5 years Monthly user cost (SaaS) – 250 Number of Users - 100
CRM	Expected lifespan – 5 years Monthly User Cost (SaaS) – 350 Number of Users - 50
Home and Away Policy Administration	Expected lifespan – 5 years Monthly User Cost (SaaS) – 200 Number of Users - 15
Call Center Application	Expected lifespan – 6 years License Cost (Perpetual) – 50000 Maintenance Cost (Perpetual) - 10000
Customer Data Access	Expected lifespan – 5 years Monthly User Cost (SaaS) – 50 Number of Users - 25

---

**To do this:**



Select the **All Objects** module

- Locate each of the **Application component** objects, one at a time, from the table shown above.
  - Expand the **Application Lifetime Cost** attribute (example image below) - this attribute is part of the **Costs** category.  
You might want to remove other categories or filter the attribute list with the text **cost**.
  - Enter the values into the fields as per the table above.
  - **SAVE** the changes to the object.
  - Continue to the next object in the table.

Costs

**Application Lifetime Cost** ✕ ^

The cost of the application over its expected lifetime.

0

**Expected lifespan**  
What is the expected lifespan of this application in years?

1 Years 2 Years 3 Years 4 Years 5 Years 6 Years 7 Years 8 Years 9 Years 10 Years

**Monthly User Cost (SaaS)**  
What is the monthly cost per user?

0

**Number of Users (SaaS)**  
What is the total number of users?

0

**License Cost (Perpetual)**  
What is the perpetual license cost?

0

**Maintenance Cost (Perpetual)**  
What is the perpetual maintenance cost?

0

---

# Locations

Create the following Location objects:

- **Name:** Back Office
- Select the **Object Type "Location"** (The Location attributes should appear below).
- **SAVE** the object.

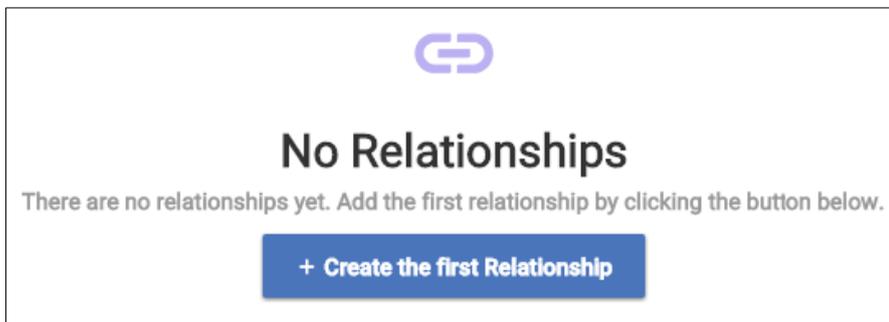


The screenshot shows a mobile application interface for editing a 'Location' object. At the top, there is a back arrow and a 'DETAILS' tab. Below this, the 'Name' field is filled with 'Back Office'. The 'Description' field is empty. There is a 'Tags' section with a plus icon. The 'Object Type' is set to 'Location'. At the bottom, there is a 'Communities' section with 'All' selected.

Select the **RELATIONSHIPS** tab



- Click the **Create the first Relationship** button, to add a new relationship



- Click on the **Relationship Type** field and choose the '**aggregates (is aggregation of)**' relationship.
- Click on the **Object Type** field and select the **Application component** type.
- In the list of **Objects**, check **Customer Data Access, Home and Away Financial Application**, and **Home and Away Policy Administration** checkboxes.

Relationship Type	
aggregates (is aggregation of)	
Object Type	
Application component	
Location: Back Office aggregates	
Name	Type
<input type="checkbox"/> Call Center Application	Application component
<input type="checkbox"/> CRM	Application component
<input checked="" type="checkbox"/> Customer Data Access	Application component
<input checked="" type="checkbox"/> Home and Away Financial Application	Application component
<input checked="" type="checkbox"/> Home and Away Policy Administration	Application component

Items per page: 5 1-5 of 6 CANCEL SAVE

Once all three Application components have been checked, **SAVE** the relationships.

The relationships should look like the following;

DETAILS RELATIONSHIPS		
Relationship Name	Object Type	Objects
aggregates (is aggregation of)	Application component	<input checked="" type="checkbox"/> Customer Data Access <input checked="" type="checkbox"/> Home and Away Financial Application <input checked="" type="checkbox"/> Home and Away Policy Administration

The recent import of Application components also included a Location named "Front Office", with a single relationship. Additional relationships can be created.

**To do this:**

- Locate the **Front Office** Location object.
- Access the **RELATIONSHIPS** tab

Smart Queries: front		
Relationship Name	Object Type	Objects
aggregates (is aggregation of)	Application component	<input checked="" type="checkbox"/> Call Center Application

Items per page: 10 1-1 of 1

- Select the **Edit** button , and check the checkboxes for **CRM** and **Web Portal**.
- **SAVE** the new relationships.

---

# Diagrams

Diagrams provide views of the objects and their relationships.

Out of the box, Agile EA provides support for all of the ArchiMate 3.0 views, and Business Capability extensions.

The diagrams provide views that are representationally consistent with the underlying model. This means, that if there is a relationship in the model between two objects that exist in a diagram, then the relationship must exist in the diagram. Also, if a relationship is drawn between two objects on a diagram, the model is updated with the relationship.

Diagrams can be rendered from the model content, using the **CONFIGURE** page, or users can draw the diagram and create new model content.

**i** Relationships between objects **Must** be represented on a diagram but can be hidden. Deleting a relationship on a diagram will result in the relationship being removed from the model also.



- Select the **Diagrams** module , and then click the **add** button .
- Fill in the fields as below
  - **Name:** ArchiSurance Application Co-operation
  - **Description:** The co-operation of the ArchiSurance Applications.
  - **Diagram Type:** Application Co-operation Viewpoint

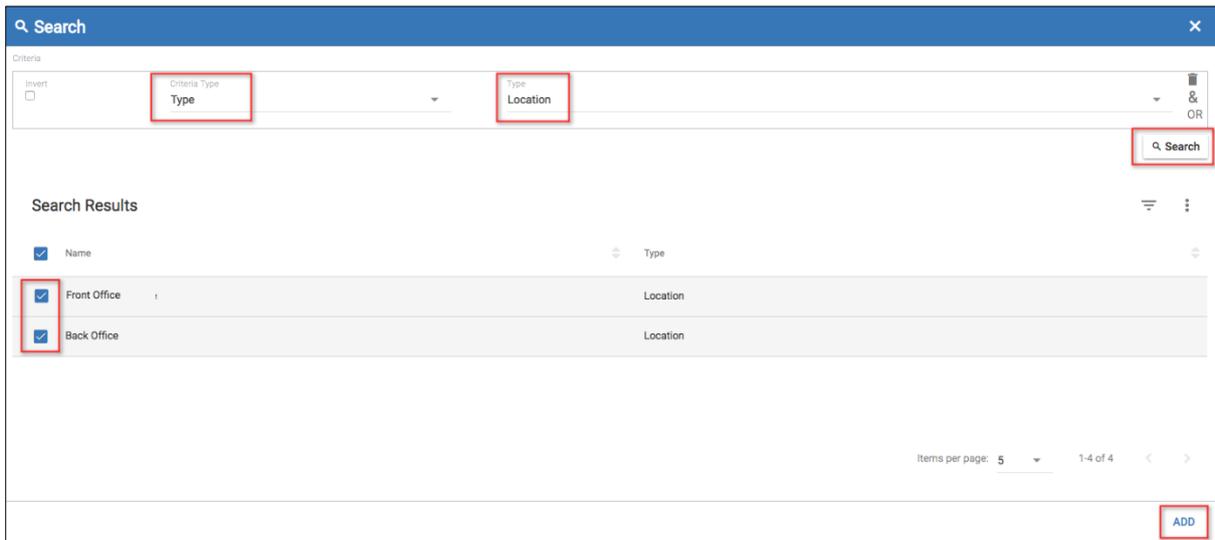


<	CONFIGURE	DIAGRAM
Name	ArchiSurance Application Co-operation	
Description	The co-operation of the ArchiSurance Applications	
Tags	+	
Diagram Type	Application Co-operation Viewpoint	

The Application Co-operation viewpoint describes the relationships between application components in terms of the information flows between them, or in terms of the services they offer. This viewpoint is typically used to create an overview of the application landscape of an organization. This viewpoint is also used to express the (internal) co-operation or orchestration of services, that together support the execution of a business process.

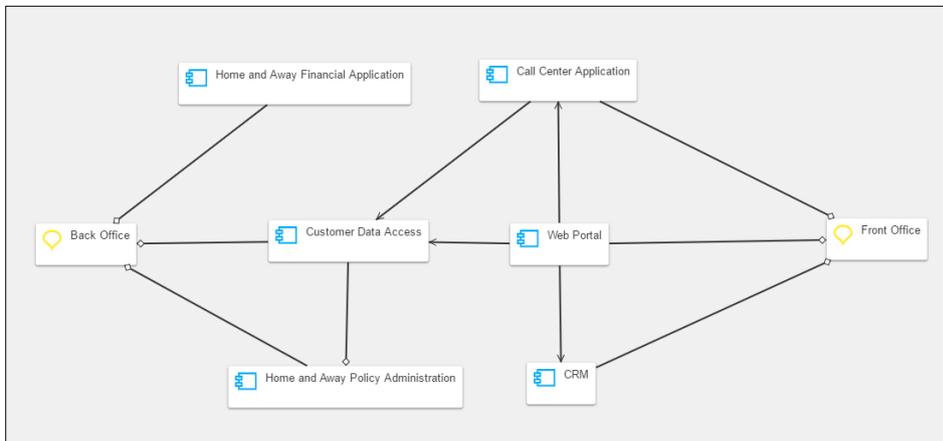
- In the **Dataset** section, click the **Search** option to select existing objects to add to the diagram.
  - Select **Type** for the **Criteria Type**, and **Location** for the **Type**.
  - Click **Search** followed by **ADD** to add both Back Office and Front Office locations to the diagram.

**i** Only the object types related with this viewpoint will be presented in the Search Box results.



- In the **Dataset** section, select the **Search** option again.
  - Select **Type** for the **Criteria Type**, and **Application component** for the **Type**.
  - Click **Search** followed by **ADD**, to add all Application components to the diagram.
  - Click the **DIAGRAM** tab at the top of the page to display the diagram view.

All relationships that exist between the objects will be automatically drawn on the diagram.



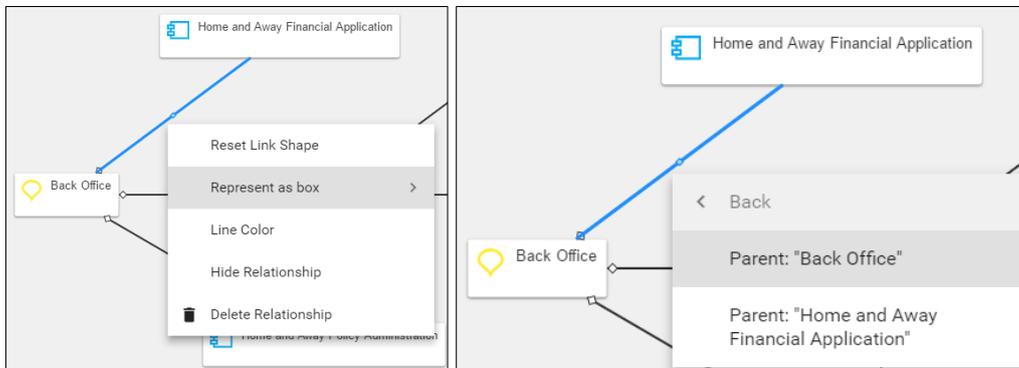
The 'nodes' can be repositioned by dragging them.

**i** | Aligning the nodes 'neatly' might be better achieved by enabling the "Snap To" Grid Setting.

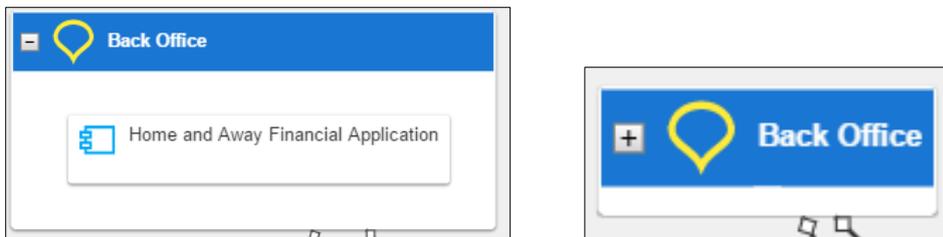
Any relationship can be represented using a 'box in box' style, where a relationship is implied by one symbol being embedded inside another.

- Right-mouse click on the relationship between **Back Office** and **Home and Away Financial Application**.

- Select the **Represent as box** menu option, and then select for the **Parent: "Back Office"** option, to set Back Office as the embedding node.

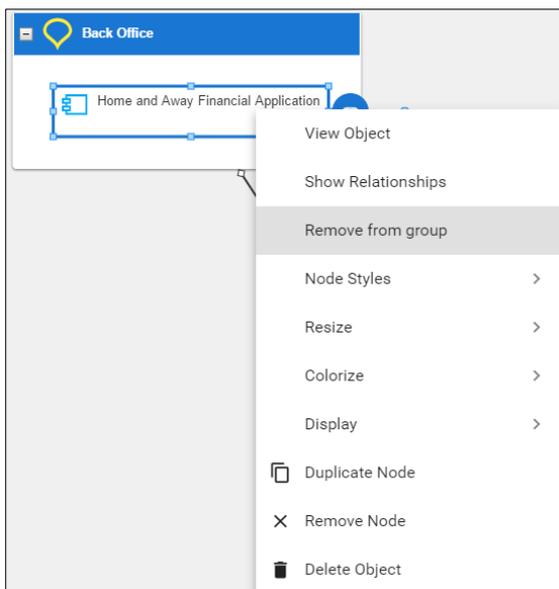


The 'parent' node is placed around the 'child' node. Clicking the parent node's  button will hide its children.

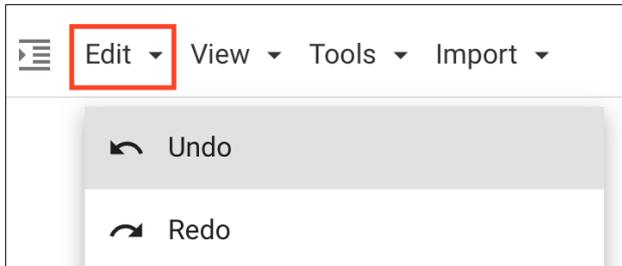


 Relationships to hidden child nodes are shown 'connected' to its parent.

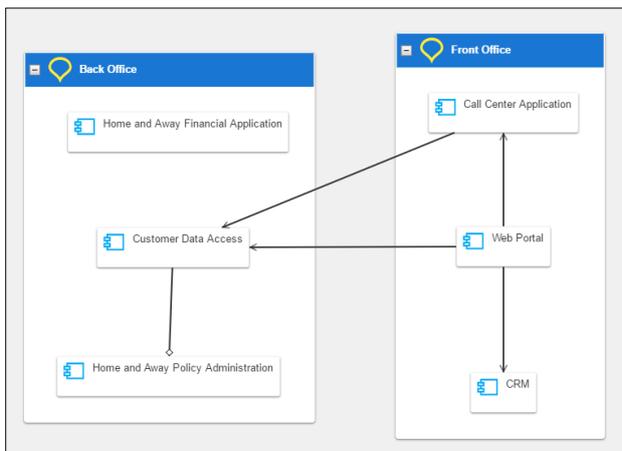
A child node can be removed from the 'box in box' group, by right-mouse clicking on the child node and select **Remove from group**.



**i** Any unwanted changes to diagrams can be reversed using the **Undo** menu option.



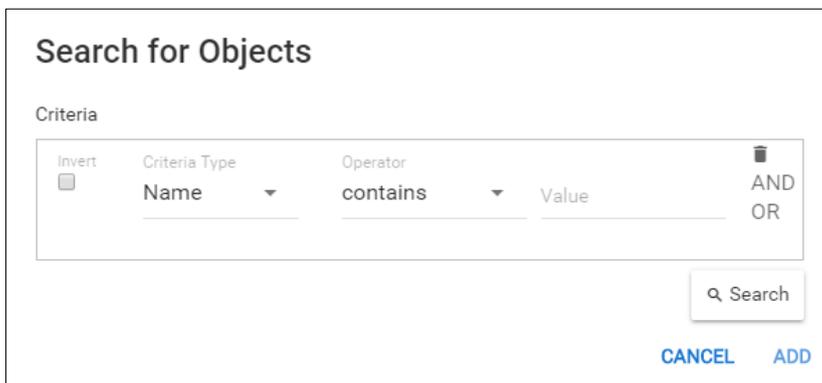
We could end up with a diagram that looks like;



**i** Initially, parent nodes are resized to be large enough to contain all child nodes. Moving the child nodes could result in the parent resizing.  
If a parent box is manually resized, its size will be used as a minimum size, preventing the parent box from automatically shrinking.

To add new objects to the diagram, either;

- Right click on the diagram and select **Insert > Existing Objects**

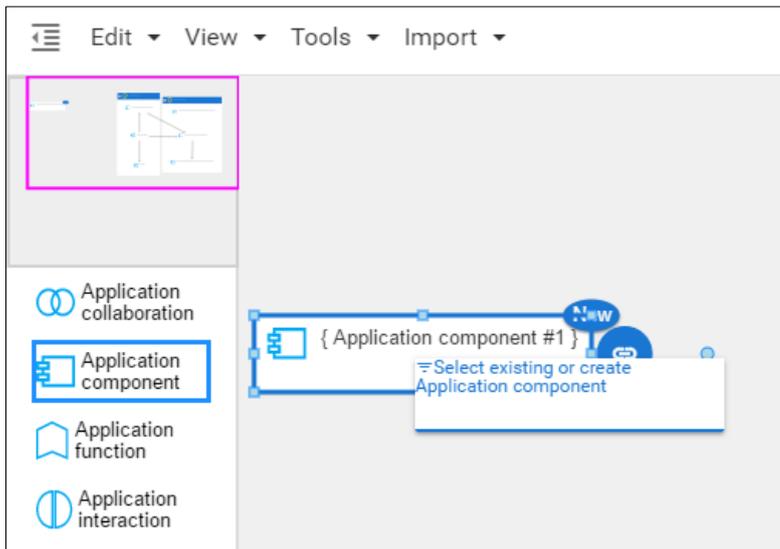


---

This will display the standard Search for Objects function to find objects to add to the diagram.

- Expand the 'drawing toolbar' panel using the **expand** button  .

Then drag the required object type onto the diagram.



When the object type is first dragged onto the diagram, it will expect a name to be entered, (if not, double-click the symbol to enter a name).

As a name is entered, a filtered list will be displayed, of all objects of the selected type that have names that contain the entered characters, that are not already on the diagram.

Either;

- continue to type a new name to create a new object, or
- select an existing object from the list.

**i** The diagram contains all the objects that were imported into the model, that are allowed on the Application Co-operation Viewpoint, therefore only new objects can be added.

Create a new diagram of type **Requirements Realization Viewpoint**.

- Click the **add** button  .

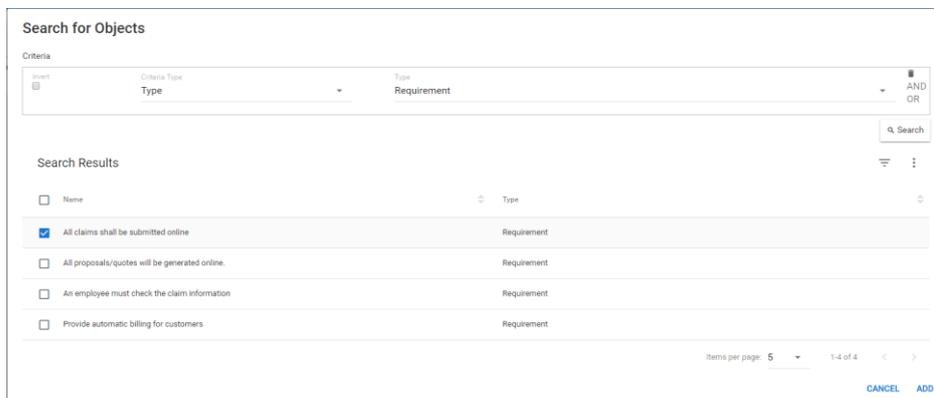
Fill in the fields as shown below;

- **Name:** Revise claims handling process
- **Description:** How the requirements are realized for claims handling process improvements.
- **Diagram Type:** Requirements Realization Viewpoint



The Requirements Realization viewpoint allows the designer to model the realization of requirements by other objects such as; business actors, business services, business processes, application services, application components. In addition, this viewpoint can be used to refine requirements into more detailed requirements. There are many more object types available on this viewpoint, than there were on the Application Co-operation viewpoint.

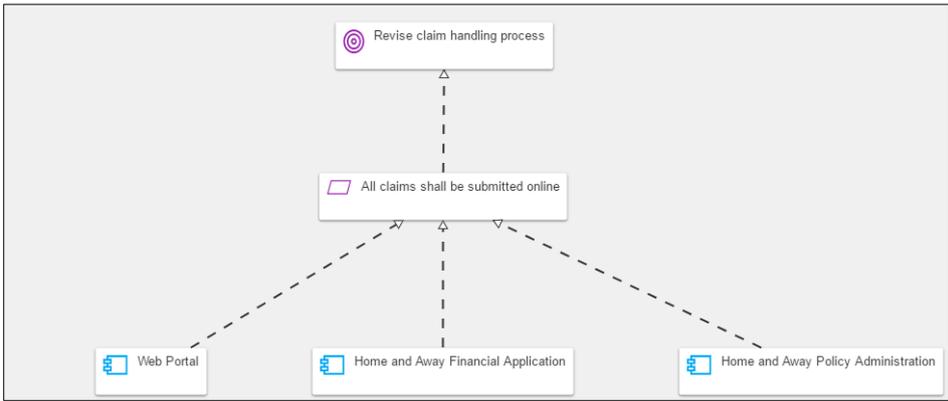
- In the **Dataset** section, select the **Search** option.
  - Select **Type** for the **Criteria Type**, and **Requirement** for the **Type**.
  - Click **Search**.
  - Deselect all of the results except for '**All claims shall be submitted online**' then click **ADD**.
  - Click **SAVE**.



- In the **Dataset** section, select the **Search** option again.
- Select **Type** from the **Criteria Type**, and **Application Component** for the **Type**.
- Click **Search**.
- Select only the following: **Home and Away Financial Application, Home and Away Policy Administration** and **Web Portal**, then click **ADD**.
- Now add all **Goal** objects to the diagram.
- Once completed, click **SAVE**.

Select the **DIAGRAM** tab from the top of the page





The diagram shows a 'slice' of the model, from a goal that is realized by a requirement, that in turn is realized by application components.

---

# Views and Visualizations

Kanban Boards and Roadmaps are the first of the existing views to be migrated into the **View Manager**. We can now use them with queries, to create dynamic datasets with multiple visualizations, rather than having to define a static dataset for each module.

**Kanban boards** are used to track work in progress. They are key to working in an agile manner and can be used to drive teams to deliver work.

EA objects can be added to kanban boards to show where they are in a work stream.

Objects are represented as cards on the board. Objects may be moved from one stage to another by dragging a card.

