



CARA Version 5.6

Configuration Manual

16 February 2022

This document is strictly confidential
and may only be read by licensed or
evaluating customers.

Table of Contents

1.	Introduction to CARA Version 5	4
2.	Configuration Introduction	8
3.	Administration	13
3.1.	Authentication Tokens	13
3.2.	Groups.....	17
3.3.	Jobs	22
3.4.	Methods.....	31
3.5.	Queues	34
3.6.	Sessions	38
3.7.	Spaces and Multi-Tenancy.....	39
3.8.	Type Definition.....	43
3.9.	Users	55
4.	Data.....	64
4.1.	Dictionaries.....	64
4.2.	Taxonomies	74
5.	Maintenance	81
5.1.	Configuration Export	81
5.2.	Configuration Import	85
5.3.	Configuration Tags.....	88
6.	General Configuration.....	90
6.1.	ACL Templates.....	90
6.2.	Capabilities	99
6.3.	Collaboration Services	102
6.4.	Connectors	107
6.5.	Custom Actions.....	110
6.6.	Dashboards	123
6.7.	eForms	126
6.8.	Lifecycles	131
6.9.	Notification Channels	139
6.10.	Notification Templates.....	141
6.11.	Queue Consumers	145
6.12.	Relation Types	148
6.13.	Reports	153
6.14.	Sequences	166
6.15.	Signing Services.....	169
6.16.	Workflow Templates	175
6.17.	Workspaces	200

7.	Type Configuration.....	204
7.1.	Type Capabilities	206
7.2.	Forms	210
7.3.	Views - General	234
7.4.	View - Actions	238
7.5.	View - Aggregations.....	241
7.6.	View - Dimensions	246
7.7.	View - Grid	251
7.8.	View - Map View	259
7.9.	View - Search	261
7.10.	View - Widgets	266
7.11.	Classification Definition	272
7.12.	Classification Restrictions.....	275
7.13.	Lifecycle Assignment.....	278
7.14.	Properties Extraction	280
7.15.	Properties Inheritance	282
7.16.	Properties Initialization	284
7.17.	Template Assignment.....	286
7.18.	Auditing	288
7.19.	Auto-Values.....	296
7.20.	Change Notifications	302
7.21.	Content Access Configuration	306
7.22.	Date Based Notifications	310
7.23.	Legal Hold Items	314
7.24.	Linking.....	317
7.25.	Purge Configuration	323
7.26.	Queuing	326
7.27.	Security	330
8.	Configuration History, Restore & Audit.....	333
9.	Supported Locales	336
10.	CARA Query Language - CQL	338
11.	Query Editor	343
12.	Scripting Utilities.....	345
13.	CARA Condition Language - CCL	348
14.	CARA System Object Model.....	350
15.	Configuration Files	356

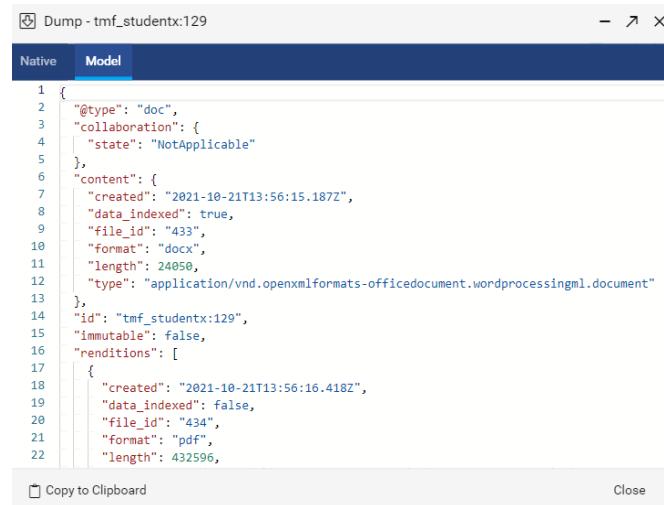
1. Introduction to CARA Version 5

Introduction

CARA is a next generation content management platform. It is one of the fastest, most configurable and user-friendly systems on the market. CARA version 5 uses the Elasticsearch application platform. Elasticsearch provides multiple benefits over many traditional platforms, including increased flexibility and potentially large speed improvements. Elasticsearch is based on Lucene, an open source Java based non-relational database.

Elasticsearch Concepts and Terminology

In Elasticsearch each individual document is stored as a JSON format file. JSON (JavaScript Object Notation) is a very common, widely adopted open-standard file format. The JSON file contains both the document's system and custom properties as well as content text. This eliminates the limitations and complexity of maintaining a separate index server when users need to search within the text content of a document. Full text index functionality is no longer an optional extra, it is a fundamental built-in feature.



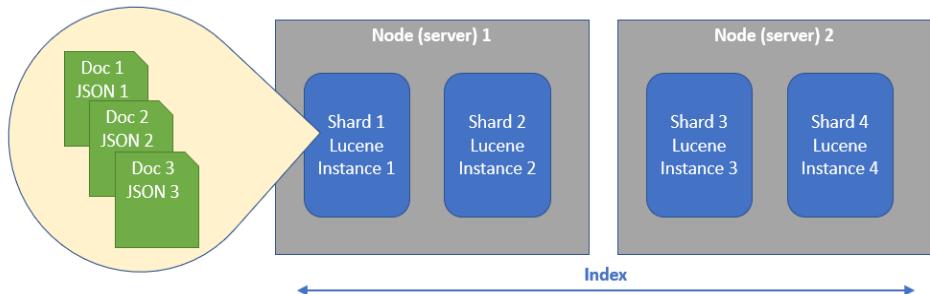
The screenshot shows a window titled "Dump - tmf_studentx:129". It has two tabs: "Native" and "Model", with "Model" selected. The content area displays a JSON document with the following code:

```
1 {
2   "@type": "doc",
3   "collaboration": {
4     "state": "NotApplicable"
5   },
6   "content": {
7     "created": "2021-10-21T13:56:15.187Z",
8     "data_indexed": true,
9     "file_id": "433",
10    "format": "docx",
11    "length": 24050,
12    "type": "application/vnd.openxmlformats-officedocument.wordprocessingml.document"
13  },
14  "id": "tmf_studentx:129",
15  "immutable": false,
16  "renditions": [
17    {
18      "created": "2021-10-21T13:56:16.418Z",
19      "data_indexed": false,
20      "file_id": "434",
21      "format": "pdf",
22      "length": 432596,
```

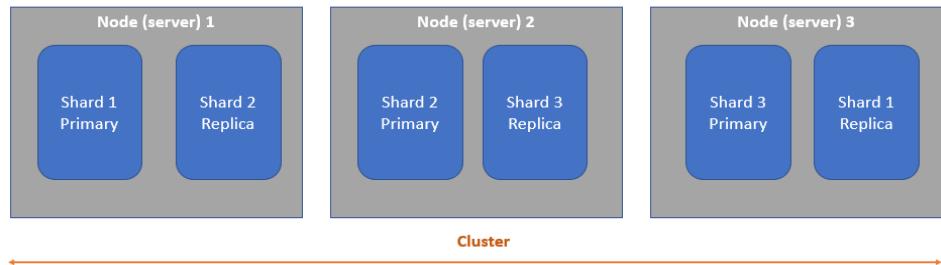
At the bottom of the window, there are buttons for "Copy to Clipboard" and "Close".

The JSON files of multiple similar documents form an index. Traditionally the data set, or index as referred to by Elasticsearch, is located on a single database server. Traditional database servers can be vertically scaled as needed, such as adding more storage, RAM or CPU processing power, but vertical scaling has diminishing returns.

With Elasticsearch the index is designed to be split across servers, providing horizontal scaling, this is a core design purpose. The index is split into multiple shards. Each shard is a stand-alone instance of Lucene.



A shard can either be a primary shard, which provides search results, or a replica/backup of a primary shard.