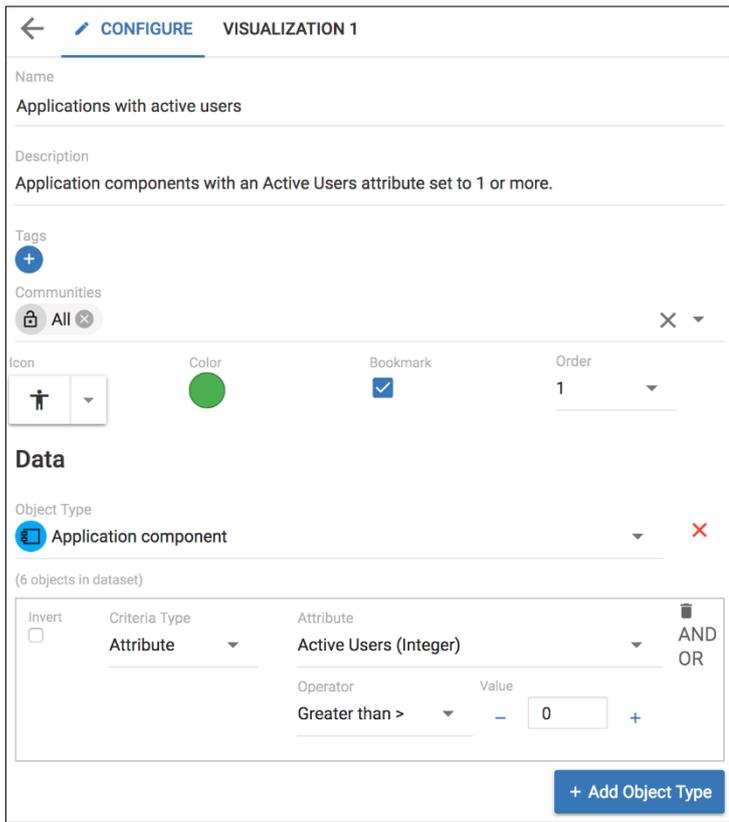
First we will define our dataset;



- Select the **Views** module , and then click the **add** button  to create an empty View.
- Fill in the view details on the **CONFIGURE** page with the details shown below, to select a very simple dataset, around applications with active users
  - **Name:** Applications with active users
  - **Description:** Application components with an Active Users attribute set to 1 or more.
  - **Icon:** Select an appropriate icon, and color, to identify the view.
  - **Bookmark:** Check the checkbox, so that the view will appear in the menu.
  - **Order:** Select 1 from the drop down, so that the bookmarked view will appear first in the menu.

The **Data** section contains the dynamic query that selects objects to be included in the view. The selection of an Object Type will include all objects of that type in the Dataset. Any selection criteria will filter the dataset.

- **Object Type:** Select **Application component**
- **Criteria Type:** Select **Attribute**
- **Criteria options:**
  - **Attribute:** Select the **Active Users (Integer)** attribute
  - **Operator:** Select **Greater Than >**
  - **Value:** set the value to **0**



All 6 Application components have been included in the dataset. The query has not reduced this number, as they all have at least 1 active user. The query can be made more complex by combining further selection criteria. Clicking **AND** will make the query more restrictive, while **OR** will add to the possibility of being included in the dataset.

Should the query be extended to **AND** the **Business Criticality (Array)** attribute that is **High: 100**, the dataset would be reduced to 3 Application components.

**i** Note that multiple Object Types, with their own queries, could be part of the view by clicking the + **Add Object Type** button.

- **SAVE** the View.

A view of, Application components that have active users, now exists. By default, the visualization will be set to Table, but simply identify the object type and its name.

# Table Visualization

The table Visualization lists the names of the objects preceded by their object type icon, followed by columns showing any selected attributes.

- Name the Visualization - **Users Table - Live** (replacing "Visualization 1")
- Ensure that the visualization is set to **Table**.
- In the **Configuration** section, the attributes to be included in the table are selected.
- Click the checkbox next to the following attributes; (It is recommended that the Filter is used to locate them.)
  - Number of Users
  - Active Users
  - Initial live date
  - Lifecycle status
  - Lifecycle – Live

It is recommended that the Filter is used to locate attributes, where too many attributes can make it difficult to locate them by scrolling.

The screenshot displays the configuration interface for a visualization titled "USERS TABLE - LIVE". The interface includes a "Name" field containing "Users Table - Live", a visualization type selector with "Table" selected, and a "Configuration" section. In the configuration section, a "Filter" dropdown is set to "User". Below it, a table lists attributes with checkboxes. The "Number of Users" and "Active Users" attributes are checked. At the bottom, there is a "Preview" button.

<input type="checkbox"/>	Object Type	Attribute Name	Categories	Description
<input checked="" type="checkbox"/>	Application compo...	Number of Users		How many users are th...
<input checked="" type="checkbox"/>	Application compo...	Active Users		How many active users...

- Click the **Preview** button.

Users Table - Live Preview

Users Table - Live

Name	Number of Users	Active Users	Initial live date	Lifecycle status	Lifecycle - Live
Call Center Application	50	80	04/01/2009	Live	04/01/2009 > 30/09/2015
CRM	35	30	07/03/2012	Live	07/03/2012 > 31/12/2019
Customer Data Access	25	24	01/01/2013	Phasing Out	01/01/2013 > 31/12/2017
Home and Away Financial Application	100	58	23/06/2012	Live	23/06/2012 > 31/12/2018
Home and Away Policy Administration	16	15	24/01/2014	Live	24/01/2014 > 31/12/2024
Web Portal	250	440	01/01/2010	Phasing Out	01/01/2010 > 01/01/2015

Items per page: 10 1-6 of 6

**i** Note that the order of the columns is determined by the order the attributes are selected.

- Select **X** in the top right, to exit the preview.
- **SAVE** the View.



Clicking the icon, that now appears at the top of the left hand menu , will open the View's visualization.

# Kanban Visualization

Additional visualizations can be created for views. To create a kanban from our existing view;

- Select the **Applications with active users** view and click the **CONFIGURE** tab.
- Click the **+** symbol in the top right corner
- Replace "Visualization 2" with **Lifecycle Kanban**
- Select the visualization type **Kanban**

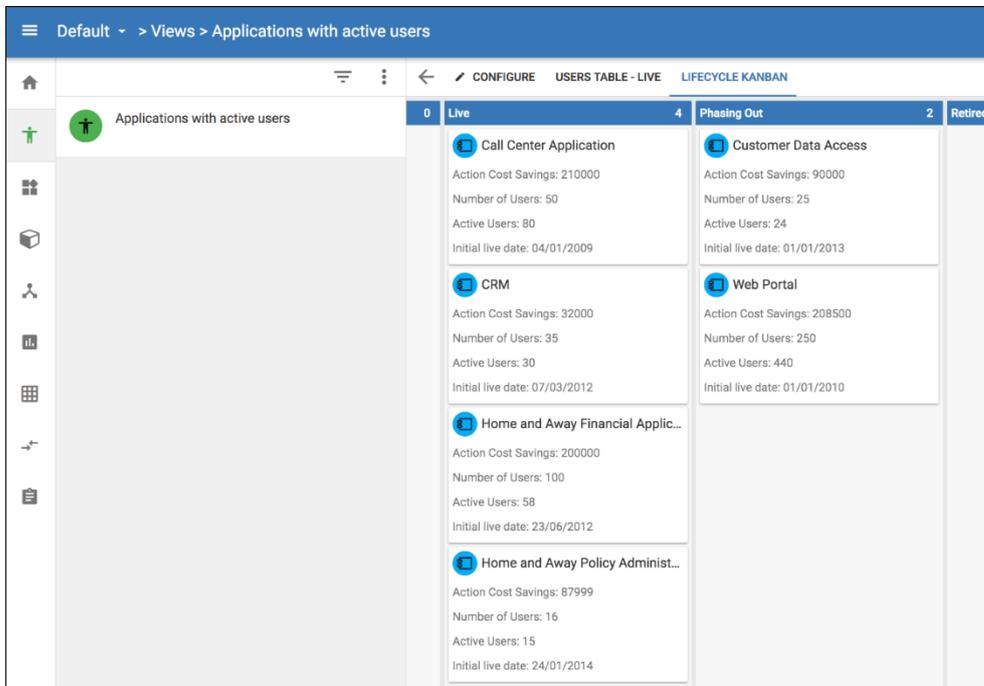
The screenshot shows the configuration interface for a Kanban visualization. At the top, the view is titled "USERS TABLE - LIVE" and "LIFECYCLE KANBAN". Below the title, there is a "Name" field containing "Lifecycle Kanban". A navigation bar shows four options: "Table", "List", "Kanban" (which is selected and highlighted with a blue border), and "Roadmap". The "Configuration" section includes an "Object Type" dropdown set to "Application component" and a "Stage Attribute" dropdown set to "Lifecycle status". Below this, there is a "Minimal Display Attributes" section with a "Filter" button and a list of attributes. The first attribute, "Application compon...", is checked with a blue box. The list also includes "Initial live date" and "Date when the first relea...". At the bottom, there is a "Preview" button and pagination information: "Items per page: 5" and "1-1 of 1 (filtered from 60)".

In the Configuration section, the object type to be used in the kanban can be selected, along with the attribute that will define the stages of the kanban.

**i** | Kanbans are currently limited to a single object type.

- Set the Object Type to **Application component** (this is the only type available in the view).
- Select the Stage Attribute to be **Lifecycle Status**.
- Check the following attributes to display on the kanban 'cards';
  - Action Cost Savings
  - Number of Users
  - Active Users
  - Initial live date

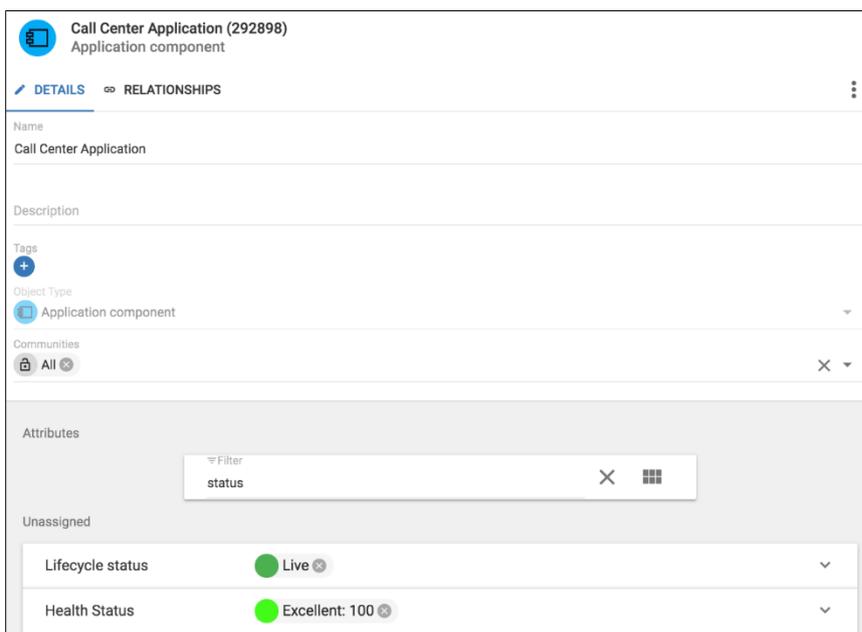
- **SAVE** the new visualization.
- Click the **LIFECYCLE KANBAN** tab to view the kanban.



The kanban board displays all 6 Application components, in either the **Live** or **Phasing Out** stages that represents the value of their **Lifecycle status** attribute.

**i** Note that the requested Display Attributes are shown on each kanban card.

- Open the **Call Center Application** object's edit window, by clicking on its name.
- Locate the **Lifecycle status** attribute to see that it is **Live**.



- 
- Click away from the edit window to close it.



If the object's card is dragged to a different stage of the kanban board, the Lifecycle status will be changed. If the Lifecycle status is changed while editing the object, the object's card will be displayed in the appropriate stage of the kanban board.

Kanban boards could be designed to visualize; a roadmap where stages are timeframes, training progress for Business actors, or track objects between the TOGAF ADM lifecycle stages.

---

# Roadmap Visualization

Roadmaps provide a mechanism to show an object over a time period. Most Enterprise Architecture (EA) objects will have a time dimension. Where objects don't have a time attribute, you can add them through the customization capabilities.

## The importance of roadmaps for Agile EA

The focus of Agile EA is on increasing the agility of the business. This means that it is essential that an organization understands where it will be at any given period of time. The enterprise architecture objects provide a blueprint of the organization. Many aspects of these objects can be described with a time dimension. The time dimension can be used to either display a milestone date at which something is expected to happen, or a date range within which something will take place.

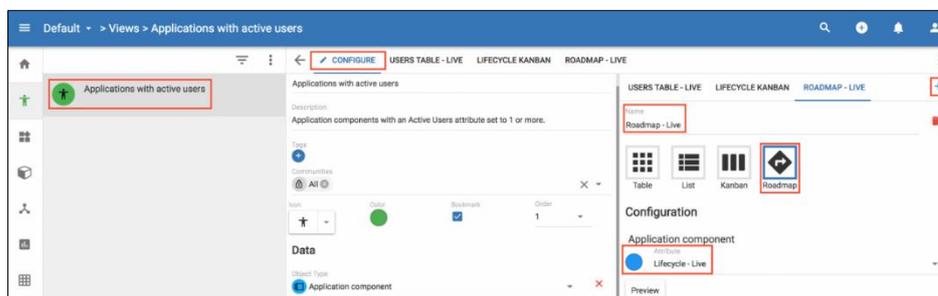
As 'time to market' and the ability to change quickly is vital for organizations to keep pace and innovate, then roadmaps become a critical view on the complexity and impact of change.

Once you have performed your business analysis, you can start to understand the tasks that will need to be completed to achieve any goals and the requirements that realize them. Something to consider is the organization's 'application portfolio', and how it will change in the short term, and longer term plans.

You would not want to begin critical projects on applications that will not be supported, or may be phased out in 2 or 3 years time, for example.

You need to understand the applications' time span, and this can be achieved by creating some simple roadmaps.

- Select the **Applications with active users** view, and click the **CONFIGURE** tab.
  - Add a new visualization, using the '+' symbol.
  - Replace "Visualization 3" with the name **Roadmap - Live**.
  - Select the visualization type **Roadmap**.
  - Click on the colored circle to modify the **color** used by the 'roadmap bars'.
  - Select the roadmap attribute to be **Lifecycle - Live**.



Each object type added to the view, could have its own color and attribute to represent objects on the roadmap.

- **SAVE** the visualization.
- Click the **ROADMAP - LIVE** tab to view the roadmap.



The roadmap visualization shows a red vertical line representing today's date, and colored bars representing the **Lifecycle - Live** date ranges of each Application component.

Clicking the arrow at the end of the colored bar will open the object's details.

The scale and extent of the roadmap can be changed, by dragging, pinching, or scrolling - depending upon the device being used.

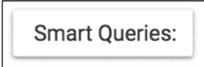
**i** Only objects that have dates that are between the start and end dates of the roadmap, will be represented.

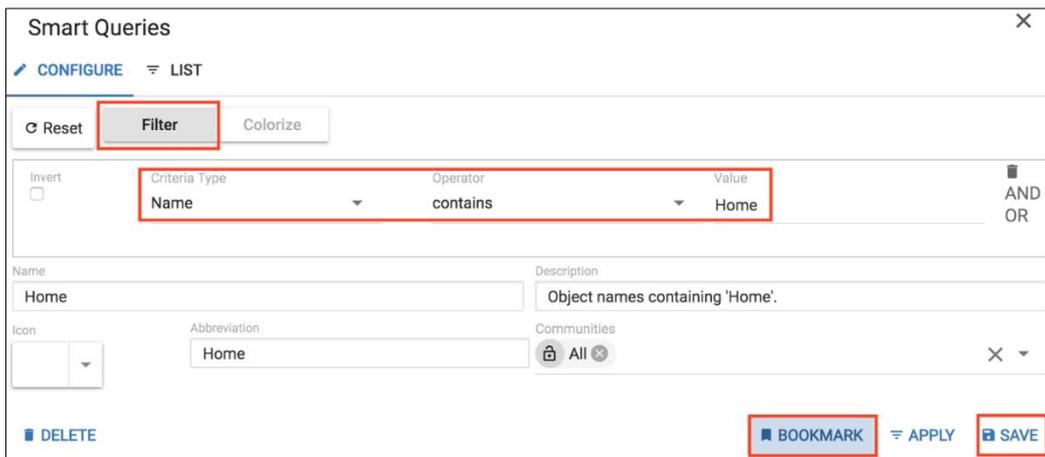
**!** Clicking on a colored bar will enable it to be dragged to a different location - this will automatically modify the date or date range attribute without requiring a save.

---

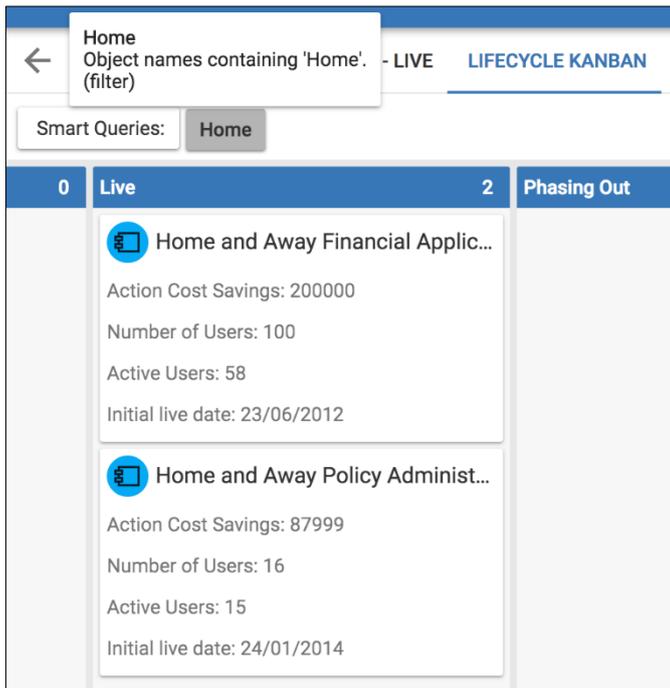
# Smart Queries

Smart queries allow users to **filter** or **color** views, to identify objects that meet specified search criteria.

- Select the **Action Menu** button  in the bottom right hand corner.
- Then select the **Smart Queries** button  from the Action menu.
- Open the smart query editor, by clicking the **Smart Queries** button  (located above the kanban).
- In the window that opens update the query fields as shown below:
  - Query type is **Filter**
  - **Criteria:** Name
  - **Comparator:** contains
  - **Value:** Home
- Describe the Smart Query:
  - **Name:** Home
  - **Abbreviation:** Home
  - **Description:** Object names containing 'Home'.
- Click **SAVE**.



- Click the **BOOKMARK** button, that appears after saving, to display the query as a button on the 'query ribbon'.
- Exit the Smart Queries window by clicking away from the window, or the X in the top right of the window.
- View the **LIFECYCLE KANBAN** visualization, and click the **Home Smart Query** button.



The kanban board will change to display just the 2 Application components that contain 'Home' in their name.

The **Smart Query** can be applied or removed, by clicking the bookmarked **Home** button.

**i** | Smart Queries are available on Views, Charts, Kanban Boards, Roadmaps, Pivot Tables, and Diagrams.

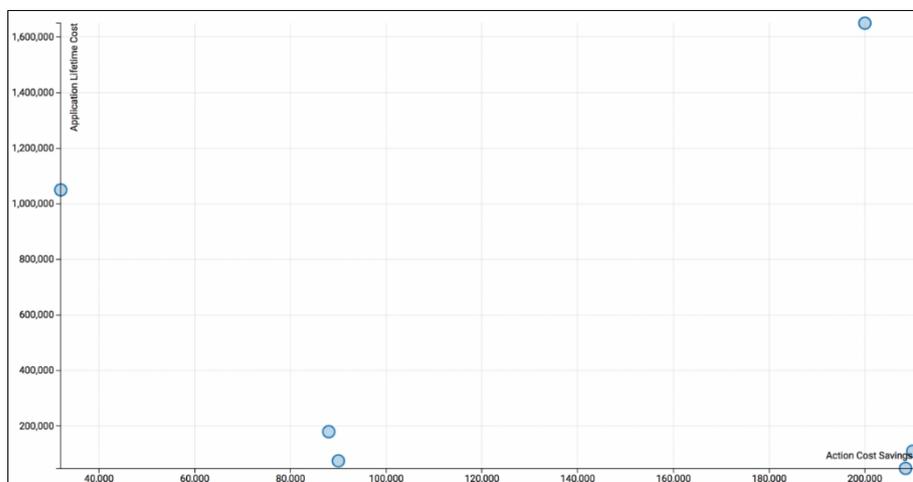
**i** | **Colorize** Smart Queries will highlight objects by using a color. Selection criteria can make use of an object's relationships, in addition to attribute values.

---

# Charts Revisited

The Stacked Bar and Word Cloud charts have been discussed earlier in the tutorial. A Scatter chart is to be used to visualize some costs of Application components.

- Create a new chart of type **Scatter** with the name **Application Costs and Savings**, and enter the following details;
  - **Description:** Application component costs and savings based on managing the application portfolio.
  - **Object Type:** Application component
  - **X:** Action Cost Savings
  - **Y:** Application Lifetime Cost
  - In the **Data** section;
    - Open the **Search for Objects** by clicking on the magnifying glass.
    - Click the **Search** button to list all Application components.
    - Click **ADD** to add all selected Application components to the dataset.
- **SAVE** the chart, then view it by clicking the **CHART** tab.



The scale of each axis is determined by the smallest and largest values for each attribute.

Hovering over any object will identify its name and the values along each axis.

Clicking the object will open its details window.

The purpose of a scatter chart is to show the possibility of a correlation between the two selected attributes. With such a small sample of six Application components, it is difficult to draw any conclusions. A straight line connecting the objects would suggest a high probability that the attribute values were somehow related.

# Pivot Table revisited

A pivot table has already been created to analyse Requirements. A pivot table is to be used to assess Application Lifetime costs.

- Create a new pivot table with the name **Application Lifetime Cost**, and enter the following details;
  - **Description:** View Application cost by their standards class and by Lifecycle Status.
  - **Type:** Application Component
  - Add all Application components to the **Dataset**.
  - **SAVE** the pivot table, then view it by clicking the **PIVOT TABLE** tab.
  - Drag the **Lifecycle status** attribute into the top row of the table.

Application components	Lifecycle status		Totals
	Live	Phasing Out	
Call Center Application	1		1
CRM	1		1
Customer Data Access		1	1
Home and Away Financial Application	1		1
Home and Away Policy Administration	1		1
Web Portal		1	1
<b>Totals</b>	<b>4</b>	<b>2</b>	<b>6</b>

- Replace **Count** with **Integer Sum**.
- Select the **Application Lifetime Cost** attribute from the new drop down list that appears.

Application components	Lifecycle status		Totals
	Live	Phasing Out	
Call Center Application	110,000		110,000
CRM	1,050,000		1,050,000
Customer Data Access		75,000	75,000
Home and Away Financial Application	1,500,000		1,500,000
Home and Away Policy Administration	180,000		180,000
Web Portal		48,000	48,000
<b>Totals</b>	<b>2,840,000</b>	<b>123,000</b>	<b>2,963,000</b>

- Replace **Table** with **Heatmap** option.

Application components	Lifecycle status		
	Live	Phasing Out	Totals
Call Center Application	110,000		110,000
CRM	1,050,000		1,050,000
Customer Data Access		75,000	75,000
Home and Away Financial Application	1,500,000		1,500,000
Home and Away Policy Administration	180,000		180,000
Web Portal		48,000	48,000
<b>Totals</b>	<b>2,840,000</b>	<b>123,000</b>	<b>2,963,000</b>

**SAVE** the pivot table.

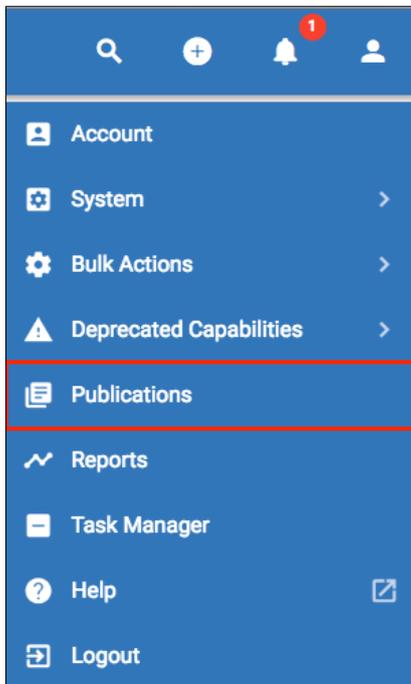
You have very quickly created an application landscape on lifetime costs by lifecycle status.

---

# Publications

Publications can be created on the platform.

- Select the **User** menu > **Publications** option.