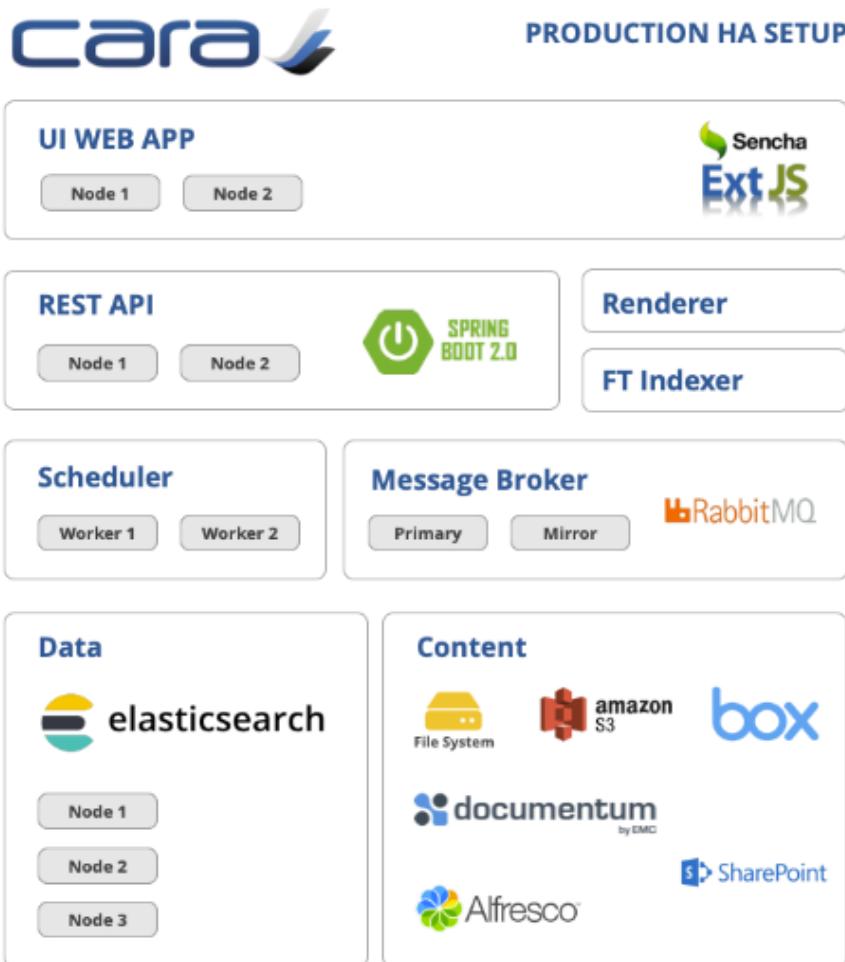


In Elasticsearch a server is referred to as a node. Each server/node can contain multiple shards. Those shards can belong to different indexes and/or be primary or backup shards.

It is not compulsory to include replica shards for all indexes, but they are considered a standard feature of an implementation. The number and distribution of primary and backup shards is entirely configurable. Usually a node will contain at least 1 primary and 1 replica shard.

A collection of nodes is referred to as a cluster. All nodes in a cluster share a common cluster name.

**New CARA
Platform
Architecture**



- Elasticsearch provides both data storage and business logic services. It can replace legacy platform servers such as the Documentum Content Server and the need for a 3rd party database such as Oracle or MS SQL. It is by design highly scalable and resilient.
- Content files are stored and managed in your choice of content storage. Connectors can be configured to use a combination of existing enterprise solutions such as Box, cloud services such as Amazon S3 or a legacy platform including Documentum. They could also make use of a simple file system share.
- Integration with other data sources and services is primarily provided via the REST API, which is based on Spring Boot 2.0
- The web based user interface is built upon the Ext JS application framework. It can be deployed on a wide range of application

servers, also within a distributed architecture for both resilience and scalability.

- Rendition services are fully integrated, providing the automatic transformation and distribution of source documents into appropriate renditions formats such as PDF and can include the seamless application of content related services such as water marking, lifecycle tracking and the recording of electronic signatures.

2. Configuration Introduction

Description

The configuration and administration of a CARA v5 environment is carried out using the options provided in the Control Panel area of the user interface. This manual describes and details those options. The sections and chapters within this manual are presented in the order they appear within the control panel:

Name	Display Name
_admin	Administrator
_all	Guests
_analytics_manager	Analytics Manager
_bulk_importer	Bulk Importer
_config_manager	Config Manager
_ft_indexer	FullText Indexer
_group_admin	Group Administrator
_group_manager	Group Manager
_query_executor	Query Executor

Access to the control panel is provided via membership of a system group such as `_admin` or `_config_manager`. Please see the groups chapter for details. Alternatively, workspace configuration can be used to provide administrative users with highly restricted control panel access in order to update only specific configuration elements, such as individual dictionaries or workspace templates. Please see the workspaces chapter for details.

Both initial setup and updates to configuration are intended to be streamlined and intuitive and are therefore carried out almost entirely within the GUI based control panel. On the few occasions where a configuration option involves direct updates to files on the server, examples of those files are provided in the Configuration Files chapter.

Recommended Configuration Order

CARA provides an extensive range of configuration options. The following is a recommended order to follow when configuring core elements of a new CARA v5 environment or adding a new business scenario to an existing environment, taking into consideration dependencies between elements. Additional configuration options are then applied based on specific business requirements.

1. Create Users and Groups.

Users and groups can imported automatically via an LDAP connection or created manually. Various password options are available for users including inline encrypted passwords, single sign-on and LDAP based passwords. Multifactor authentication is supported for both user logins and event signatures.

2. Set User System Capabilities.

System capabilities refer to options which apply across the application, such as access to the inbox or log files. This is in contrast to type-specific user capabilities such as the ability to create or view selected document types. Users are added to groups and those groups have specific system capabilities enabled or denied.

3. Define at Least 1 Type with Custom Attributes.

Types correlate to the distinct document and other object types that users will work with. A particular department may have a single type of document they work with or many. For each type define the attributes of the type. All CARA v5 types have a set of core attributes by default such as name and created date. Other attributes are added as required. Once a type has been initially defined and saved, it becomes available for further configuration in the Types panel of the Control Panel.

4. Create Dictionaries and Taxonomies for the Main Classification.

Dictionaries provide the basis of drop-down lists from which users may select property values for documents and other

objects. Dictionaries are also used to present users with classification choices when creating new documents and objects. Within CARA a taxonomy is where 2 or more dictionaries are related, the choice a user makes from 1 dictionary determines the choices available in subsequent dictionaries.

5. Set User Type Capabilities.

Select the type-specific capabilities which are available to each group of users who will have access to that type. A group of users may have the capability to create documents of one type but to only view documents of another type.

6. Update the Type Properties Form.

When users create, update or view documents, the properties form displays the document's attributes, which users may be expected to update manually or may be updated automatically by auto-value rules.

A properties form is automatically generated for a type when that type is initially defined. The form initially only contains a single attribute: object_name. Additional properties are added and a layout designed. For each attribute on the form an appropriate form control is chosen, such as plain-text entry, drop-down list, radio buttons etc.. Each attribute can have multiple conditions set, such as when the attribute is visible, editable or compulsory.

7. Configure the Default View.

The CARA user interface is highly configurable and is tailored to each document or object type as a distinct view. A single document or object type can have multiple views, for example the view provided to authors of a type can be different from that provided to the readers. Users can have access to a single view or multiple views.

8. Add the View To a Workspace.

Views are organised into workspaces and groups are given access to the workspaces relevant to them. Workspaces can

also be used to provide administrators with limited access to update type specific configuration.

9. Associate a Taxonomy with the Classification Definition.

Select the taxonomy which will provide the users' classification choices when creating documents or objects of that type.

10. Setup Template Assignment and Import Content Templates.

Import content templates, such as documents created in MS Word or Excel. Content templates are associated with a document classification when imported.

When multiple versions of a template exist, select which will be provided to users.

Content templates can also be used to set default attribute values.

11. Add Linking Rules.

Documents can be placed in folders automatically using linking rules. Create linking rules that select the folder path based on document classification or other properties. Linking rules create the required folders as needed.

Alternatively users can be allowed to manually select the folder path, in which case linking rules are not required.

12. Set Type Security.

Create ACL's which grant permissions to groups of users or individual users. ACL's can be static or dynamic.

Create the security rules that determine which ACL will be applied when a document or other object is created or updated. A single security rule could be used for all documents or objects of a type, or multiple conditional security rules could be used per type.

13. Create and Assign Lifecycles.

If documents will pass through a series of lifecycle states such as draft, approved and superseded, define the lifecycles and assign

which will be applied for new documents or objects of a particular type.