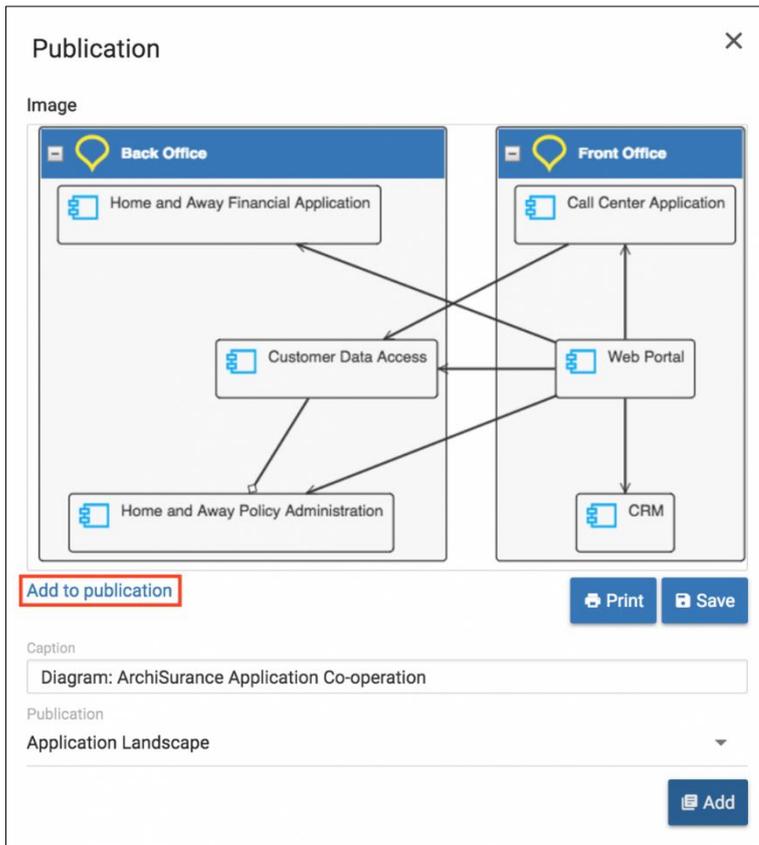
- **ADD** a new publication .
- Enter **Application Landscape** as the name of the publication.



- **SAVE** the publication.
- Open the diagram **ArchiSurance Application Co-operation**.

- Click the Action menu in the bottom right of the screen .

- Click the Publications option .



The **Print** button will allow a PDF version of the diagram to be saved, while the **Save** button will save a PNG version of the diagram.

- Click **Add to publication**, and accept the default Caption.
- Select the **Application Landscape** publication.
- Click **Add**, to append the diagram to the publication.
- Open the **Applications with active users** view, and its **Users Table - Live** visualization.
- Add this table to the Application Landscape publication.
- Return to the **Application Landscape** publication.
- Click on the **CONFIGURE** tab.

Images of the diagram and table have been inserted into the publication. The publication can be modified using the features from the rich text area's toolbar;



**Example Publication created in the Agile EA Tutorial**

**Diagram: ArchiSurance Application Co-operation**

**Users Table - Live**

Name	Number of Users	Active Users	Initial live date	Lifecycle status	Lifecycle - Live
Call Center Application	50	80	04/01/2009	Live	04/01/2009 > 30/09/2015
CRM	35	30	07/03/2012	Live	07/03/2012 > 31/12/2019
Customer Data Access	25	24	01/01/2013	Phasing Out	01/01/2013 > 31/12/2017
Home and Away Financial Application	100	58	23/06/2012	Live	23/06/2012 > 31/12/2018
Home and Away Policy Administration	16	15	24/01/2014	Live	24/01/2014 > 31/12/2024
Web Portal	250	440	01/01/2010	Phasing Out	

Items per page: 10 1-6 of 6

**View: Applications with active users, Visualization: Users Table - Live**

- **SAVE** the publication.

That completes this tutorial. You have now begun your journey to using the Agile Enterprise Architecture platform and have just touched the surface.

Now go ahead and add your own objects and views into the platform and share and collaborate with your colleagues.

---

# Customization Tutorial

**Role needed to use:** [Client Administrator](#)

 This tutorial is currently being updated to match the changes in the attribute UI.

**Important: Where you see a ⇒ symbol, there is a task for you to perform.**

  Please allow 45 minutes to complete this tutorial.

In this tutorial we will look at customization of the metamodel, working through:

- Creating Object Types
- Creating Relationships between Object Types
- Adding Attributes to Object Types
- Creating instances of Object Types

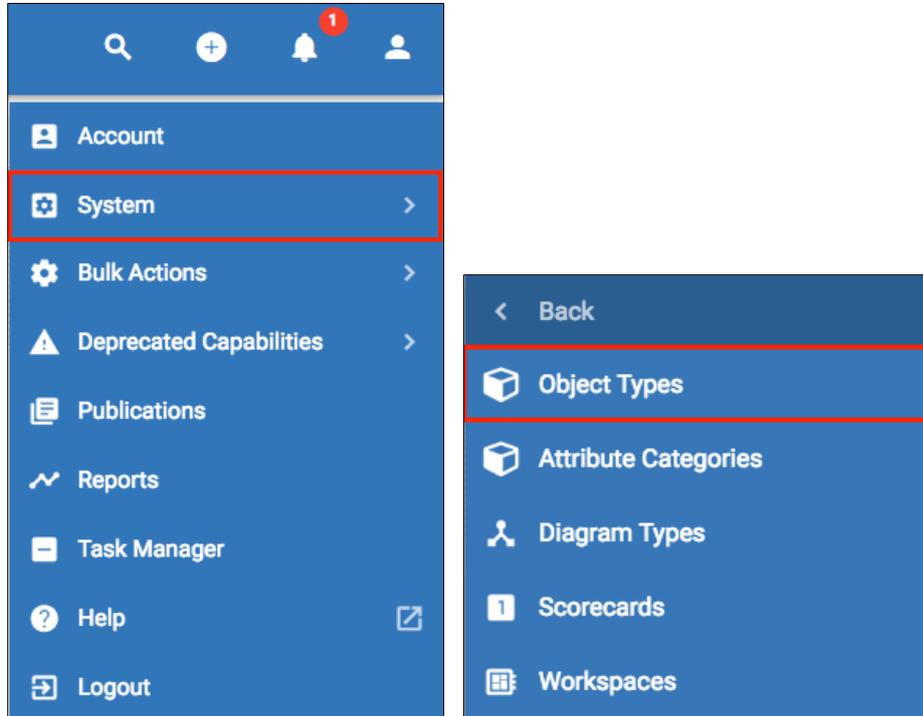
In order to remain neutral of any methodology, we'll be basing our examples on Airlines, Aircraft and Pilots.

If you would like to understand what a metamodel is, See [metamodel](#) help.

---

# Creating Object Types

Select the **User** menu > **System** > **Object Types** option



## Creating an Airline

Click the **add** button , on the bottom right hand corner of the list.

Enter the following object type details;

- Name: **Airline**
- Description: **An airline is a company that provides air transport services for traveling passengers and freight.**
- Icon: **Select an icon**
- Color: **Select a color**

The form has two columns. The left column contains: a 'Name' field with 'Airline' entered; an 'Icon' field with a globe icon selected; and a 'Color' field with a blue circle selected. The right column contains a 'Description' field with the text 'An airline is a company that provides air transport services for traveling passengers and freight.'

Click **SAVE** to create the new Airline object type.

## Creating an Aircraft

Click the **add** button again, and enter the following object type details;

- 
- Name: **Aircraft**
  - Description: **An aircraft is a machine that is able to fly by gaining support from the air.**
  - Icon/Color: **Select an icon and color.**

Name <b>Aircraft</b>		Description An aircraft is a machine that is able to fly by gaining support from the air.
Icon 	Color 	

- Click **SAVE**.

## Creating a Pilot

Create a new object type using the following details;

- Name: **Pilot**
- Description: **A person authorized to fly aircraft.**

Name <b>Pilot</b>		Description A person authorized to fly aircraft.
Icon 	Color 	

- Click **Save**.

---

# Creating Relationships between Object Types

Whilst in the Pilot object type, select the **RELATIONSHIPS** tab  (this will appear after saving the new object type).

**i** When an object type is created, it is initialized with **associated with** relationships to all the other object types.

## Pilot flies Aircraft

Click the **add** button  to create a new relationship for the Pilot object type.

Enter the following details

- From Name: **flies**
- To Name: **flown by**
- Object Type: **Aircraft**



- Click **SAVE**.

A new relationship now exists between Pilot and Aircraft. A Pilot flies an Aircraft, and an Aircraft is flown by a Pilot.

## Pilot employed by Airline

Add a new **employed by** relationship from a Pilot to an **Airline**. The Airline **employs** the Pilot.

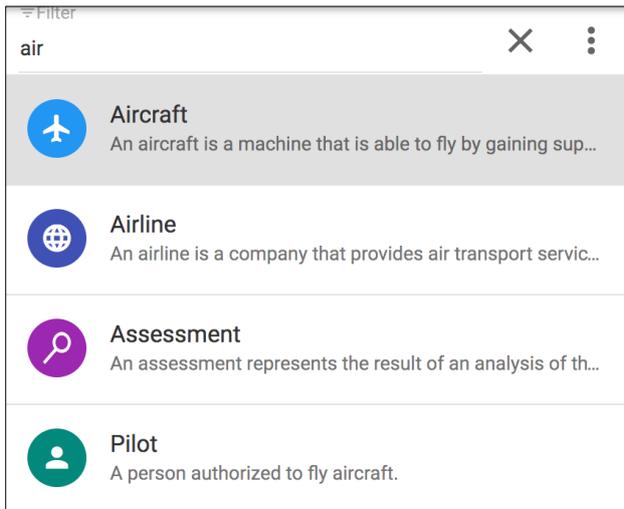


- Click **SAVE**.

Another new relationship now exists between Pilot and Airline. A Pilot is employed by an Airline, and an Airline employs a Pilot.

## Aircraft owned by Airline

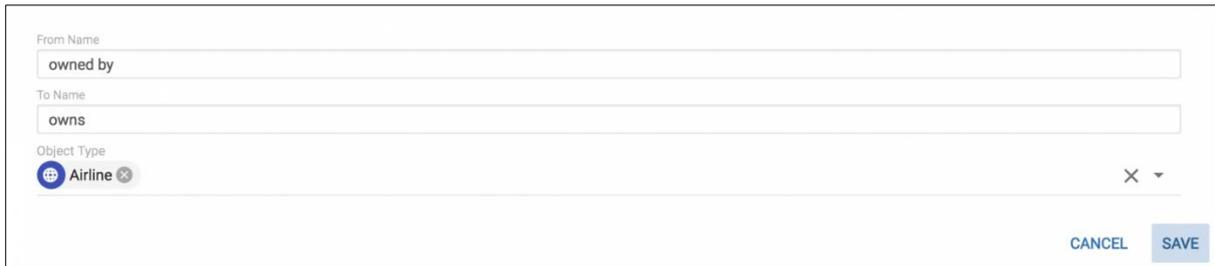
From the **Object Type List** on the left, locate and click on the **Aircraft**.



Click on the **RELATIONSHIPS** tab, and scroll down to view the **flown by** relationship to the Pilot.

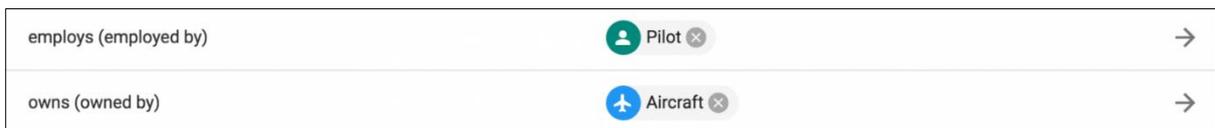


Add an **owned by** relationship from the Aircraft to the Airline. The Airline owns the Aircraft.



Click **SAVE**.

Open the **Airline** object type, and view the newly created relationships in the **RELATIONSHIPS** tab.



---

# Adding Attributes to Object Types

Attributes are required by Object Types to store specific detail about the objects. If no attributes are added to an object type, objects will still have a Description.

**i** For more information on the rich set of attribute options available to each of the attribute types, go to the [customizable metamodel](#) section.

Return to the **Aircraft** object type, and view the **CONFIGURE** tab.

Click the Can't find your attribute? Click here to create a new one. option, to append an attribute to the object type.

Select the **Attribute Type** drop down and choose **'Array'** from the list.



Once the Attribute Type has been selected, properties for the attribute type are presented. Enter the following details;

- Name: **Range**
- Description: **What is the range of the aircraft?**
- Key 1: **Short Haul**
- Value 1: **1000**

Select the arrow button next to the Attributes type drop down (Array), to reveal the 'rules'.

Check the **Single Select [ss]** option, to restrict the attribute value to a single entry.

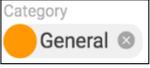
Click **ADD NEW VALUE**, to add a new key/value entry to the Array.

Enter; Key 2: **Long Haul** Value 2: **5000**

Click **SAVE**.

The rule associated with the **Range** attribute will ensure that users can only enter either **Long Haul** or **Short Haul**.

Attributes can be grouped by associating them with a Category.

Click the blue + circle associated with the Category, and enter the value General .

Click **SAVE**.

Let's take a look at how this attribute is used in objects.

---

# Create instances of Object Types

Click the **All Objects** module from the menu , and then the **add** button , to add a new object.

Fill in the instance details as below, selecting the **Aircraft** object type.

- Name: **F11**
- Description: **A dutch airliner.**

Name
<b>F11</b>
Description
<b>A dutch airliner.</b>
Tags

Object Type
 <b>Aircraft</b>
Communities
 <b>All</b> 

- **SAVE** the object.

Add another **Aircraft** type using the following detail;

- Name: **767**
- Description: **A commercial wide body airliner.**

Name
<b>767</b>
Description
<b>A commercial wide body airliner.</b>
Tags

Object Type
 <b>Aircraft</b>
Communities
 <b>All</b> 

- **SAVE** the object.

Add another object of type **Airline** using the following detail;

- Name: **FlyMe Airways**
- Description: **Provides commercial flights within Asia/Pacific.**

Name  
**FlyMe Airways**

---

Description  
**Provides commercial flights within Asia/Pacific.**

---

Tags  
+

Object Type  
Airline

---

Communities  
All

- **SAVE** the object.

Open the **RELATIONSHIPS** tab.

Click the **Create the first Relationship** button, to add a new relationship.

- Select **owns (owned by)** for the Relationship Type.
- Select **Aircraft** for the Object Type.
- Check both the **767** and **F11** Aircraft objects.

Relationship Type  
**owns (owned by)**

---

Object Type  
**Aircraft**

---

Airline: FlyMe Airways owns

<input checked="" type="checkbox"/>	Name	Type
<input checked="" type="checkbox"/>	767	Aircraft
<input checked="" type="checkbox"/>	F11	Aircraft

Items per page: 5    1-2 of 2

**CANCEL** **SAVE**

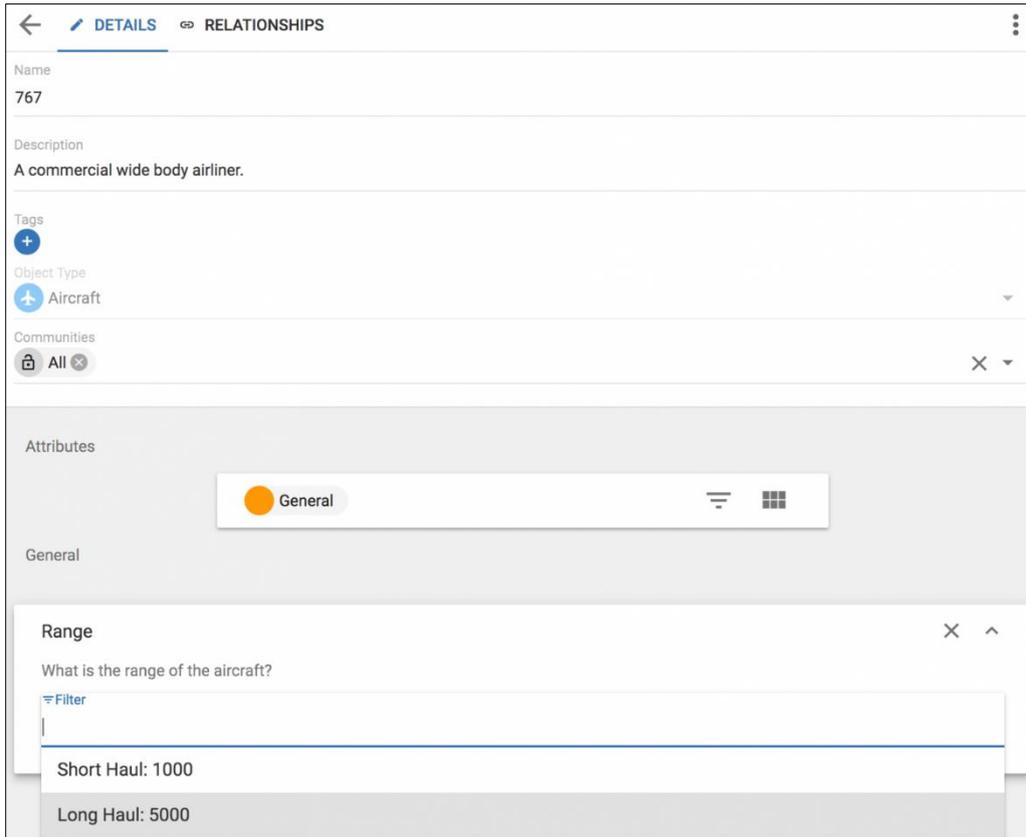
- **SAVE** the relationships.

Hover over **767** in the relationship list and select **NAVIGATE**.



View to the **DETAILS** tab, and locate the **Range** attribute in the General category.

Click on the attribute to expand it, and select **Long Haul: 5000**.



**i** Note that the 'single select [ss]' rule set for the Range array attribute, will only allow the selection of one value..

**SAVE** the change.

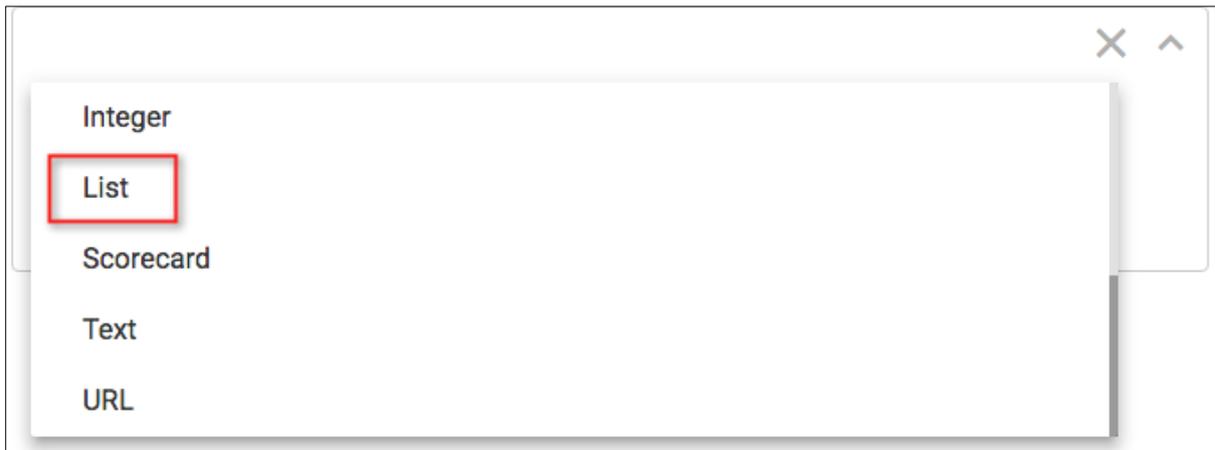
## **Adding Attributes revisited**

Select the **User** menu > **System** > **Object Types** option, to make further changes to the metamodel.

Select the **Aircraft** object.

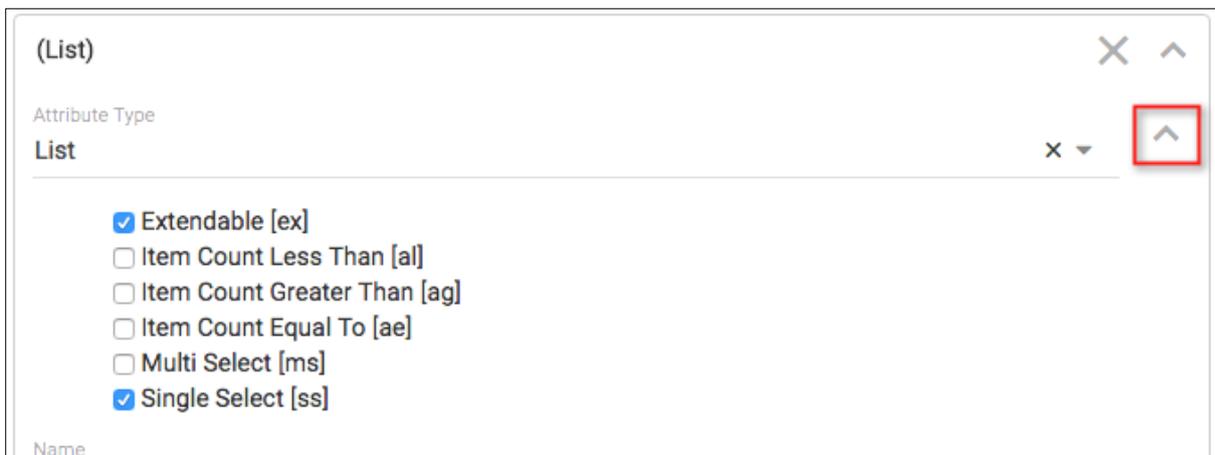
Click the Can't find your attribute? Click here to create a new one. to add a new Attribute to the Aircraft object type.

Click the **Attribute Type** drop down and select **List**.



**i** | A **list** attribute type will present a dropdown of preset values to the user.

Select the arrow button next to the Attribute Type drop down.



This will reveal the rules that can be set for the selected attribute type.

Check the '**Extendable [ex]**' and the **Single Select [ss]** options.

Select the arrow button next to the Attribute Type drop down again, to close the rules.

**i** | The **Extendable [ex]** rule will allow users to enter their own values and add to the list.

Fill out the attribute details;

- Name: **Manufacturer**
- Description: **Who manufactures this aircraft?**
- Value 1: **Boeing**  
Select an icon by clicking the dropdown that precedes the name.

---

Also set a 'Heatcolor' by clicking the dropdown that follows the name - check the **Set As Heatcolor [hc]**, and select a color.

- Click + **ADD NEW VALUE** to enter multiple values.
- Value 2: **Airbus**  
Select an icon.
- Value 3: **Lockheed**  
Set a Heatcolor.
- Value 4: **McDonnell Douglas**

The screenshot shows a configuration window for a 'List' attribute named 'Manufacturer'. The description is 'Who manufactures this aircraft?'. There are four values defined: Boeing, Airbus, Lockheed, and McDonnell Douglas. The category is set to 'General'.

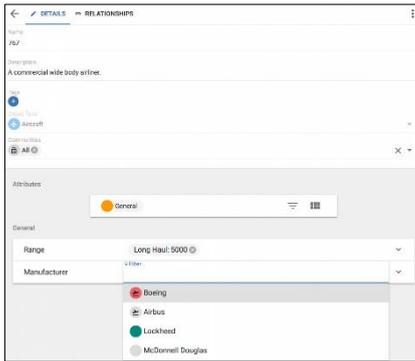
Set the Category value to **General**.

**SAVE** the object type, to add the new List attribute **Manufacturer**.

## Updating Instances

The name of the Manufacturer of an Aircraft can now be identified for each of the Aircraft.

- Click the **All Objects** module.
- Select the **'767'** object.
- View the **Details** tab.
- Select the **Boeing** option from the Manufacturer list.



- **SAVE** the object.

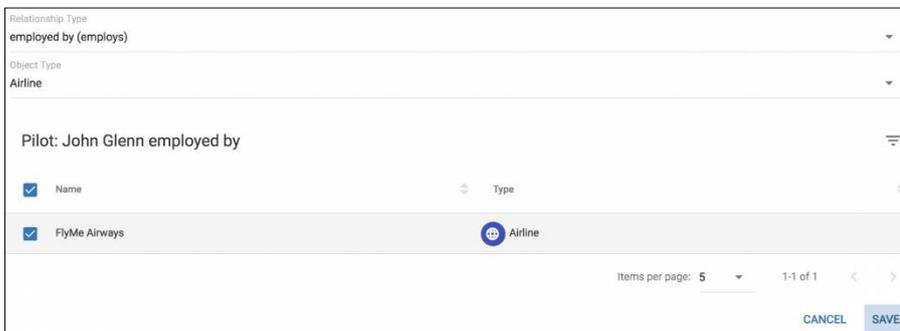
## FlyMe Airways Pilot

Create a new object of type **Pilot** with the following detail;

- Name: **John Glenn**
- Description: **Commercial pilot.**



- **SAVE** the object, and create a new relationship in the **RELATIONSHIP** tab;
  - Relationship Type: **employed by (employs)**
  - Object Type: **Airline**
  - Check **FlyMe Airways**



---

In this tutorial we have;

- Created new object types.
- Created Relationships between them.
- Added different attribute types, with rules.
- Created instances of the new object types, created relationships between them, and stored information in their attributes.

---

# Migration

## Open Group File Format (OGFF) to SPP - online exchange

Role Applicable to: [Client Administrator](#)

Migrate data from an ArchiMate source tool (**ArchiMate plug-in for Unicom System Architect, Archi and BizDesign tools**) into the Comma Separated Value (csv) format for import into the erwin Strategic Planning Platform (SPP), using our online exchange tool.

**i** Before you begin please make sure you have your data in an **Open Group File Format** xml file - it is also recommended that you already have your SPP login area prepared prior to beginning this process.

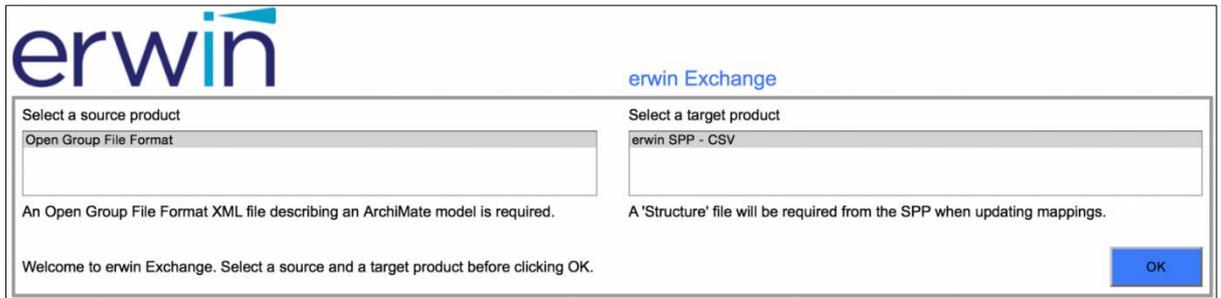
This example takes you through a standard migration where there has been no customization of the metamodel in the source tool or Agile EA... (although customizations can be handled). The output will be a set of CSV files that can be imported into the Agile EA platform.

**!** We assume that your metamodels in the source tool and the SPP have been matched previously to using this data conversion, if any customization has taken place, then please ensure you also extend the mapping file with these customizations. If you want to find out about your metamodel customizations within System Architect, then there is a macro that can be used [here](#).

---

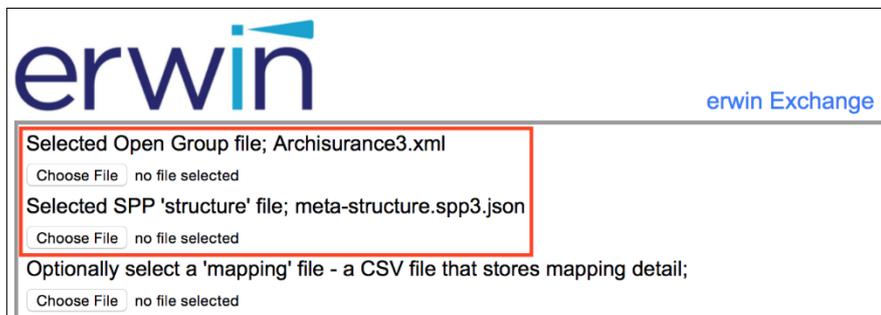
# Input, Mapping and Output

1. Navigate to <http://dev.corsocloud.com:82/erwinexchange/>
2. Select the 'Open Group File Format' in the left hand box and 'erwin SPP - CSV' in the right hand box, then click **OK**.



The screenshot shows the 'erwin Exchange' interface. At the top left is the 'erwin' logo. Below it, there are two columns for product selection. The left column is titled 'Select a source product' and contains a dropdown menu with 'Open Group File Format' selected. Below this dropdown is the text: 'An Open Group File Format XML file describing an ArchiMate model is required.' The right column is titled 'Select a target product' and contains a dropdown menu with 'erwin SPP - CSV' selected. Below this dropdown is the text: 'A 'Structure' file will be required from the SPP when updating mappings.' At the bottom left, there is a welcome message: 'Welcome to erwin Exchange. Select a source and a target product before clicking OK.' At the bottom right, there is a blue 'OK' button.

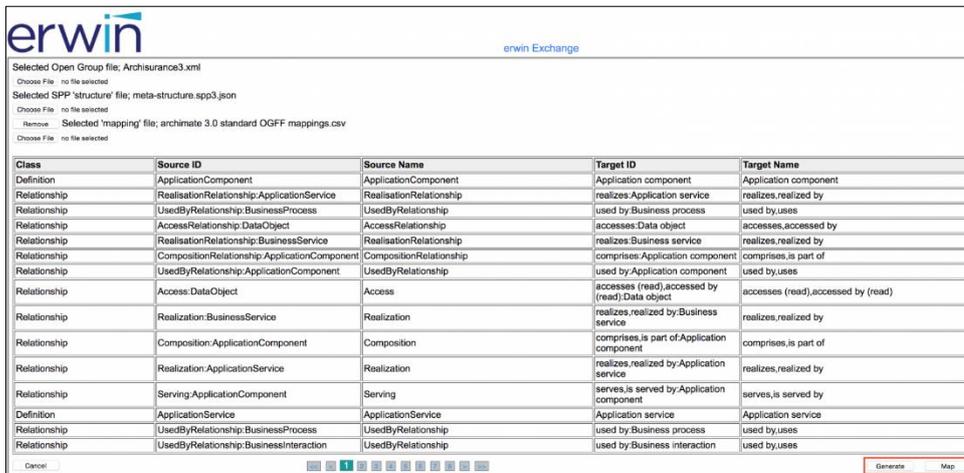
3. On the next page you will upload some files.
  - Click the **Choose File** button and select your (Open Group File Format) XML file
  - Upload an SPP structure file
    - If you haven't modified your ArchiMate metamodel at all within the SPP platform, this file can be downloaded [here](#).
  - Extract the .spp file and upload it using the **Choose File** button
    - The content of the SPP structure can be created using the following URL; **<https://myerwin.io/ea/api/meta-structure>**



The screenshot shows the 'erwin Exchange' interface with a red box highlighting the file selection status. The status is as follows: 'Selected Open Group file; Archisurance3.xml' with a 'Choose File' button and 'no file selected' text below it. 'Selected SPP 'structure' file; meta-structure.spp3.json' with a 'Choose File' button and 'no file selected' text below it. Below these, there is a section for a mapping file: 'Optionally select a 'mapping' file - a CSV file that stores mapping detail;' with a 'Choose File' button and 'no file selected' text below it.

- Upload a mapping file - **the example assumes that an existing mapping file is to be used - otherwise the mapping needs to be performed.**
    - If you haven't created a mapping file, a 'base file' file can be downloaded [here](#). (This file does not contain all relationships, and no Properties.)
4. The **Check** button in the bottom right , can be clicked at any time to validate the uploaded files - red text will indicate an invalid file.

**i** | Note that the Check button will change to Map once the first two files are valid.



If a mapping file has been uploaded, there will be 2 options for output - Generate and Map.

5. To output all object type csv data files at once, select the **Generate** button.

- A list of all available data files to be output will be presented to you - uncheck the checkboxes of the files you do not want to output



- Select the **Download** button

A **zip** file containing the **csv** files for import will be downloaded to your machine.

**i** Note that the Generate button will only be active if a mapping file has been uploaded. The Download button will appear after generation of the files has occurred.

6. If you would like to output a particular object type, select the **Map** option, instead of Generate.

- Click on the particular object type row in the left-hand table and select the **'Create'** option. An option for a **'Data File'** will then be seen, select this to download your object type CSV for the currently selected object type. You will need to repeat this step for each object type you would like to output.

erwin Exchange

Select from the available Items list, adding them to the list below;

Type ID	Type Name	Mapped ID	Mapped Name	Type ID	Type Name
ApplicationComponent	ApplicationComponent	Application component	Application component	<input type="radio"/>	BPMN Event
ApplicationFunction	ApplicationFunction	Application function	Application function	<input type="radio"/>	BPMN Gateway
ApplicationService	ApplicationService	Application service	Application service	<input type="radio"/>	BPMN Lane
BusinessActor	BusinessActor	Business actor	Business actor	<input checked="" type="radio"/>	BPMN Pool
BusinessCollaboration	BusinessCollaboration	Business collaboration	Business collaboration	<input type="radio"/>	BPMN Task
BusinessEvent	BusinessEvent	Business event	Business event	<input type="radio"/>	Business actor
BusinessFunction	BusinessFunction	Business function	Business function	<input type="radio"/>	Business collaboration
BusinessInteraction	BusinessInteraction	Business interaction	Business interaction	<input type="radio"/>	Business event
BusinessInterface	BusinessInterface	Business interface	Business interface	<input type="radio"/>	Business function
BusinessObject	BusinessObject	Business object	Business object	<input type="radio"/>	Business interaction
BusinessProcess	BusinessProcess	Business process	Business process	<input type="radio"/>	Business interface
BusinessRole	BusinessRole	Business role	Business role	<input type="radio"/>	Business object
BusinessService	BusinessService	Business service	Business service	<input type="radio"/>	Business process
CommunicationNetwork	CommunicationNetwork	Communication network	Communication network	<input type="radio"/>	Business role
Contract	Contract	Contract	Contract	<input type="radio"/>	Business service

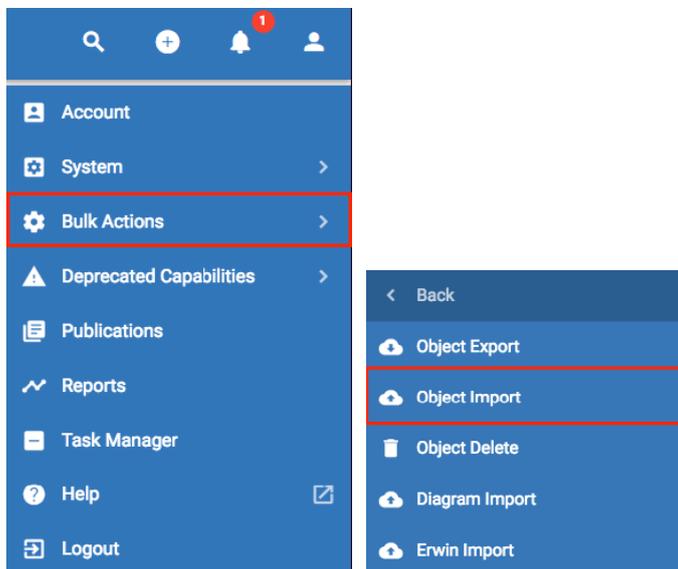
Available items;

Home Relations Properties Data File Create

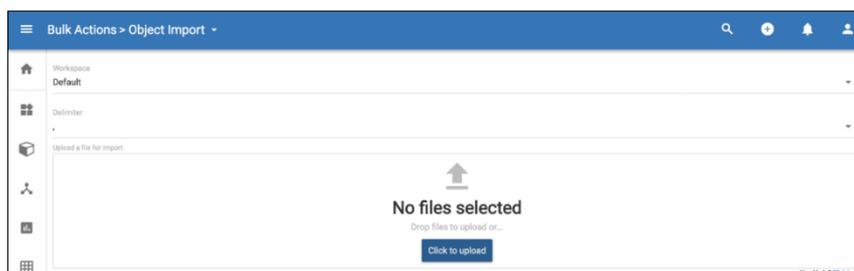
**i** Note that unless the Relationships and/or Properties have been mapped, the created Data File will only contain the mapped object, and not its relationships and/or properties.

# Run import of data into SPP

1. In Agile EA (as an administrator), open **User** menu > **Bulk Actions** > **Object Import**



2. Click on the **Click to upload** button and choose all of the csv files that you would like to import. (the csv files downloaded using the Generate option will need to be extracted from the zip file downloaded before they can be uploaded).



3. Press the **IMPORT** button in the bottom right hand corner.
4. After the import has run, click on the **Objects** view and choose one of your objects that has been imported. Notice that the relationships are present.

**i** The following sections will need to be used if you have customized the metamodel of either the source ArchiMate tool or the SPP client area.

---

## Re-Generate SPP Structure File (for customized SPP metamodels)

If you have modified your meta structure then please navigate to <https://myerwin.io/web/meta-structure> (you will need to log-in to your SPP client area) and save the text displayed as a .spp file by right clicking the page, selecting **Save As...** and then changing the file name to have .spp as the extension.

This will give you an up to date SPP structure file to upload.

# Add Customizations to the Mapping File (for customized source tool or SPP metamodels)

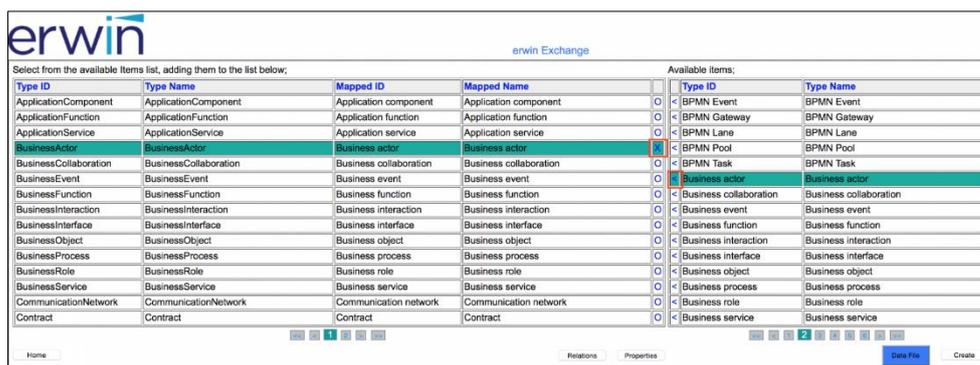
Your choices are to either upload an existing mapping file, or create a new one (csv format), by leaving the **Selected 'mapping' file** box blank.

From within the mapping editor, using the **Map** option, you can

## 1. Map object types

Select a row of the left hand table (source object types) - the row should be highlighted in green...Then select the corresponding row in the table on the right (SPP object types - named 'Available items'), by selecting the '<' for that row - the row will also highlight in green and you will see that the **Mapped ID** and **Mapped Name** columns of the left hand table are now filled with the mapping.

**i** Note: to remove a mapping, highlight the row (turns green) and then select the 'X' at the end of that row - the Mapped ID and Mapped Name columns will now be empty for that row.



## 2. Map properties and relations between the tool metamodels

Once a mapping entry exists in the left-hand table, you can map the properties and relationships for that object type...Selecting each object type and then using the **Relations** or **Properties** buttons at the bottom of the page will open a further mapping selection page for the two chosen object types.

**i** Once in the new page, relations and properties can be mapped in the same way as the object types in (1), selecting the **Return** button to move back to the object type mapping list.

## 3. Map customized object types, properties, and relations

Customizations from either source tool or SPP metamodels will be picked up from the OGFF input XML or the up-to-date .spp structure file...These will appear as new object type, relations or properties to be mapped where possible (as mentioned at the beginning of the page, we assume that the metamodels have been synched within both tools).

---

#### 4. Change any mappings that have been created and are outdated/incorrect (from an existing mapping file).

If you choose to upload one of our base mapping files (for non-customized environments), or an existing mapping file of yours, the existing mappings will appear in the table.

You can extend/change these, by selecting the row in the left-hand table and then selecting the '<' of the row to map to in the right-hand table. This will overwrite object type, relation, or property mappings.

Once your re-mapping is complete (you can map all object types, relationships, and properties at once), you can **Save** a mapping file (for future use), using the **'Mappings'** option. (This option will show once the original mappings have been changed).

Once this mapping files has been created, it can be used to output the data files needed, using the method above.

---

# System Architect to erwin EA Data Migration

**Role Applicable to:** [Client Administrator](#)

Migrate **model content** from System Architect into the correct format for import into the erwin EA tool. To do this we use our offline exchange tool, that allows us to pull the metamodel and chosen data out of System Architect and map it to the metamodel we have pulled out of erwin EA. The result is data in a format that is ready to import into the erwin EA tool.

The sections below show you how to migrate a non-customized and a customized System Architect metamodel into erwin EA.

---

## 2 Steps for importing models - non-customized System Architect metamodels

This example takes you through a migration with a non-customized metamodel in the source tool System Architect or the destination tool erwin EA... the output will be a set of CSV files (one per object type) that can be imported into the erwin EA tool.

1. Create mapping files and data outputs
2. Run import

### Create mapping files and data outputs

Within this section, you will

- Create an erwin EA Agile schema file
- Create a mapping file
- Create some explorer diagrams within System Architect, to extract the model information that you want
- Download the exchange tool
- Perform the mapping and output some CSV files

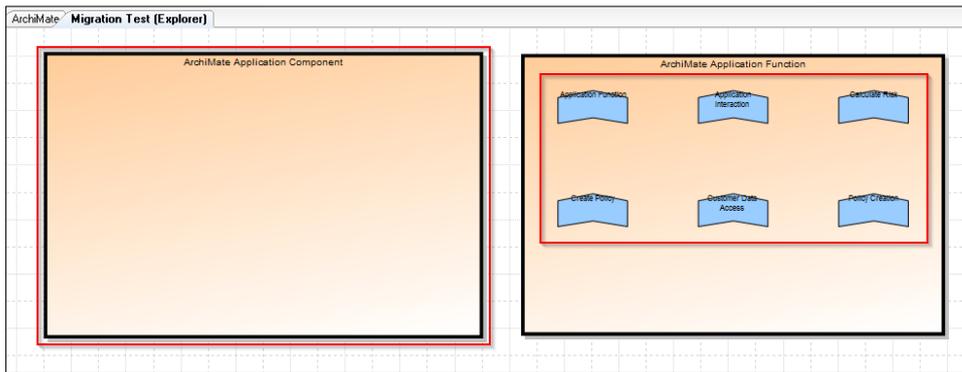
1. We need to let our exchange tool know the metamodel within our erwin EA client area - Now that both the meta models for System Architect and erwin EA match, you need to download the schema for your customized erwin EA environment.

Click on the following URL (you will need to login): <https://myerwin.io/ea/api/meta-structure>

**Select File > Save Page as.** and save the file with a .spp extension, such as **erwinEA-metaStructure.spp**

2. Create an **Explorer** diagram within System Architect. The diagram needs to contain either
  - the **specific definition instances** that you want to migrate to erwin EA.
  - the **collections** for each **definition type** that you want to migrate to erwin EA - The exchange tool will export all instances of a definition type for mapping if it finds an **empty collection** on the diagram
3. Save the diagram

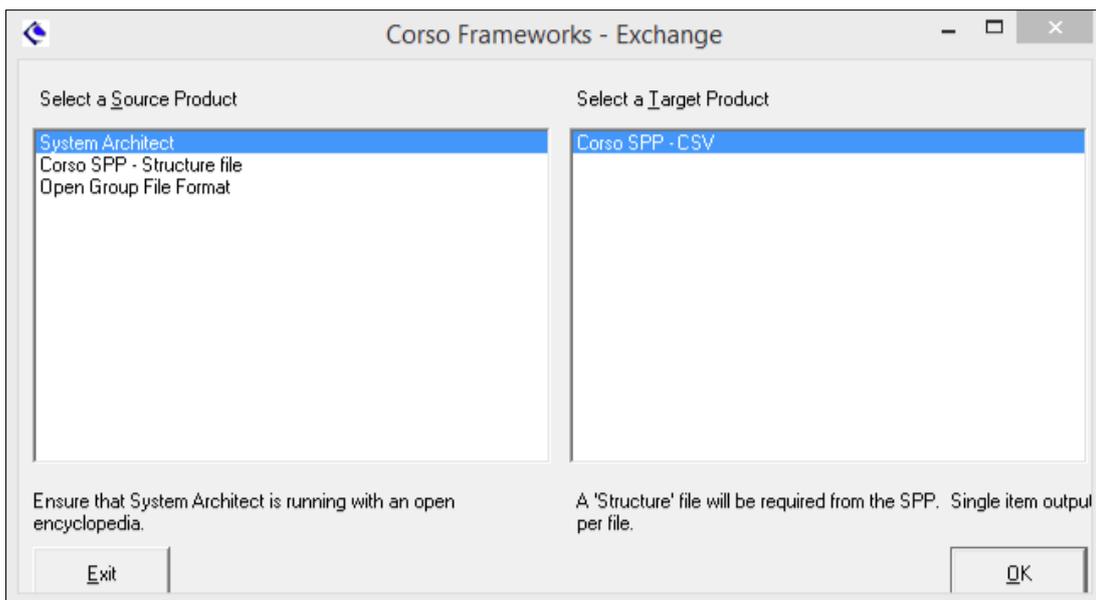
In the example below, we will transfer the specific definition instances of type Application Function and all definition instances of type Application Component within the workspace that we are in.



An example System Architect explorer diagram

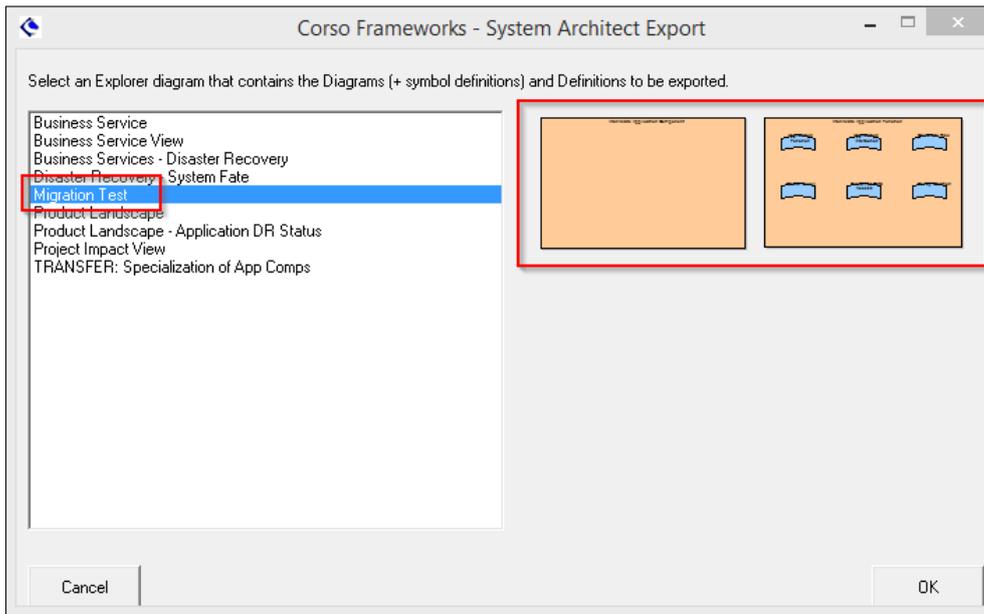
- Next download a copy of the **exchange** tool, extract it and start it on the Windows environment where your System Architect instance is located.