Lifecycles are commonly used but can be considered optional.

14. Create Workflow Templates.

Workflow templates are used to assign tasks to users, such as to review or approve a document. Creating a workflow template includes setting task steps, recipient assignment rules, instructions and notifications. Workflow templates are often combined with lifecycles.

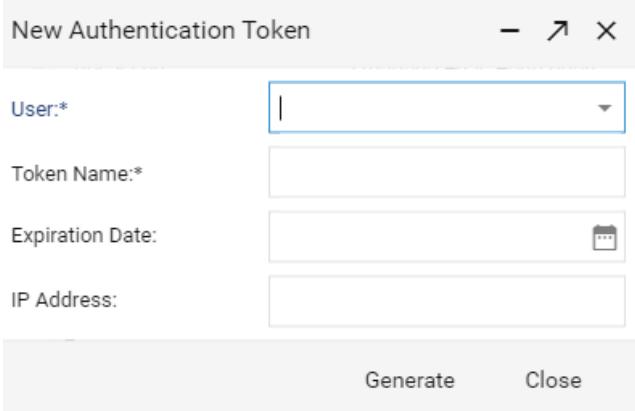
Workflow templates, which are optional, can be type specific or general purpose.

15. Backup Configuration.

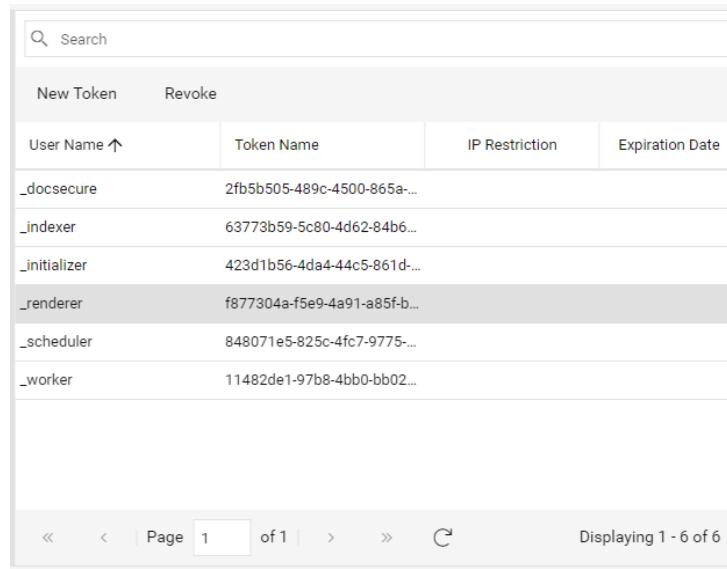
CARA provides extensive options for exporting and importing configuration. The current configuration can be exported or configuration as it was at a named point in time. Configuration can be exported and imported as a whole or as individual elements.

3. Administration

3.1. Authentication Tokens

Description	<p>Authentication tokens are used to allow external applications to connect securely to CARA without providing explicit username and password credentials. For example a custom rendering engine could use an authentication token to connect to CARA when processing rendition requests.</p> <p>Authentication tokens are generated and stored internally within CARA and optionally can be restricted to a dedicated IP address. They are tied to a CARA user account which must be created in advance. They are typically generated with an expiry date. Authentication tokens can be revoked by CARA administrators at any time.</p>
Creating Authentication Tokens	 <ol style="list-style-type: none">From the CARA Control Panel select Administration > Auth Tokens.The Auth Tokens panel opens, displaying all currently active authentication tokens. For each authentication token the following information is displayed: User Name that the token is tied to.

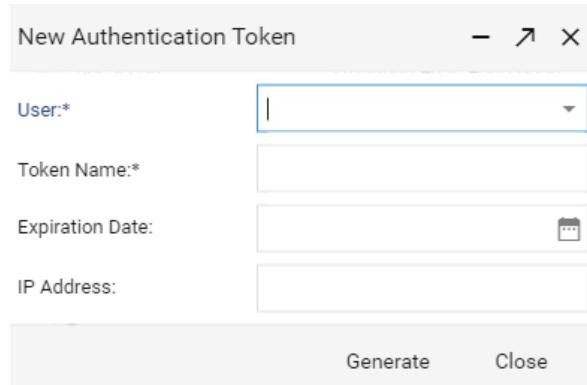
- The **Token Name**.
Any **IP Restriction**, if set.
An **Expiration date**, if set.