! IMPORTANT: A System Architect encyclopedia needs to be open for the exchange tool to extract data from System Architect.



erwin exchange tool

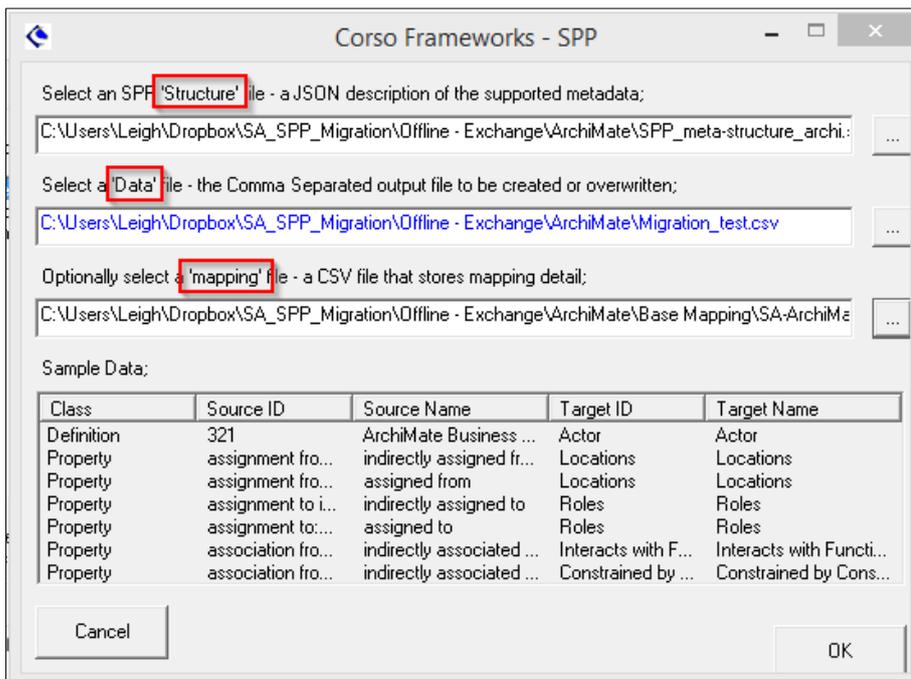
- Choose **System Architect** as the source product and **Corso SPP - CSV** as the destination product.
- Next, choose the explorer diagram you've just created (e.g. Migration Test) from the list available and press OK.



7. In the select an SPP '**Structure**' file box, select the .spp file downloaded at step 2.
8. Next, in the '**Data**' file box, select a location and filename (with .csv extension) for the output files to be generated to.
9. In the last field, '**mapping**', select a location and create a filename press **OK**.

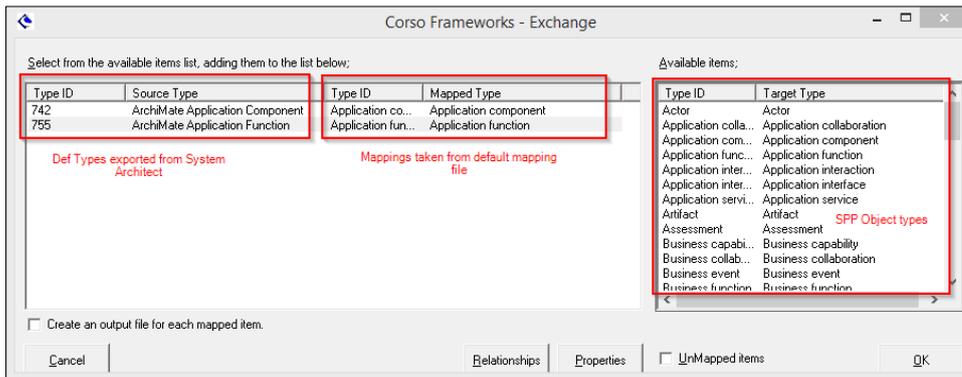
If you have an existing mapping file, you can enter this file location (black text denotes an existing file, blue will create new)

This file will store all the mappings between System Architect and erwin EA object types, attributes and relationships.



You will now see

- a list of object types that were selected within the System Architect explorer diagram (left field, left hand columns)
- mappings taken from the selected mapping file (left field, right hand columns)
- a list of objects that were picked up from the erwin EA structure file (right field)



### Mapped object types in exchange

Mapping is now complete.

Properties and Relationships are also mapped - these can be viewed if you select the **Relationships** or the **Properties** buttons.

10. Next, click on the checkbox '**Create an output file for each mapped item**'. Selecting this will output data files for all of the mapped object types in the list.

**i** Leaving this checkbox blank will require an object type to be selected from the list, before an output data file can be created for the selected type

11. Press **OK** and then **yes** when asked if you would like to continue. The mapping will now occur - a message will be displayed upon completion (this can take some time and progress can be monitored from a report that runs in the open System Architect session).
12. If you look in your designated output directory, you will now see files created for each of the object types and one containing all relationships (the mapping file is also created or updated).

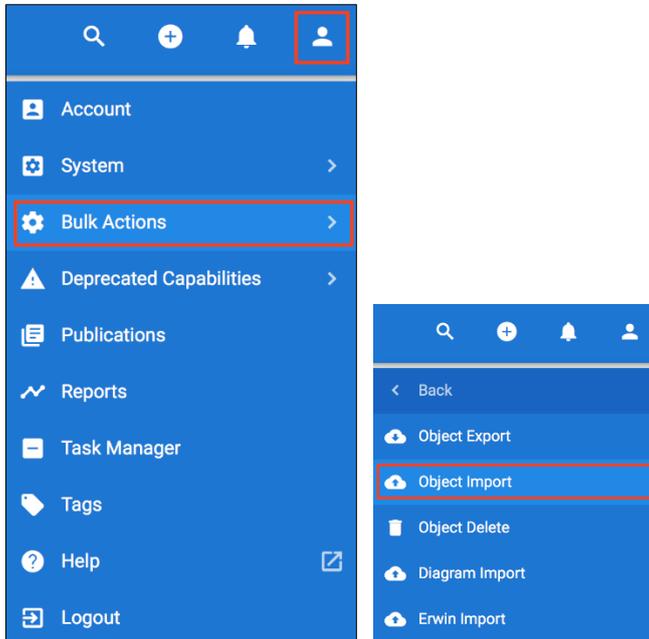
**i** If you selected to create multiple files, the data files will be appended with the System Architect definition type number

Open the files and perform a sanity check on the data, to ensure it is in the form you expect.

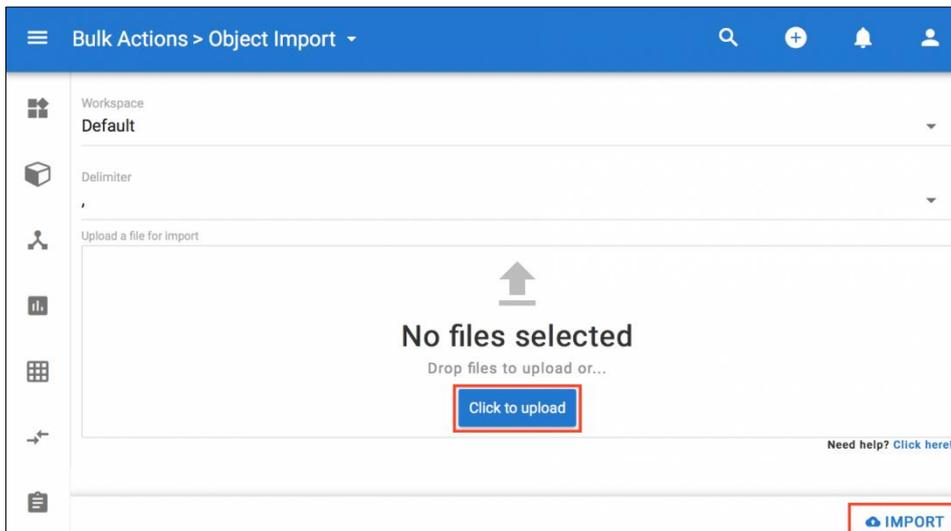
---

## Run import of data

1. In erwin EA (as an administrator), open **User > Bulk Actions > Object Import**.



2. Click on the **Click to upload** button and choose all of the csv files that you would like to import.



3. Press the **IMPORT** button in the bottom right hand corner.
4. After the import has run, click on the **Objects** view and choose one of your objects that has been imported. Check that the Attributes are populated correctly.

Run some tests to compare the data within erwin EA against the data within System Architect.

Once the model content has been migrated from System Architect, the [Diagram Import](#) can commence.

---

## 3 Steps for importing models - Customized System Architect metamodels

This example takes you through a migration with a customized metamodel in the source tool System Architect... the output will be a set of CSV files (one per metatype) that can be imported into the erwin EA tool.

1. Match metamodels in System Architect and erwin EA
2. Create mapping files and data outputs
3. Run import

### Match meta models in SA and EA

First of all, you need to understand the scope of the meta model that you wish to map to erwin EA.

A good understanding of System Architect customization may be required if customizations have been performed. It is important that you understand how to [customize](#) the erwin EA platform before starting this guide.

The following shows an example of a customized metamodel and provides two options on how to work out what needs to be mapped into erwin EA.

**i** All base ArchiMate and TOGAF object types, diagram types and relationships should be matched between the two tools out of the box.

**Option 1** - Use the **Meta Model Report** macro found [here](#) - The output will allow you to see what you need to create within the erwin EA client area.

**Option 2** - In System Architect, locate the usrprops.txt file that represents your customizations.

Below are suggested components that may need matching in erwin EA, if they have been customized within System Architect

1. Create or identify an **object type** in Agile EA that matches any System Architect **DEFINITION** keywords.

The screenshot shows a configuration window titled 'CONFIGURE' for a custom object type. The 'Name' field contains 'Pilot'. The 'Description' field contains 'A pilot flies military and commercial aircraft.' The 'Communities' dropdown is set to 'All'. The 'Packages' dropdown is set to 'ArchiMate'. There is a 'Logo:' field at the bottom left and an 'ADD' button at the bottom right.

**Agile EA custom object type**

---

*RENAME DEFINITION "User 3" TO "Pilot"*

*DEFINITION "Pilot"*

*{*  
*...*  
*}*

### System Architect usrprops example

2. Create **relationships** in erwin EA for **relationships** that exist within System Architect.



The names do not have to exactly match. The mapping file in step 2 will allow mapping of different names.

From	From Name	To	To Name	From
Pilot	flies	Aircraft	flown by	Pilot

### Agile EA custom relationship

*DEFINITION "Aircraft"*

*{*

*CHAPTER "Relationships"*

*PROPERTY "flown by"*

*{ EDIT ListOf "Pilot" MIRRORS "flies" RELATE BY "flown by" LENGTH 4500 HELP "Choose the pilot that flies this aircraft" }*

*}*

*DEFINITION "Pilot"*

*{*

*CHAPTER "Relationships"*

*PROPERTY "flies"*

*{ EDIT ListOf "Aircraft" MIRRORS "flown by" RELATE BY "flies" LENGTH 4500 HELP "Choose the aircraft this pilot flies" }*

*}*

### System Architect usrprops example

3. Create **RelationshipList** attributes in erwin EA object types to match **Properties that represent relationships** in System Architect definitions.

Relationships can have attributes created to represent them as they are within System Architect - For this we would use a RelationshipList attribute... In erwin EA relationships are handled differently and can represent themselves as just an entry within the objects' relationships table, the attribute does not have to exist for the relationship to be seen.

	<p><i>CHAPTER "Relationships"</i></p> <p><i>PROPERTY "flies"</i></p> <p><i>{ EDIT ListOf "Aircraft" MIRRORS "flown by" RELATE BY "flies" LENGTH 4500 HELP "Choose the aircraft this pilot flies" }</i></p>
<p><b>Agile EA custom RelationshipList</b></p>	<p><b>System Architect usprops example</b></p>

4. Create other **attribute** types to match **properties** within System Architect.

All of the property types in System Architect can be mapped to **attribute types** in erwin EA.

	<p><i>CHAPTER "Details"</i></p> <p><i>PROPERTY "Range"</i></p> <p><i>{ EDIT Text LIST "Range" LENGTH 50 HELP "What is the range of the Aircraft?" }</i></p>
--	---

<b>Agile EA custom attribute</b>	<b>System Architect usrprops example</b>
----------------------------------	--

**i** In migrating HeterogeneousListOf properties, the relationships are transferred on an individual basis and can be seen as separate relationships in erwin EA objects - If the user wants to represent these relationships with an attribute, the user will need to create multiple relationshiplists within the destination object type, one for each object type available to be selected within the source System Architect heterogeneouslistof property

5. Create **diagram types** in erwin EA to match different **types of diagram** in System Architect.

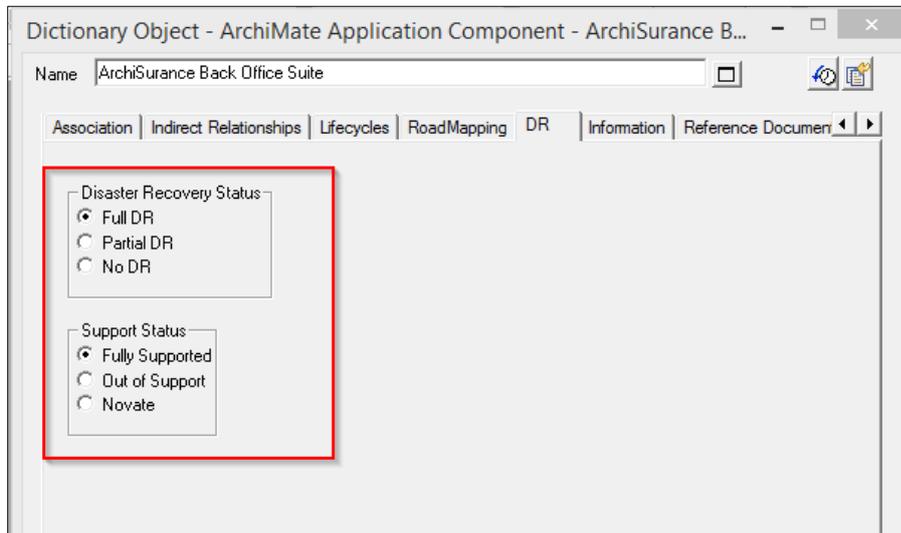
**i** Note: use the 'Add All' button to add all of the relationship types between the object types to the diagram type

	<p><i>RENAME DIAGRAM "User 1" TO "Travel"</i></p> <p><i>RENAME SYMBOL "User 1" TO "Airline"</i></p> <p><i>RENAME SYMBOL "User 2" TO "Aircraft"</i></p> <p><i>RENAME SYMBOL "User 3" TO "Pilot"</i></p> <p><i>SYMBOL "Airline" { ASSIGN TO "Travel" }</i></p> <p><i>SYMBOL "Aircraft" { ASSIGN TO "Travel" }</i></p> <p><i>SYMBOL "Pilot" { ASSIGN TO "Travel" }</i></p>
<b>erwin EA diagram type</b>	<b>System Architect usrprops example</b>

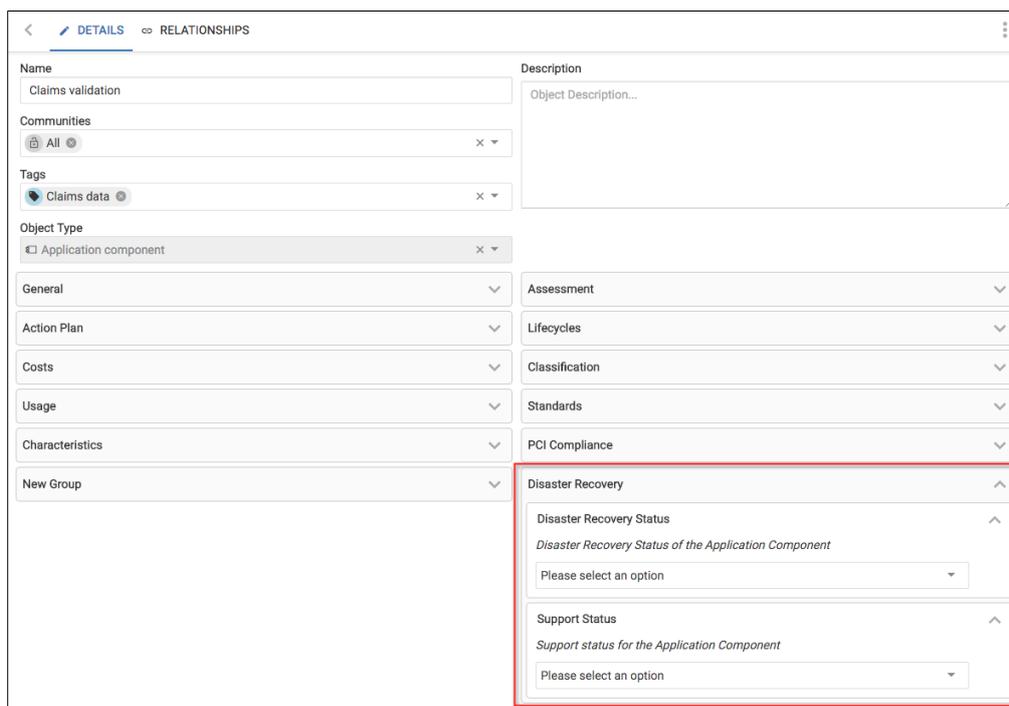
**Example:**

In this worked example we have matched some customized properties (named Disaster Recovery Status and Support Status) within an Application Component definition type in System Architect, into erwin EA with some corresponding 'List' attributes in the application component object type.

We will then map the data and produce the output files, before importing into erwin EA



### Custom properties in System Architect



### Matched attributes in erwin EA

## Create mapping files and data outputs

Within this section, you will

- Create an erwin EA Agile schema file
- Create a mapping file as a base for mapping
- Create some explorer diagrams within System Architect, to extract the model information that you want

- Download the exchange tool
  - Perform the customization mapping and output some CSV files
1. We need to let our exchange tool know the metamodel within our erwin EA client area - Now that both the meta models for System Architect and erwin EA match, you need to download the schema for your customized erwin EA environment.

Click on the following URL (you will need to login): <https://myerwin.io/ea/api/meta-structure>

Select File > Save Page as... and save the file with a .spp extension, such as "erwinEA-metaStructure.spp"



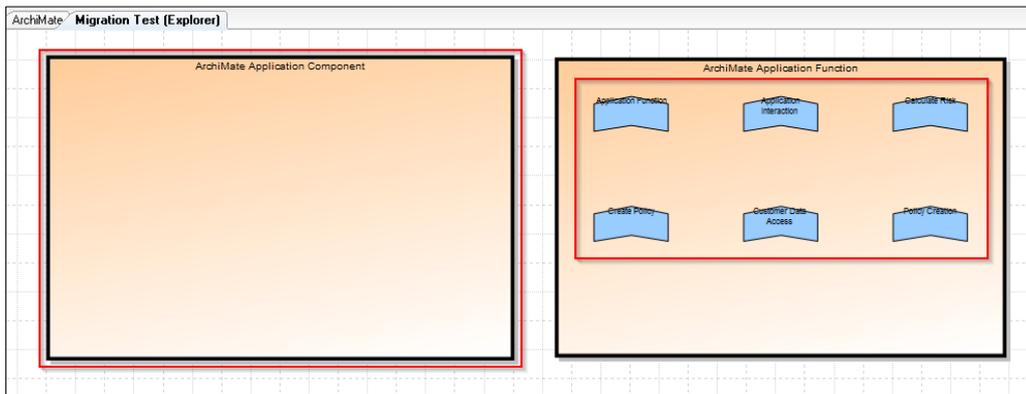
You will need regenerate this structure file, each time you change the structure of the meta model in erwin EA.

2. Create an **Explorer** diagram within System Architect. The diagram needs to contain either
  - the **specific definition instances** that you want to migrate to erwin EA
  - the **collections** for each **definition type** that you want to migrate to erwin EA - The exchange tool will export all instances of a definition type for mapping if it finds an **empty collection** on the diagram

The diagram can also contain custom definition types.

3. Save the diagram

In the example below, we will transfer the specific definition instances of type Application Function and all definition instances of type Application Component **within the workspace that we are in.**

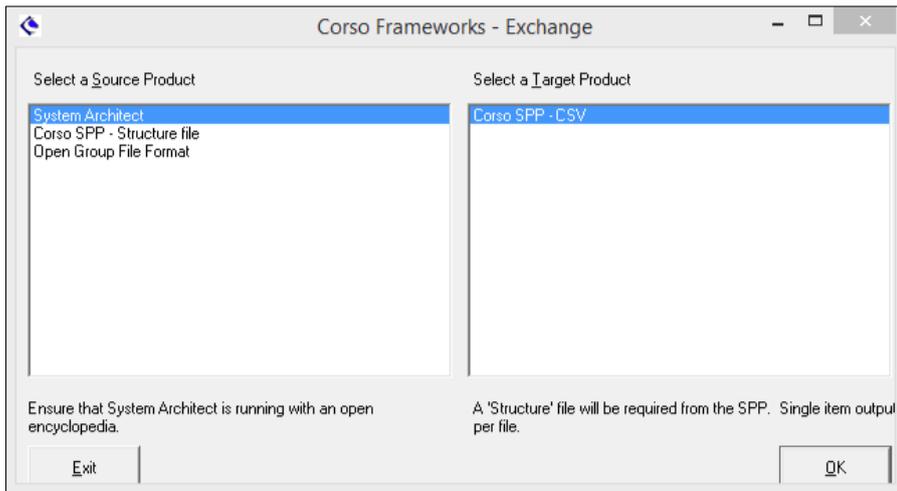


**An example System Architect explorer diagram**

4. Next download a copy of the [exchange](#) tool, extract it and start it on the Windows environment where your System Architect instance is located.

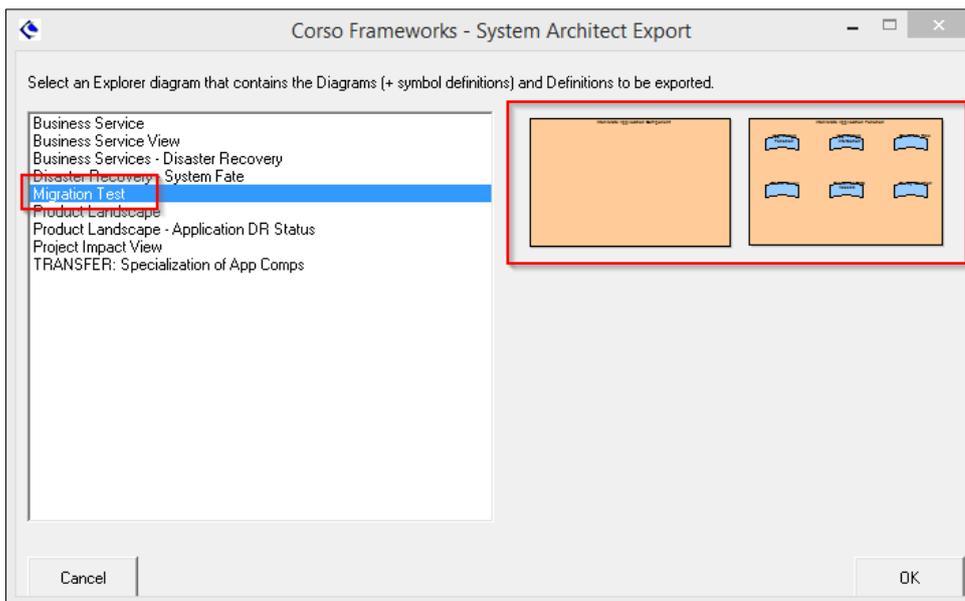


**IMPORTANT:** Your System Architect instance needs to be open for the exchange tool to run



**erwin exchange tool**

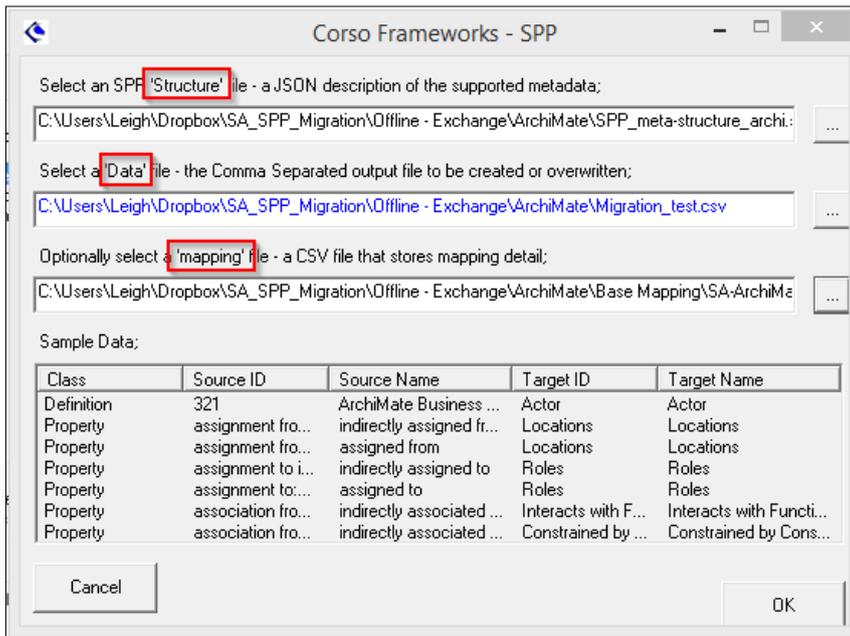
5. Choose **System Architect** as the source product and **Corso SPP - CSV** as the destination product.
6. Next, choose the explorer diagram you've just created (e.g. Migration Test) from the list available and press OK.



7. In the select an SPP '**Structure**' file box, select the .spp file created at step 2.1
8. Next, in the '**Data**' file box, select a location and filename (with .csv extension) for the output files to be generated to.
9. In the last field, '**mapping**', select a location and create a filename press OK.

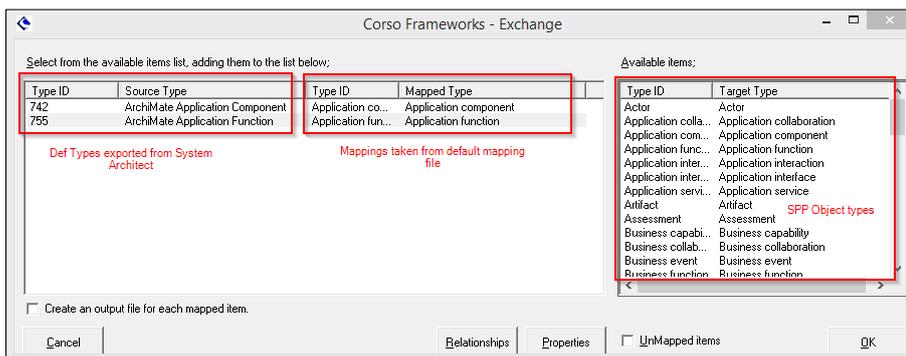
If you have an existing mapping file, you can enter this file location (black text denotes an existing file, blue will create new)

This file will store all of the mappings between System Architect and erwin EA object types, attributes and relationships.



10. You will now see

- a list of object types that were contained within the System Architect explorer diagram (left field, left hand columns) - this will include any custom definition types
- mappings taken from the selected mapping file (left field, right hand columns) - You will see spaces in the centre columns if more mappings now need to occur (if you have any custom object types to map)
- a list of object types from the erwin EA structure file (right field) that can be mapped to the empty object types from System Architect.



### Exchange mapping of object types

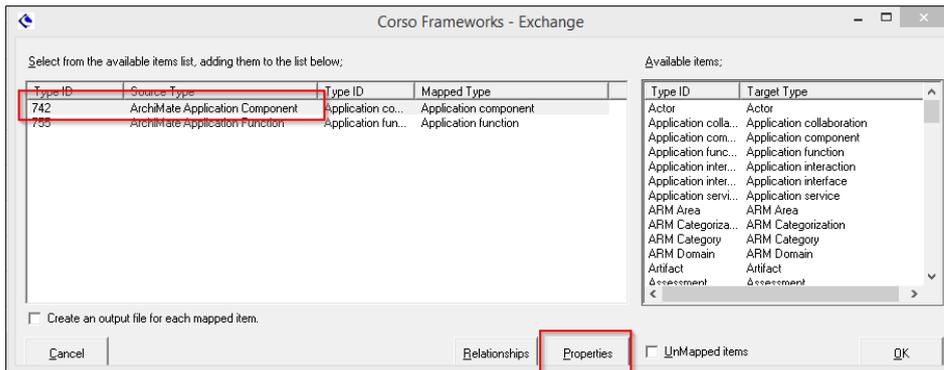
We can see that we can use the standard mapping at this point, as our object types themselves are not customized, although some of the attributes within them have been customized.

If you did want to map an object type, you would simply **Drag** each **Agile EA object type** (right hand pane) onto a definition type on the left hand pane.

Pressing **Enter**, or **double-clicking**, will also copy the 'Available item' into the 'Mapped' columns.

**i** | Note: To remove a mapping, select the row in the left hand table and use the delete key

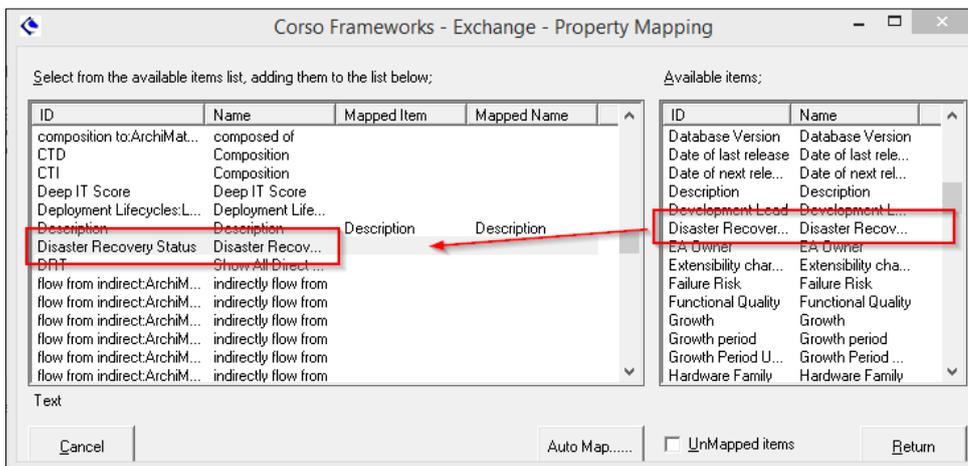
- Now, choose the mapped type **ArchiMate Application Component** in the left column and click **Properties**.

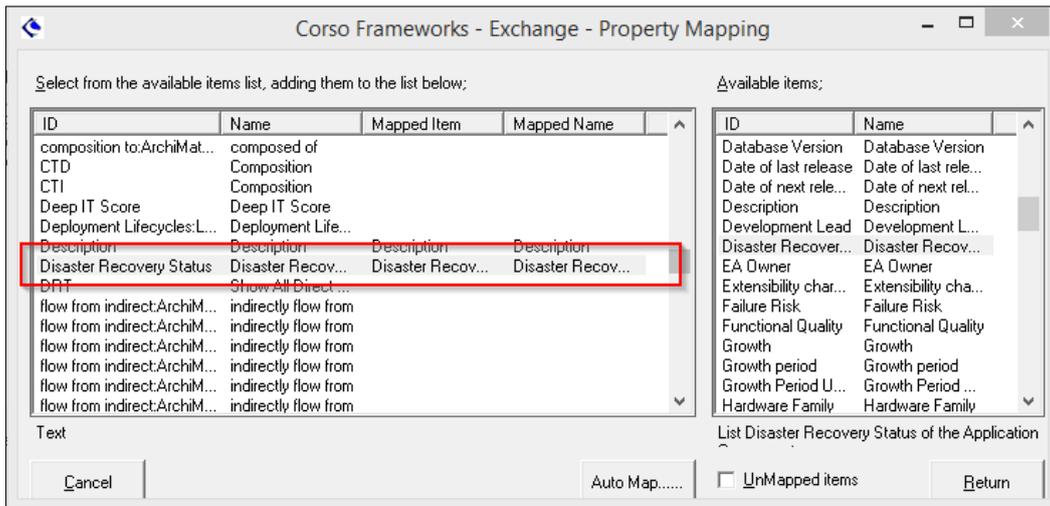


If you specified the names of erwin EA attributes to be the same as System Architect properties, you can press the **Auto Map** button. This will match properties by name.

**i** | Note: The AutoMap button can be used more than once for properties and relationships - The more times it is clicked the wider the mapping criteria will get - Please do check the properties and relations that are automapped before proceeding to output of data.

- Scroll the left hand columns until you find the **Disaster Recovery Status** property from within System Architect. Scroll in the right hand columns until you find the corresponding attribute within the erwin EA attribute list. Select the appropriate attribute from the right hand column and drag onto the corresponding property in the left hand column.
- Do the same for the **Support Status** property also. Once complete, select the **Return** button





(Optional) You can look at the **Relationships** section of the mapped types also.

**i** Note: The System Architect 'relationships' come from properties that reference other definitions (eg. ListOf) and not from the relationship types. Heterogeneous properties will generate a relationship entry for each definition type referenced.

- At this point the mapping is complete - We have used the base mapping file and added in our custom attributes for mapping - Next, click on the checkbox **'Create an output file for each mapped item'**.

Selecting this will output data files for all of the mapped object types in the list.

**i** Note: Leaving this checkbox blank will require an object type to be selected from the list, before an output data file can be created for the selected type.

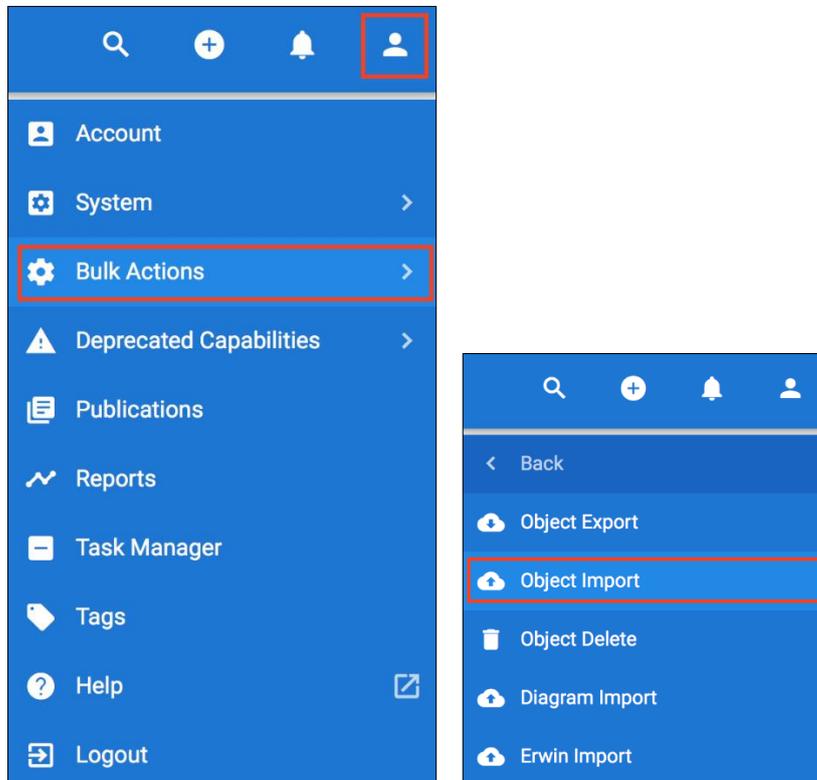
- Press **OK** and then **yes** when asked if you would like to continue. The mapping will now occur - a message will be displayed upon completion (this can take some time and progress can be monitored from a report that runs in the open System Architect session).
- If you look in your designated output directory, you will now see files created for each of the object types and one containing all relationships (the mapping file is also created or updated).

**i** Note: If you selected to create multiple files, the data files are appended with the System Architect definition type number

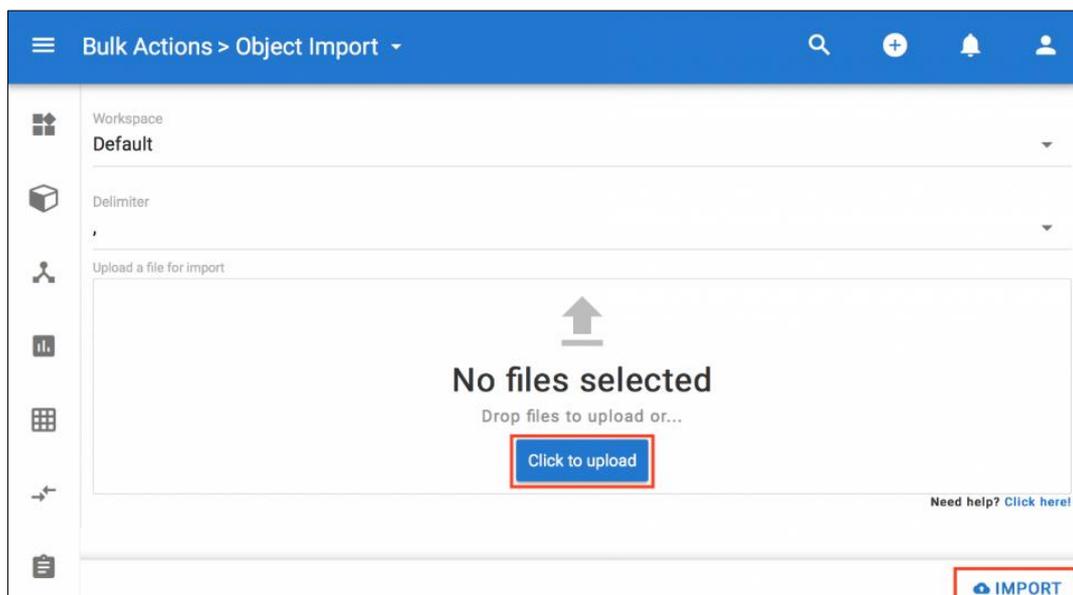
In your output .csv file, you should see the corresponding data has been exported and is now ready for import into erwin EA.

## Run import of data

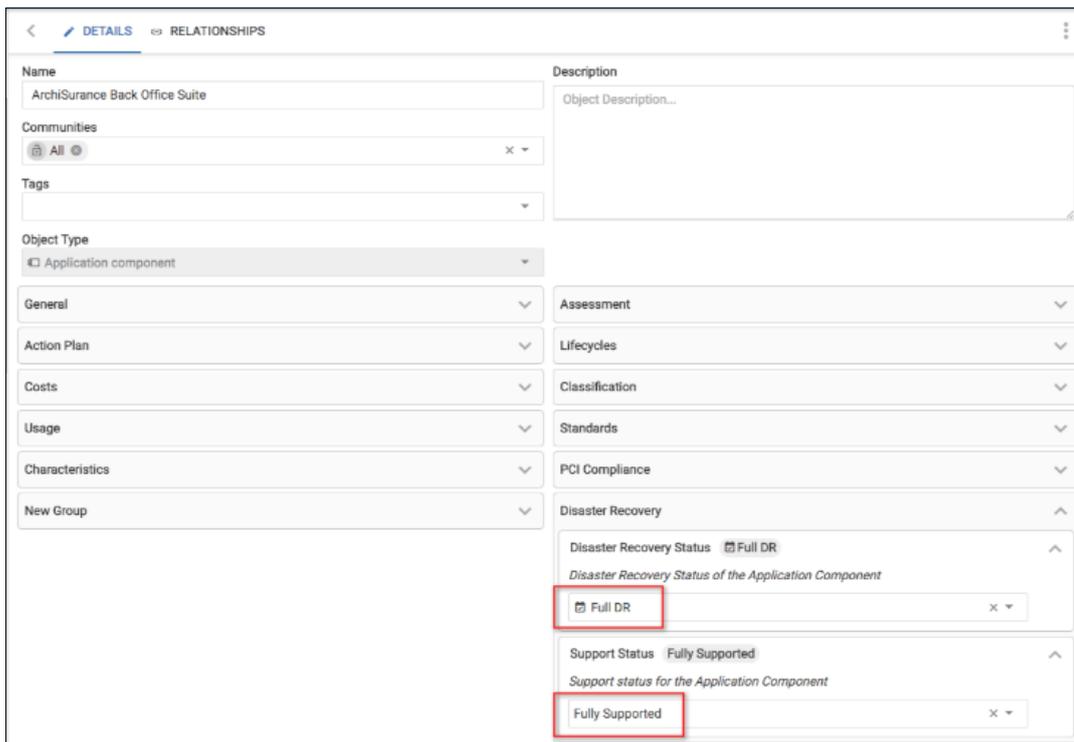
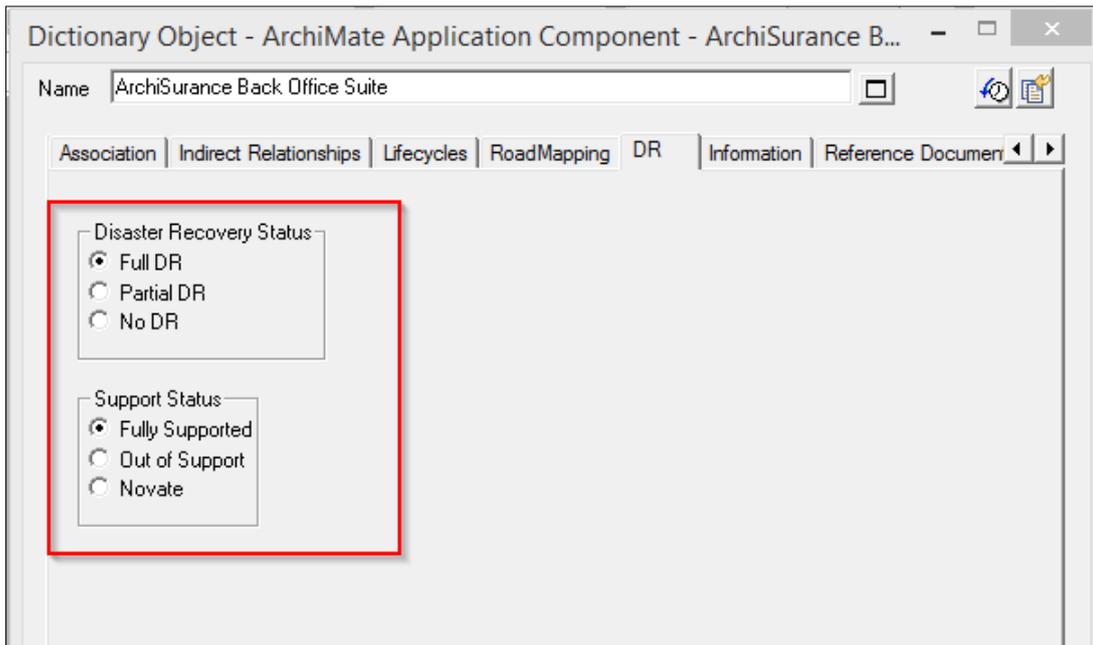
1. In erwin EA (as an administrator), open **User > Bulk Actions > Object Import**



2. Click on the **Click to upload** button and choose all of the csv files that you would like to import.



3. Press the **IMPORT** button in the bottom right hand corner.
4. After the import has run, click on the **Objects** view and choose one of your objects that has been imported (ArchiSurance Back Office Suite in our example). Notice the Attributes do now exist and are populated correctly.



Run some tests to compare the data within erwin EA to what was seen within System Architect.

Once the model content has been migrated from System Architect, the **Diagram Import** can commence.

---

# System Architect Diagram Import

**Role Applicable to:** [Client Administrator](#)

Import of diagrams from the System Architect tool is now enabled on the erwin EA agile tool. Diagrams need to be output in the standard System Architect XML format, which the erwin EA agile tool can then read and allows users to map to erwin EA agile diagram types.

**!** **IMPORTANT:** Diagram Import needs to be performed after the model content is transferred from within System Architect - The process for this can be found [here](#) - Doing it in this order allows the objects imported onto the diagrams to be identified and re-used from your model content.

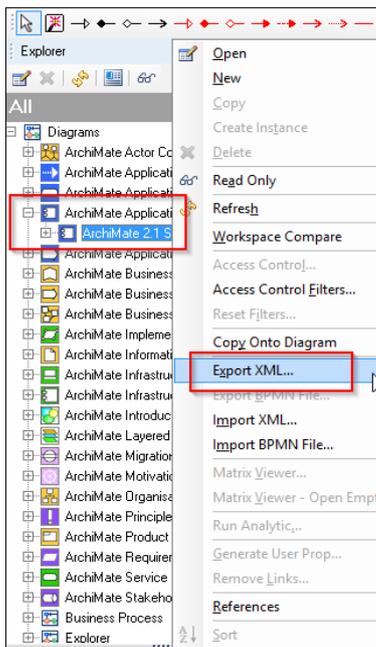
# Export from System Architect

To export Diagrams from System Architect, either

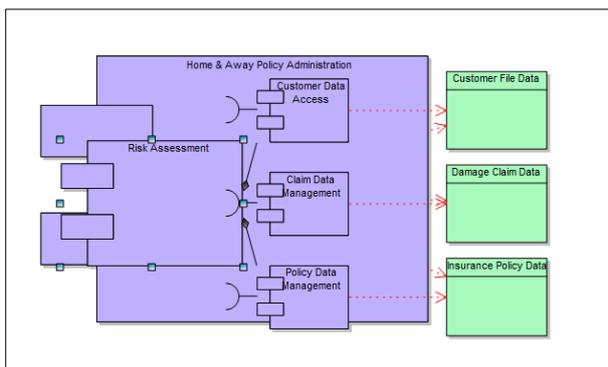
- Install and use **XML Diagram Export** macro to export all diagrams

or

- Expand the Diagrams section, expand a diagram type section, right click a diagram and select **Export XML...**



In this example, we have exported an **ArchiMate Application Structure** diagram into System Architect XML format



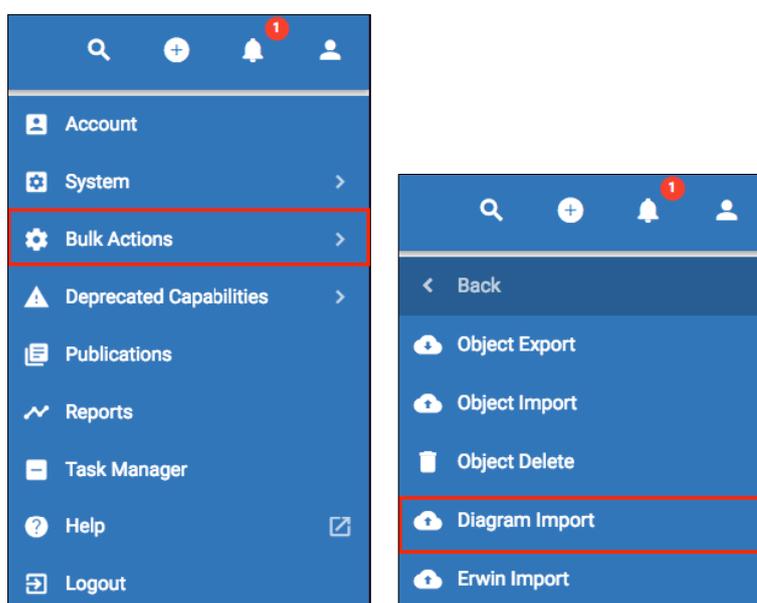
---

# Import into erwin EA agile

As part of our migration process, we have already matched the metamodels of erwin EA agile to our System Architect encyclopedia. This included diagram type, but as we are using a standard ArchiMate diagram type, erwin EA agile already supports it - We have also transferred our model into our erwin EA agile client area from System Architect - Now we will import the diagram.

Log into your client area on the erwin EA agile platform.

1. Navigate to the **User menu > Bulk Actions > Diagram Import**



2. Upload the exported diagram XML file into the **Diagram** field.
3. Upload the mapping file you used for the migration of the model content into the **Mapping** field.
4. Select the **+ Read** button in the bottom right hand corner.

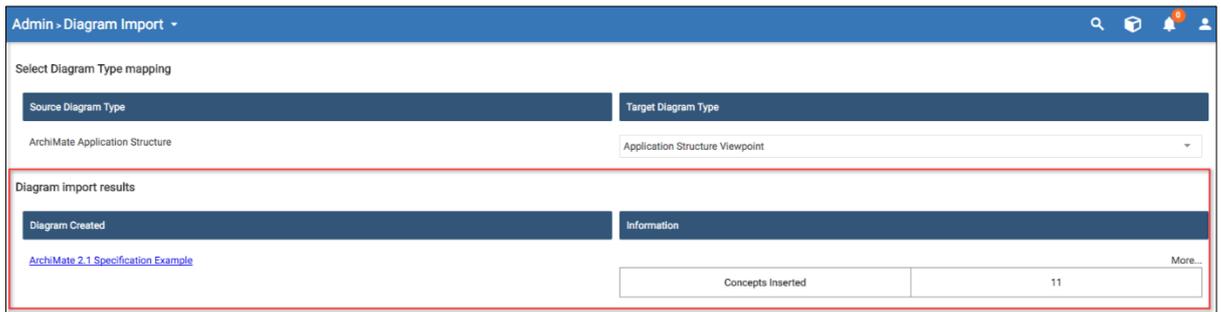
This will identify the diagram types that have been exported from System Architect and allow you to map them to diagram types within erwin EA agile.

5. Map the **ArchiMate Application Structure** diagram to the same type from the drop down ;



6. Select the **+ Create** button.

The mapping will be completed and results shown;



- Select the **More...** option to see what was inserted onto the diagram and more importantly what wasn't;

**Objects Inserted** - Count of Objects inserted

**Names of Inserted Objects** - Names of the inserted objects

**Objects not in SPP** - Concepts not inserted into the diagram, because they do not exist with your erwin EA agile model

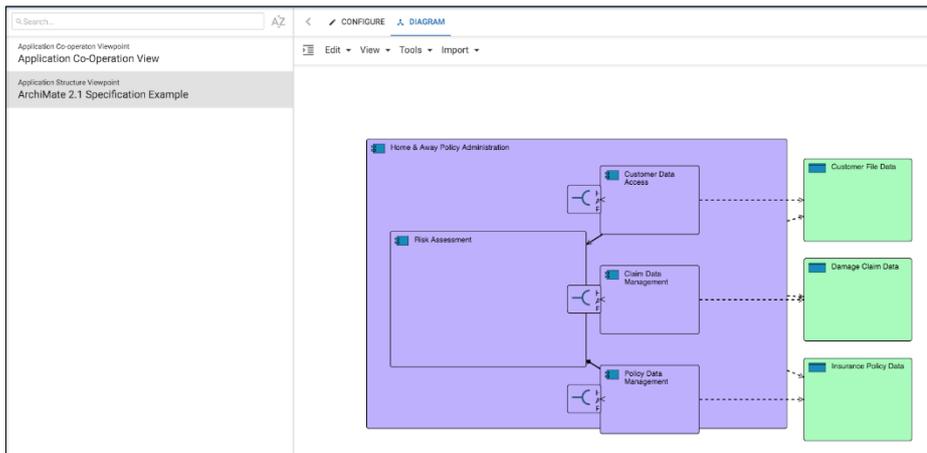
**Types not mapped** - Any object types not mapped, because they do not exist within your erwin EA agile metamodel

**Objects restricted by diagram type** - Objects that were not inserted into the diagram, because the erwin EA agile diagram type chosen does not allow that object type to be placed onto it

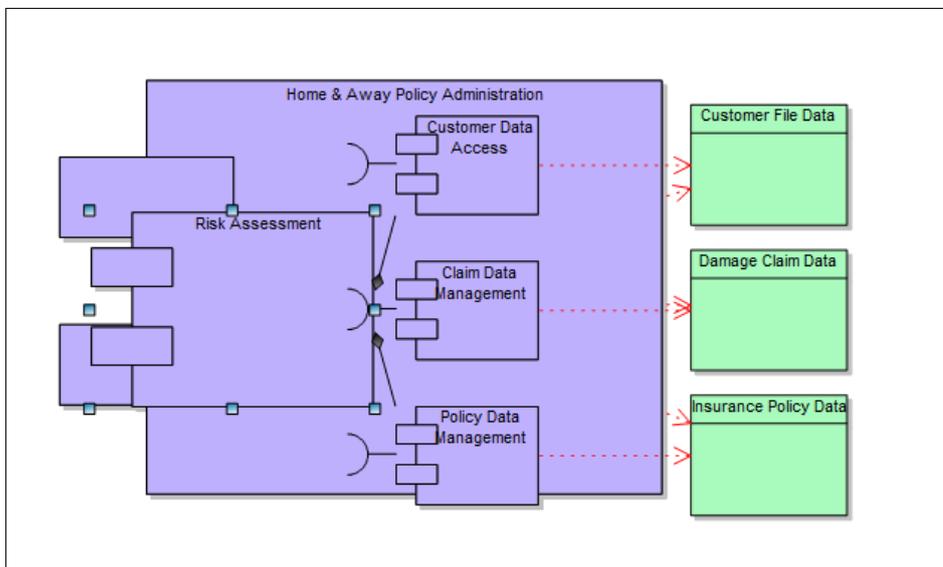
Less...	
Concepts Inserted	11
Names of Inserted Concepts	Home & Away Policy Administration, Risk Assessment, Damage Claim Data, Insurance Policy Data, Customer File Data, Claim Data Management, Customer Data Access, Policy Data Management, HAPA - RA_CDA, HAPA - PDM_RA, HAPA - CDM_RA
Concepts not in SPP	0
Types not mapped	0
Concepts restricted by diagram type	0

- Select the hyperlink of the diagram name, underneath the **Diagram Created** section.

This will open the diagram that has been created;



**Imported diagram within erwin EA agile**



**Original diagram within System Architect**

- IMPORTANT:** The current implementation does not maintain box-in-box relationships from System Architect diagrams - If the box-in-box is only a graphical representation, with no relationship behind it, then the coordinates will be mapped on the erwin EA agile diagram - But if there is a relationship for box-in-box, the relationship is shown on the diagram and the user needs to re-enable the box-in-box representation on the erwin EA agile diagram.

---

# System Architect Macros

**Role Applicable to:** [Client Administrator](#)

On this page, find some useful System Architect macros to help you at the beginning of the **migration** process to the erwin Data Foundation Platform

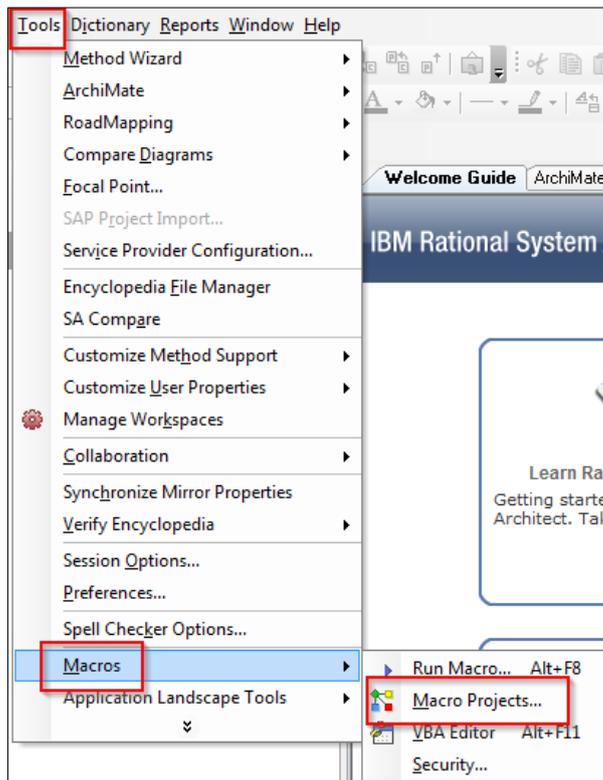
**MetaModelReport** - This macro can be run within System Architect - The output will help you to understand both the metamodel you have built (and thus needs to be matched within the SPP client area) and also which object types you are actually using - Allowing you to narrow the focus of the data that actually needs to be collated into explorer diagrams for migration

**OutputXMLDiagrams** - This macro can be run within System Architect - The output will be an xml that contains all of the diagrams from the System Architect encyclopedia that it has been run on - These xml files can then be used to **import the diagrams** into the erwin DFP.

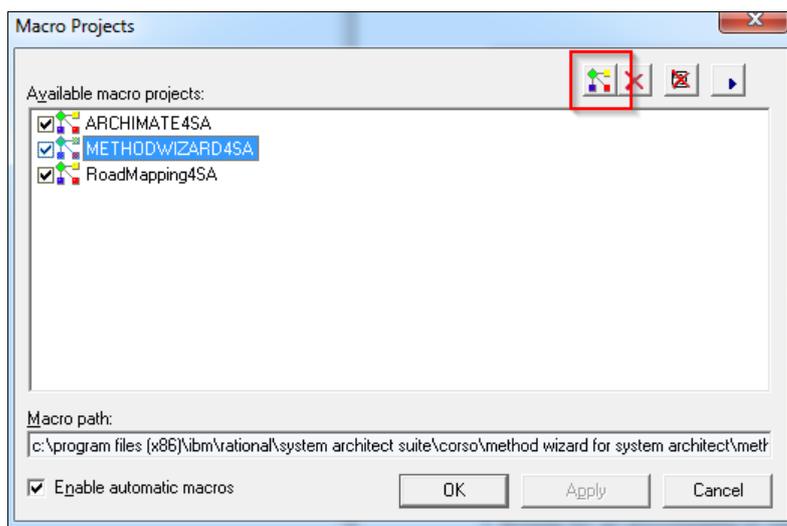
---

# How to use the macros

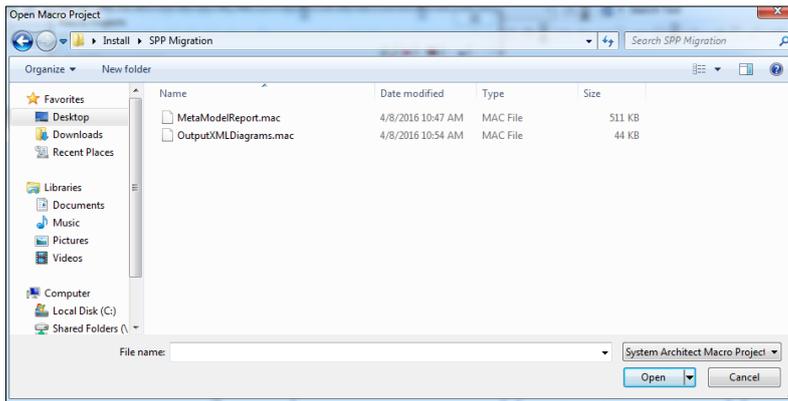
1. Download the macro file(s) from above and place onto the system where System Architect is installed
2. Open System Architect and an encyclopedia
3. Select **Tools > Macros > Macro Projects** from the menu



4. In the Macro Project dialog, select to add a new project



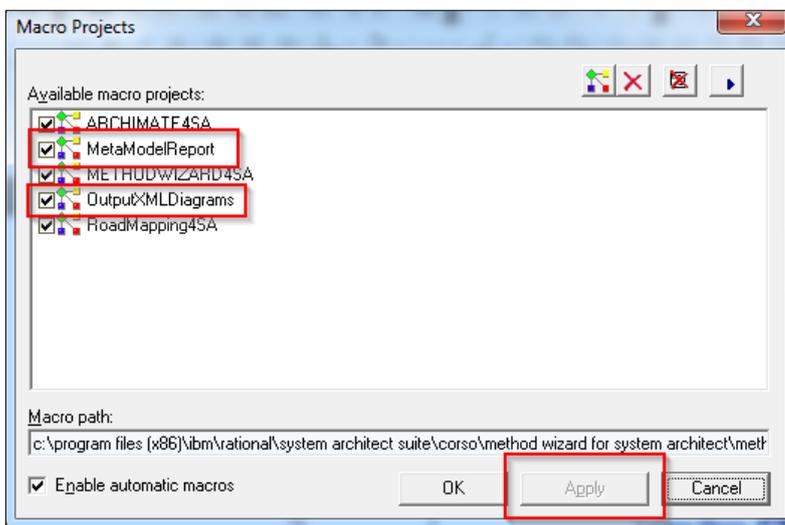
5. Select the Macro File from the location on the system



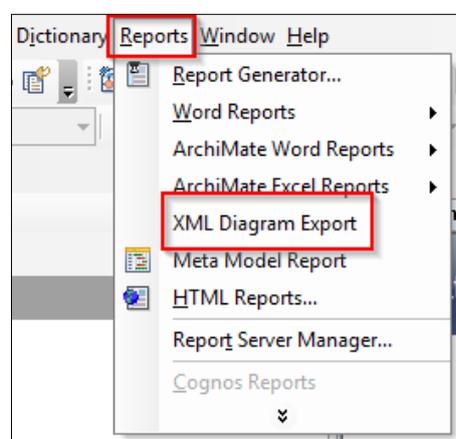
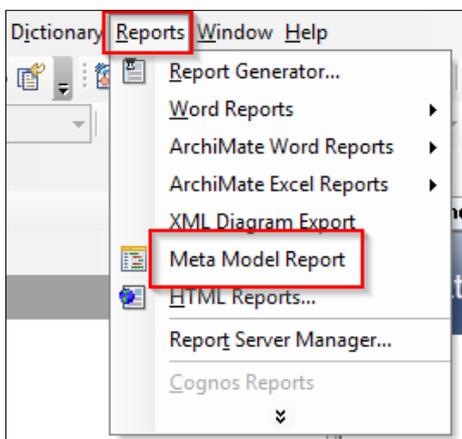
You will need to repeat steps 4&5 for each macro you would like to add

6. Notice that any added macros now appears in the Macro Projects list

7. Select the **Apply** button, followed by the **OK** button



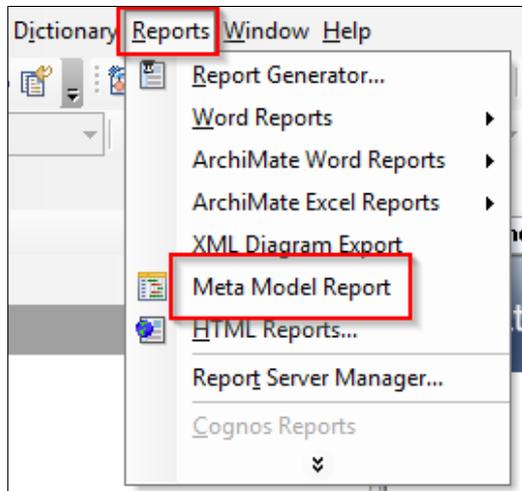
Any new macros will now be seen within the System Architect environment - The 2 we have added have menu items attached, under the **Reports** menu items



---

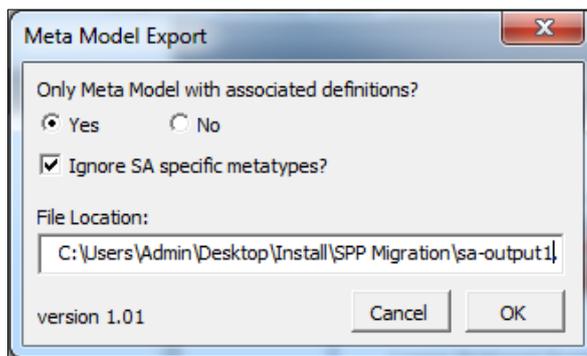
# Meta Model Report Macro

1. Select the **Reports > Meta Model Report** menu option



2. Select the options from the dialog that is displayed

- **Only Meta Model with associated definitions?** - This will only pick out the definition types that have instances created against them - To help you identify the parts of the model that you are using
- **Ignore SA specific metatypes** - this will ignore the systemic definition types that are not considered to be model content
- Enter a location for the output file and select **OK**



The file will be output - this file is split into sections

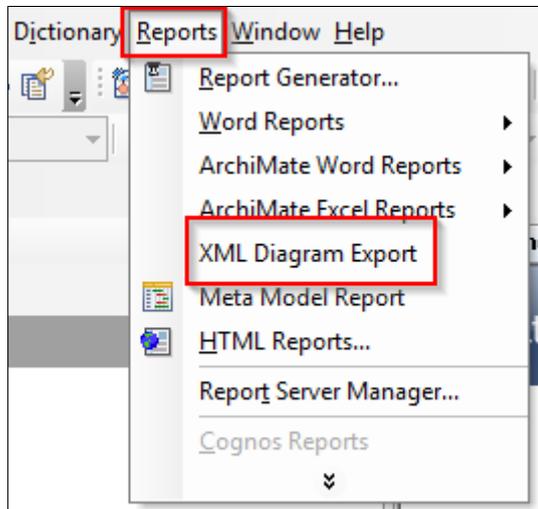
1. Object types that exist
2. Relationships for each object type
3. Groups and Attributes for each object type
4. Diagram Types identified

Once you have this information, you can easily begin the Match meta models section in the **migration process**.

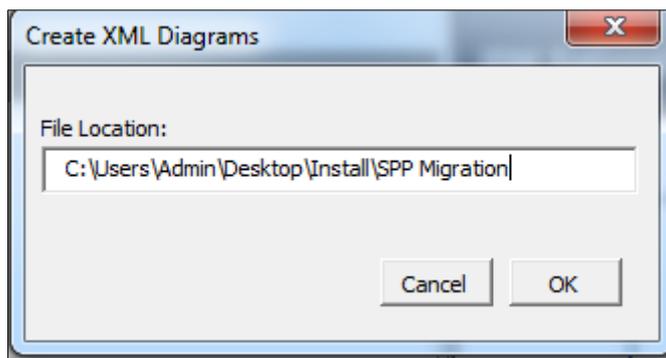
---

# XML Diagram Export Macro

1. Select the **Reports > XML Diagram Export** menu option



2. Enter a location for the output files and select **OK**



A success message will be displayed, also telling you how many XML files were created containing diagrams

These xml files can then be used to **import the diagrams** into the erwin DFP

---

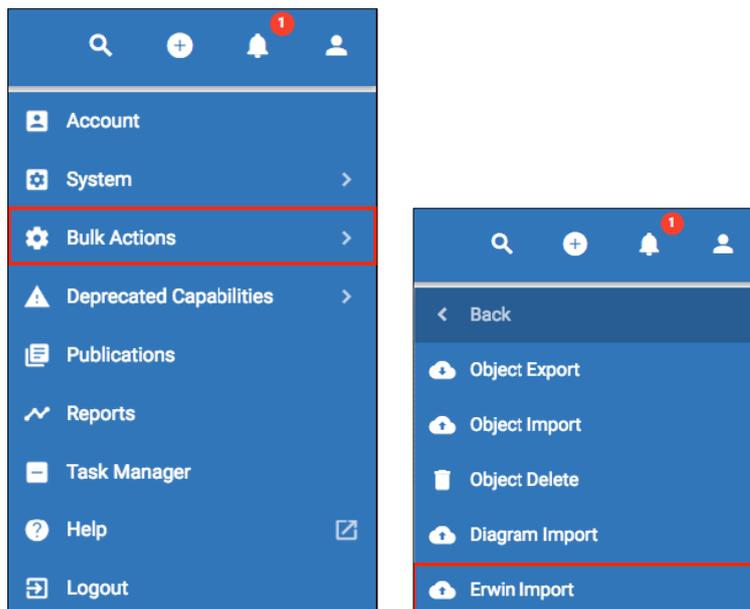
# Migrating model data from erwin DM to erwin EA Agile

Role Applicable to: [Administrator](#)

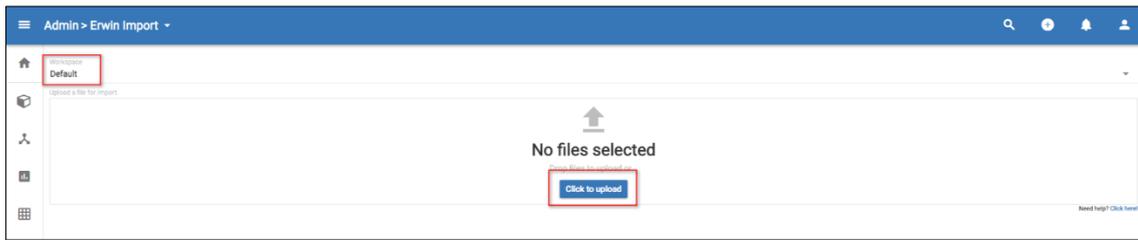
Logical data models can be moved into erwin EA Agile from the erwin Data Modeler tool. Users and interested stakeholders can then link the entity model objects into the rest of the enterprise architecture.

## *To do this*

1. In erwin DM, use the **File > Save As** option to save the model.
  - Before saving, select the '**Save as type**' option to be **XML Standard (\*.xml)**
2. We can then import this xml file into the erwin EA tool
  - Log-in to the erwin EA tool as an administrator
  - Select the User menu > Bulk Actions > Erwin Import option



- Select the Workspace you would like to import the model data into
- Select the 'Click to Upload' button to upload the model xml file
- Click the Import button in the bottom right hand corner



Once complete, a message will be displayed to you, telling you the objects that have been created.

After you have run the import, if you go into the object list menu, you will be able to see the entity objects have been imported, along with their attributes and relationships.

**IMPORTANT:** The erwin Import v1 is only additive, meaning an existing entity will be updated with new attributes and relationships on further imports. Attributes or relationships that have been removed from an entity in the erwin model will not be removed from the erwin EA model on further imports (the user will need to remove these, using the entity relationship diagram in erwin EA). Import and update is based on matching the object name.

---

# Developer API

## SPP API

### Data Foundation Platform API

**Role needed to use:** [Client Administrator](#)

erwin provides an API (Application Programming Interface) for listing objects and object types that are available in a customer client area.

The API can extract data in a JSON format and help you integrate with your existing business applications.

You must be a client administrator to access the API service with authentication credentials. The authentication uses basic username and password authentication.

The erwin API is available to all erwin EA agile clients. To use the API, you will need to provide the following URL: [api.myerwin.io](https://api.myerwin.io) and the valid method name for the API calls.

For example: <https://api.myerwin.io/object-types>

### API Calls

#### Workspaces

##### GET /workspaces

Retrieve all workspaces.

##### **Implemented query parameters:**

- None

##### **Expected query parameters:**

- None

##### **Errors:**

- None

##### **Example:**

<https://api.myerwin.io/workspaces>

---

## Object Types

### GET /object-types

Retrieve all object types.

#### **Implemented query parameters:**

- None

#### **Expected query parameters:**

- None

#### **Errors:**

- None

#### **Example:**

<https://api.myerwin.io/object-types>

### GET /object-types/:type\_id

Read the object type with id :type\_id.

#### **Errors:**

- 404 if the object type does not exist.

#### **Example:**

<https://api.myerwin.io/object-types/26555>

## Objects

### GET /objects

Retrieve all objects on the default workspace.

#### **Example:**

<https://api.myerwin.io/objects>

### Query parameters (mixable):

You can mix the following paramaters to further refine your queries.

- **GET /objects?workspace=ws**

Retrieve all objects on the workspace ws by workspace id.

The workspace id can be retrieved by examining the URL in the platform when a workspace is open.

#### **Example:**

---

<https://api.myerwin.io/objects?workspace=248>

- **GET /objects?type=id**

Retrieve all objects that are of type id.

**Example:**

<https://api.myerwin.io/objects?type=26555>

- **GET /objects?name=string**

Retrieve all objects whose name (case sensitive) contains the given string.

**Example:**

<https://api.myerwin.io/objects?name=Claim%20Handling%20Improvement>

- **GET /objects/:id**

Retrieve the object with id :id.

**Example:**

<https://api.myerwin.io/objects/62397>

- **Errors:**

404 if the object type does not exist.

---

# API Example

This example makes use of a google sheet to demonstrate how calls can be made to the SPP API.

We're making use of the set of google scripts supplied by 'Fast Fedora' available [here](#).

For more information on the JSON parsing script click [here](#).

Create a google sheet ([sheets.google.com](https://sheets.google.com))

When in a sheet, open Tools... Script Editor

Ensure you embed a copy of the '[google-docs/scripts/ImportJSON/Code.gs](#)' script from 'Fast Fedora' in the script window. The code is also available

here: <https://github.com/bradjasper/ImportJSON/blob/master/ImportJSON.gs>

For authentication, you need to add the following code before the main section (SPP uses basic authentication):

```
function ImportJSONBasicAuthentication(url, query, parseOptions, username, password) {  
  var fetchOptions = {  
    "headers": {  
      "Authorization": 'Basic ' + Utilities.base64Encode(username + ':' + password)  
    },  
    muteHttpExceptions: true  
  };  
  return ImportJSONAdvanced(url, fetchOptions, query, parseOptions, includeXPath_,  
defaultTransform_);  
}
```

Save the script with the save button.

Now you can use our [example google sheet](#) with the Corso API by making your own copy of it ([file... make a copy](#)).

Change the username and password to your own username and password.

The sheet template contains a set of tabs which you can navigate through and see the examples working with your data.

The sheet is designed to give you a feel for the 'art of the possible' with the API.

For more information or more help, contact corso support.

---

# API Limits

The Corso SPP API is available to administrators of the Corso SPP. The limits below are the commercial terms under which the API may be run. For additional usage and limit requests, please contact your account manager.

Corso SPP	API Requests - Upper Limit
Trial Users	100 requests/day/organization
Standard License	Max - 5000 requests/day/organization or 250 requests /user license whichever is lower

## Important Notes

- The PST Time zone will be considered.
- We notify SPP administrators (Users with "Administrator" profile) if your organization exceeds the API limit. We also provide API Statistics for better assessment of your integration requirements.
- To optimize your API usage, get maximum **200 records** with each request and in future editions of the API; insert, update or delete maximum **100 records** with each request.

*Example:* If you have purchased 10 user licenses, you are allowed 2500 requests per day. Using each of those 2500 requests, you can read a maximum 200 records.

**That would be 2500 x 200 i.e., 500,000 records can be read per day.**

- In case, your application requires more than the upper limit, your additional API requests will not be processed. To avoid data transfer issues, please assess your API requirements well in advance. If you need any help, please contact our Support at [support@corsosoftware.com](mailto:support@corsosoftware.com)

## Check API Request Usage (Future Edition)

You can view the total limit for API calls and the current day's API usage in Corso.

1. In Corso SPP click **Admin > Developer Space > SPP API**.
2. In the SPP API page, the **Usage statistics of API** section represents the per day usage and total limit for the API calls.
3. When the user calls the API, **Check Calls** button will be displayed. Click the button to find number of API calls made in your client based on client and username. All the active and confirmed usernames will be displayed in the dropdown list. You can select any username to know the API call count details of that user.



These UI changes will be shown only when the user has made at least one api call. If users did not call any API, then these details will not be displayed.

---

# Reporting with JS Reports

The example below was constructed using the on-premise version of JS Reports - Installation information can be found at <https://jsreport.net/on-prem>

The example below explains how JSReports can use JavaScript calls to the API to create a report.

## The Example Report

Project Context; Application Report							
Application	Description	Start Date	End Date	Project	Project Start	Project End	Project Manager
Back Office Suite	Integrated back-office automation replaces the separate back-office applications.	06/02/2017	30/11/2020	CRM System Integration	31/10/2016	30/09/2017	Perry White
				Back Office System Integration	01/01/2016	30/11/2017	Martha Kent

The report above show an **application component** object and some of its **attributes**, as well as related **work package** objects (projects), picking out some of those attributes and then a related **actor** object (project manager)

## JS Reports

For this example, it is assumed that JSReports has been installed locally, with access to 'jsreport studio'.

JSReports creates output using a 'template', which is composed mainly of HTML. The template identifies an 'engine', a 'recipe', any required JavaScript, and some sample JSON test data. The engine refers to the syntax used in locating data for insertion, and the recipe refers to the way the output is formatted, e.g. PDF.

## Supporting Files

Users can download the **template**, **data**, and **script** used for this example. The content of each file can then be copied and pasted into the appropriate feature of jsreport studio. The test data can be linked to the template and run without alteration, however the script will require changing before it will run successfully. Once the required modifications have been made to the script, it can be linked to the template.

## The Template

The following template (download file from above) describes a table of Applications and their related Projects;

```

1 <h1>Project Context; Application Report</h1>
2 <hr>
3 {{#if error}}
4 {{error}}
5 {{else}}
6 <table>
7 <tr>
8 <th>Application</th>
9 <th>Description</th>
10 <th>Start Date</th>
11 <th>End Date</th>
12 <th>Project</th>
13 <th>Project Start</th>
14 <th>Project End</th>
15 <th>Project Manager</th>
16 </tr>
17 {{#each application}}
18 <tr>
19 <td>{{name}}</td>
20 <td>{{description}}</td>
21 <td>{{startDate}}</td>
22 <td>{{endDate}}</td>
23 {{#each project}}
24 {{#if @first}}
25 {{else}}
26 </tr><tr><td></td>
27 <td></td>
28 <td></td>
29 <td></td>
30 {{/if}}
31 <td>{{name}}</td>
32 <td>{{startDate}}</td>
33 <td>{{endDate}}</td>
34 <td>{{manager}}</td>
35 {{/each}}
36 </tr>
37 {{/each}}
38 </table>
39 {{/if}}
40 <hr>
41 <style>
42 table, th, td {
43 border: 1px solid black;
44 border-collapse: collapse;
45 font-family: Arial, Helvetica, sans-serif;
46 font-size: 10px;
47 padding: 5px;
48 }
49 th {
50 border-bottom: 2px solid black;
51 background-color: #3F6CAF; /* Dark Blue */
52 color: white;
53 font-weight: bold;
54 }
55 tr { text-align: left; }
56 tr:nth-child(even) { background-color: #C5CEE2; } /* Light Blue */
57 tr:nth-child(odd) { background-color: #E4E8F2; } /* Pale Blue */
58 </style>

```

Embedded in the HTML are statements from the 'handlebars' engine, which locates data using delimiters {{ and }}. Iterations and decisions can be introduced using {{#each ... }} and {{#if ... }} statements. The template should be associated with some test data.

---

## **Example JSON Test Data**

Some simple JSON test data is not only essential for testing and developing the template, but will act as a useful reference when generating the JavaScript JSON output;

```
{
  "application": [
    {
      "name": "Back Office Suite",
      "description": "Integrated back-office automation replaces the separate back-office applications.",
      "startDate": "06/02/2017",
      "endDate": "30/11/2020",
      "project": [
        {
          "name": "CRM System Integration",
          "startDate": "31/10/2016",
          "endDate": "30/09/2017",
          "manager": "Perry White"
        },
        {
          "name": "Back Office System Integration",
          "startDate": "01/01/2016",
          "endDate": "30/11/2017",
          "manager": "Martha Kent"
        }
      ]
    }
  ]
}
```

The JSON object above contains an “application” array of objects. Each ‘application’ object has a “name”, “description”, “startDate”, “endDate”, and a “project” array of objects. Each ‘project’ object has a “name”, “startDate”, “endDate”, and a “manager”.

## **PDF Output**

Using the ‘phantom-pdf’ recipe, the sample data can be rendered into a PDF document, as shown below

---

## Project Context; Application Report

Application	Description	Start Date	End Date	Project	Project Start	Project End	Project Manager
Back Office Suite	Integrated back-office automation replaces the separate back-office applications.	06/02/2017	30/11/2020	CRM System Integration	31/10/2016	30/09/2017	Perry White
				Back Office System Integration	01/01/2016	30/11/2017	Martha Kent

### Required Modifications to the Script

- Replace the “**username**” and “**password**” in `erwinGlobals()`, with valid access details for your own Agile EA area.
- Each of the three **object type ids**, “**692**”, “**736**”, and “**719**”, will need to be changed to the appropriate ids of ‘**Business Actor**’, ‘**Work Package**’ (Project), and ‘**Application Component**’ from the users client area.
- Both of the Project **property ids**, “**1289**”, and “**1293**” need to be changed to ‘**Date Range**’ attribute property ids. The “**457378**” id will need changing to a ‘**Relationship List**’ attribute that references ‘**Business Actors**’ from the users client area.
- Both of the Application property ids, “**1104**”, and “**1107**” need to be changed to ‘**Date**’ attribute property ids. The “**457377**” id will need changing to a ‘**Relationship List**’ attribute that references ‘**Work Packages**’ from the users client area.

The object types can be substituted, but the example script expects that ‘Applications’ will have an attribute that is a list of ‘Projects’, which have an attribute that is a list of ‘Actors’. The example script also requires that the property types of ‘Date Range’ and ‘Date’ remain unchanged.

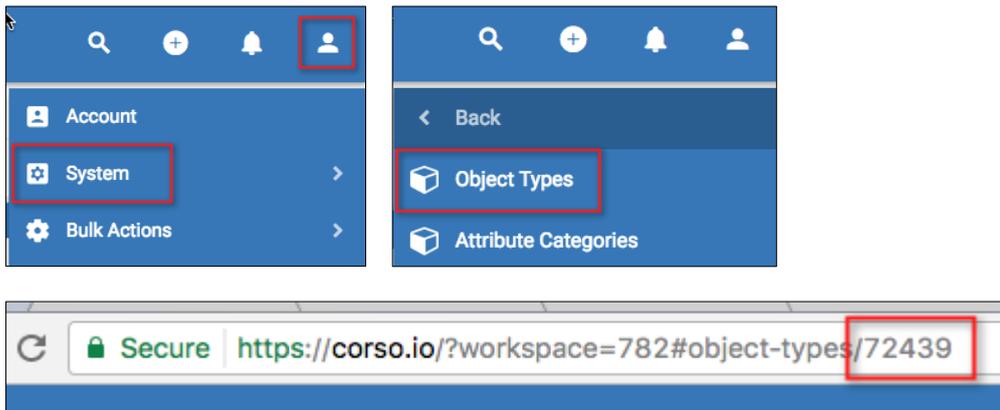
Users may need to create some of these attributes if they do not exist within the client area already.

*Below are some API calls to show how the needed script information (Object types, attribute and attribute property ID's) can be gathered.*

### Identifying Object Type IDs

In this example, three object type ids have been used; 692 (business actor), 736 (work package), and 719 (application component). These ID's will be unique to a particular users ‘client area’ and you will need to find them.

- To identify the id of any object-type in an open workspace in Agile EA; access the ‘User’ menu, its “System” submenu, and select the “Object Types” option.
- Upon selecting an object type such as “Application Component”, the displayed URL will look something like; <https://myerwin.io/?workspace=448#object-types/719>.



## Identifying Object Type Attributes

Although the `getActorNames` function (used in the script) checks for the existence of an “id” and “name”, it is expected that they exist. This expectation comes from identifying the properties of the Object Type by using an API URL, e.g. “<https://api.myerwin.io/object-types/692>”.

```
{
  "id": "692",
  "name": "Business actor",
  "description": "A business actor is an organizational entity that is capable of performing behavior. Examples are humans, departments and business units.",
  "logo": "spp-icon-mt-actor",
  "attributes": [...],
  "relations": [...]
}
```

The generated JSON shows that the object has “id” and “name” attributes, as well as “description”, “logo”, and “attributes” and “relations” arrays. Note that the detail of the “attributes” and “relations” has been hidden to simplify the output.

## Identifying Object Type Attribute Properties

In this example, the attributes are referenced using their “id” rather than their “name”. An extract of the “attributes” array from the URL returning object-type 736, is shown below;

```
"attributes": [
  {
    "id": "1289",
    "name": "SU. Starting up a Project",
    "objectTypeId": "736",
    "type": "DATE_RANGE_ATTRIBUTE"
```

---

```
  },
  {
    "id": "1293",
    "name": "CP. Closing a Project",
    "objectTypeId": "736",
    "type": "DATE_RANGE_ATTRIBUTE"
  },
  {
    "id": "457378",
    "name": "Manager",
    "objectTypeId": "736",
    "relationshipType": {
      "id": "39864"
    },
    "direction": "B2A",
    "relationshipTypeId": "39864",
    "type": "RELATION_LIST_ATTRIBUTE"
  },
],
```

It is also useful to view the JSON generated for an instance of the object-type. The following extract of the “attributes” object was taken from the JSON returned by the URL “<https://api.corso.io/objects/227093>”.

```
"attributes": {
  "1289": {
    "type": {
      "id": "1289",
      "name": "SU. Starting up a Project",
      "objectTypeId": "736",
      "type": "DATE_RANGE_ATTRIBUTE"
    },
    "value": {
```

---

```
    "start": "2016-01-01T17:05:00.000+0000",
    "end": "2017-06-30T16:05:00.000+0000"
  }
},
"1293": {
  "type": {
    "id": "1293",
    "name": "CP. Closing a Project",
    "objectTypeId": "736",
    "type": "DATE_RANGE_ATTRIBUTE"
  },
  "value": {
    "start": "2016-01-01T17:16:00.000+0000",
    "end": "2017-06-29T15:18:00.000+0000"
  }
},
"457378": {
  "type": {
    "id": "457378",
    "name": "Manager",
    "objectTypeId": "736",
    "relationshipType": {
      "id": "39864"
    },
    "direction": "B2A",
    "relationshipTypeId": "39864",
    "type": "RELATION_LIST_ATTRIBUTE"
  },
},
}
```

---

## Agile EA API JSON Output

### Pieces of the script to update

The test JSON data can now be substituted with data from Agile EA. This is achieved by JavaScript routines that are triggered by JSReport's 'beforeRender' event. The output from Agile EA can then be restructured to mirror the format of the test data.

### beforeRender()

When the template 'runs', it will execute the "beforeRender" function in the associated 'script'.

```
// Request data to be used in report before rendering.  
function beforeRender(req, res, done) {  
    erwinGlobals();  
    setData(req, done);  
}
```

In this example, 'global' values are established before setting the "data" object in "req". The "done()" function will be called upon completion of the script.

### erwinGlobals()

```
var erwin = {}; // Used to store global constants  
// Establish some global constants for source Workspace  
function erwinGlobals() {  
    var userName = "userName"; // The API will require a login  
    var password = "password"; // The password will be encoded  
    erwin.requestHeader = "Basic " + encodeCredentials(userName, password);  
    erwin.workspace = ""; // A workspace "id" can override the 'default'.  
}
```

It will be necessary to replace the "username" and "password", before any JSON will be returned. This detail will be encoded and used in the request header. Note that data will be returned from the 'default' workspace, unless a workspace "id" is used.

---

## setData()

The purpose of the script is to replace the “data” object in the ‘request’, with data from a workspace within Agile EA.

```
// Get the Report Data to overwrite 'req.data'
// 1. Request Actors - storing their names
// 2. Request Projects - updating the Manager's name
// 3. Request Applications - linking related Projects.
function setData(req, done) {
  var apiActors = getApiUrl("692"); // 692 = Business Actor
  require('request')(prepareRequest(apiActors), function(err, actorRes, body) {
    var actors = {};
    if (checkForError(req, err, actorRes, body, "Error requesting Actors:")) {
      done(); // terminate early
    } else {
      getActorNames(body, actors);
      processProjects(req, done, actors);
    }
  });
}
```

In this example the intension is to build a ‘collection’ of Business Actor Names, which will be used to update the “manager” detail of a collection of Projects, which in turn will be linked to the “project” detail of a collection of Applications.

A ‘URL’ is created from the id of the Business Actor object type, in this case “692”. This is then used to ‘prepare the request’ for JSON data.

If an error occurs, the “done” function is called, terminating the “beforeRender” event, otherwise the “actors” object is built from the detail in the “body”, and passed to the processing of the Projects.

---

## processProjects()

The processing of the Projects is similar to the Actors;

```
// Create a collection of Projects and update their manager's name.
function processProjects(req, done, actors) {
  var apiProjects = getApiUrl("736"); // 736 = Work Package (Project)
  require('request')( prepareRequest(apiProjects), function(err, projectRes, body) {
    var projects = {};
    if (checkForError(req, err, projectRes, body, "Error requesting Projects:")) {
      done(); // terminate early
    } else {
      getProjectDetail(body, projects, actors);
      processApplications(req, done, projects);
    }
  });
}
```

Projects are represented by Work Package object types; "736".

If no error occurs the "projects" object is built from detail in the "body" and names from the "actors" object, before being passed to the processing of the Applications.

## processApplications()

The processing of the Applications will conclude with the update of "req.data";

```
// Create a collection of Applications linked to associated Projects
function processApplications(req, done, projects) {
  var apiApplications = getApiUrl("719"); // 719 = Application Component
  require('request')( prepareRequest(apiApplications), function(err, appRes, body) {
    var applications = [];
    if (checkForError(req, err, appRes, body, "Error requesting Applications:")) {
      done(); // terminate early
    } else {
      getApplicationDetail(body, applications, projects);
    }
  });
}
```

---

```
nameSort(applications);  
// Substitute the req.data with the sorted application array object  
req.data = {  
    application: applications  
};  
}  
done(); // successful completion  
});  
}
```

The Application Component object type, “719” is used to request Application data.

If no error occurs the “applications” array is built from detail in the “body” and related Projects, before sorting, and being used to override the existing ‘test data’.

## getActorNames()

A simple function is required to extract the required “name” property from the detail returned from the request for Business Actors, and build a collection that can be accessed using its “id”.

```
// Extract the Names from the returned Actor detail.  
function getActorNames(body, actors) {  
    body.forEach(function (object) {  
        var actor = {};  
        if (object.id) {  
            if (object.name) {  
                actor.name = object.name;  
                actors[object.id] = actor;  
            }  
        }  
    });  
}
```

For each of the objects in the “body” that has an “id” and “name” property, create an object consisting of the “name” value keyed by the “id” value, and append it to “actors”.

---

## getProjectDetail()

In addition to its name, the Project requires a 'start date', 'end date', and the name of its manager. This additional detail comes from objects within the "attributes" array.

```
// Update the 'projects' object with details;
// Name, Start Date, End Date, and Manager.
function getProjectDetail(body, projects, actors) {
    var startDateAttributeld = "1289",
        endDateAttributeld = "1293",
        managersAttributeld = "457378";
    var namesIds = [];
    body.forEach(function (object) {
        var project = {};
        if (object.name) {
            project.name = object.name;
        }
        if (object.attributes[startDateAttributeld]) {
            project.startDate = getDate(object.attributes[startDateAttributeld].value.start);
        }
        if (object.attributes[endDateAttributeld]) {
            project.endDate = getDate(object.attributes[endDateAttributeld].value.end);
        }
        if (object.attributes[managersAttributeld]) {
            namesIds = getOtherIds(object.attributes[managersAttributeld].value, object.id);
            project.manager = getIdName(namesIds, actors);
        }
        projects[object.id] = project;
    });
}
```

Three variables have been assigned values that identify the required attributes. A "project" object is created for each of the Projects in the "body", and updated with a "name", "startDate", "endDate", and "manager". The dates are to be formatted differently to the way that they appear in the exported JSON, and the name of the manager(s) is to be returned from the "actors" using the "id" from the exported JSON.

---

## getApplicationDetail()

The detail required for the Application is its name, description, start and end dates, and the related Projects.

```
// Update the 'applications' object with details;
// Name, Description, Start Date, End Date, and related Projects.
function getApplicationDetail(body, applications, projects) {
    var startDateAttributeld = "1104",
        endDateAttributeld = "1107",
        projectsAttributeld = "457377";
    body.forEach(function (object) {
        var appObject = {};
        if (object.name) {
            appObject.name = object.name;
        }
        if (object.description) {
            appObject.description = object.description;
        }
        if (object.attributes[startDateAttributeld]) {
            appObject.startDate = getDate(object.attributes[startDateAttributeld].value);
        }
        if (object.attributes[endDateAttributeld]) {
            appObject.endDate = getDate(object.attributes[endDateAttributeld].value);
        }
        appObject.project = [];
        if (object.attributes[projectsAttributeld]) {
            objectValue = object.attributes[projectsAttributeld].value;
            linkProjects(appObject, objectValue, object.id, projects);
        }
        if (appObject.project.length === 0) {
            appObject.project = [{}];
        }
        applications.push(appObject);
    });
}
```

This function is very similar to GetProjectDetail() except that it contains additional code that builds the application's project array.

## An Example Report using data from the API

Once the script has completed, the report is rendered with the substituted data, and could look something like this;

<b>Project Context; Application Report</b>							
Application	Description	Start Date	End Date	Project	Project Start	Project End	Project Manager
Auto Insurance Application	Being replaced with the integrated back-office suite.	7/4/2016	1/19/2018	Back Office System Integration	1/1/2016	6/29/2017	Martha Kent
Back Office Suite	Integrated back-office automation replaces the separate back-office applications.	6/1/2017	8/31/2020	Back Office System Integration	1/1/2016	6/29/2017	Martha Kent
Call Center Application							
Data Warehousing Solution	A completely new data warehousing solution is to be introduced.	1/1/2018	12/31/2020	Data Warehousing	9/1/2016	9/30/2020	James Olsen
Document Management System	The document management system is to be replicated on its main server and back-up server.						
General CRM System	A general CRM system is to be used by the entire organization.	2/6/2017	12/18/2020	CRM System Integration	1/1/2016	6/30/2017	Perry White
Home & Away Financial Application	Home & Away currently uses its pre-merger policy administration and financial applications.	2/2/2015	10/27/2017	Back Office System Integration	1/1/2016	6/29/2017	Martha Kent
Legal Expense Back Office System	A completely new back-office application suite is to be introduced.	1/1/2017	11/30/2018	Back Office System Integration	1/1/2016	6/29/2017	Martha Kent
Legal Expense CRM System	The CRM functionality for Legal Expense insurance customers is to be taken over by the general CRM system.	1/4/2016	9/29/2017	Back Office System Integration	1/1/2016	6/29/2017	Martha Kent
				CRM System Integration	1/1/2016	6/30/2017	Perry White
Web Portal	The three divisions will adopt a common web portal.						

---

# EA agile CSV Import/Export API (with Postman)

erwin recently added a write API that allows Professional licensed users, to write data to the EA agile platform, programatically.

Watch the [video](#) to see this in action - With some more technical information below

## Object Import API – CSV/ZIP files

### Single or Multiple Objects of the same Object Type

- API Request Type: Post / multipart/form-data
- Security: Basic Auth [Username and Password]
- Scope: Workspace
- Object Collision Detection: Object Type + Name
- Object Creation: Yes (if no collision)
- Object Update: Yes (if collision) – Only supplied properties are overwritten. If empty value passed, no change to target property
- Object Deletion: ?? TBC
- Returns: JSON describing the success state of each item
- Querystring:
- workspace - workspace id of the workspace to import into.
- CSV file
  - Must contain; Name and Object Type columns in the CSV to import
  - Must have the extension ZIP if compressed, CSV or Text will be treated as a CSV file
- CSV file [raw example]:  
Content-Disposition: form-data; name="file"; filename="20170530092014\_Feature\_Objects.csv"  
Content-Type: text/csv  
Name, Object Type, Description, Communities, Tags  
Test object 1,Feature,Test desc, All,""
- Output: JSON results of the import.
- Example: POST <http://myerwin.io/api/object-import?workspace=123>

---

## Relationships Import API – CSV/ZIP files

### Single or Multiple Relationships of any Relationship Type

- API Request Type: Post / multipart/form-data
- Security: Basic Auth [Username and Password]
- Scope: Workspace
- Relationship Collision Detection: Association Type + Name + From Object + To Object
- Relationship Creation: Yes (if no collision)
- Relationship Update: No – Relationships carry no content
- Relationship Deletion: ?? TBC
- Returns: JSON describing the success state of each item
- Querystring:
  - workspace - workspace id of the workspace to import into.
  - CSV file
    - Must contain; “Relationship Source Name”, “Relationship Target Name”, “Source Object Name”, “Target Object Name”, “Source Object Type Name”, “Target Object Type Name” columns in the CSV to import
    - Must have the extension ZIP if compressed, CSV or Text will be treated as a CSV file
    - File should be called “Relationships.csv”
- CSV file [raw example]:

Content-Disposition: form-data; name="file"; filename="Relationships.csv"  
Content-Type: text/csv  
Relationship Source Name, Relationship Target Name, Source Object Name, Target Object Name,  
Source Object Type Name, Target Object Type Name  
composes, is composed of, Unified views across the product data, Unification of Object Content  
across products, Epic, Feature
- Output: JSON results of the import.
- Example: POST <http://myerwin.io/api/object-import?workspace=123>

### Combined Import API – ZIP files

- Using a ZIP compressed file, it is possible to combine multiple different Object Types, and Relationships in a single import
- ZIP structure:
  - Object Type 1.csv
  - Object Type 2.csv
  - Relationships.csv
- When the ZIP containing multiple files is received, it is decompressed, and imported one at a time (object type files first).
- As for individual CSVs a response file is returned indicating the success state of the operation

---

## **Object and Relationship Export**

### **Export from a View**

Watch the [video](#) to see this in action

### **View Export API – Zip of CSV files**

#### **Object and Relationship Export – Export Views**

- API Request Type: Post / multipart/form-data
- Security: Basic Auth [Username and Password]
- Scope: View on a Workspace
- Returns: binary ZIP file
- Querystring:
  - workspace - workspace id of the workspace to export
  - view – view id of the view to export
- Output: raw binary ZIP file containing all the CSVs that make up the view
- Example: POST <http://erwin.io/api/object-export?workspace=123&view=456>

---

# MMF API

For more information on our API please visit: <https://developer.myerwin.io/>

---

# About Us

Quest creates software solutions that make the benefits of new technology real in an increasingly complex IT landscape. From database and systems management, to Active Directory and Office 365 management, and cyber security resilience, Quest helps customers solve their next IT challenge now. Around the globe, more than 130,000 companies and 95% of the Fortune 500 count on Quest to deliver proactive management and monitoring for the next enterprise initiative, find the next solution for complex Microsoft challenges and stay ahead of the next threat. Quest Software. Where next meets now. For more information, visit [www.quest.com](http://www.quest.com).

## Technical support resources

Technical support is available to Quest customers with a valid maintenance contract and customers who have trial versions. You can access the Quest Support Portal at <https://support.quest.com>.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request
- View Knowledge Base articles
- Sign up for product notifications
- Download software and technical documentation
- View how-to-videos
- Engage in community discussions
- Chat with support engineers online

View services to assist you with your product.