A screenshot of a web-based application interface titled "New Token". At the top, there is a search bar with a magnifying glass icon and the placeholder text "Search". Below the search bar, there are two buttons: "New Token" and "Revoke". The main area is a table with four columns: "User Name ↑", "Token Name", "IP Restriction", and "Expiration Date". The table contains six rows of data. The first row is highlighted with a grey background. The columns show the user names and their corresponding token IDs. The last row shows the user name as "_worker" and the token ID as "11482de1-97b8-4bb0-bb02...". At the bottom of the table, there is a navigation bar with icons for "Page", "1", "of 1", and "Displaying 1 - 6 of 6".

User Name ↑	Token Name	IP Restriction	Expiration Date
_docsecure	2fb5b505-489c-4500-865a...		
_indexer	63773b59-5c80-4d62-84b6...		
_initializer	423d1b56-4da4-44c5-861d...		
_renderer	f877304a-f5e9-4a91-a85f-b...		
_scheduler	848071e5-825c-4fc7-9775...		
_worker	11482de1-97b8-4bb0-bb02...		

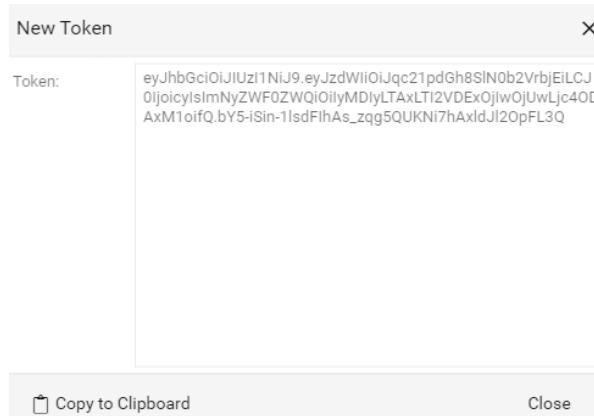
3. Click **New Token**, the **New Authentication Token** window opens:



A screenshot of a modal dialog box titled "New Authentication Token". The dialog has a standard window title bar with minimize, maximize, and close buttons. Inside, there are four input fields: "User:" with a dropdown arrow, "Token Name:" with an empty input field, "Expiration Date:" with a date picker icon, and "IP Address:" with an empty input field. At the bottom right of the dialog are two buttons: "Generate" and "Close".

4. Please note that the token properties cannot be modified once created, as those properties are used by the algorithm when generating the token. If properties need to be updated, such as an IP address restriction being added, a new authentication token will have to be generated and passed to the 3rd party, and the old token revoked.

5. Enter the **User Name** of the user which the authentication token will be tied to. The username must match a valid existing user within CARA. The user must have the property Source set to System.
6. Enter a **Token Name**.
7. Optionally select an **Expiration Date**. While not required it is considered standard practice to limit the duration of a token.
8. Optionally enter an **IP Address** which the authentication token will be tied to. If specified, CARA will only accept connections from the authentication token which originates from that IP address.
9. Click **Generate**, the new token is generated and displayed in a separate window:

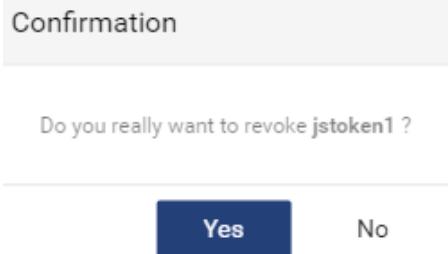


10. Either use **Copy to Clipboard** or manually select the token string and copy. Paste the token string into a suitable format for providing to the 3rd party. Please note that once the window has been closed, for security purposes the token string cannot be viewed again.
11. **Close** the **New Token** window.

Revoking a Token

12. The 3rd party provided with the token should make use of it within the CARA REST API, it can either be hardcoded or read from a file.

1. Tokens can be revoked by highlighting them in the **Auth Tokens** window and selecting **Revoke**.
2. Click **Yes** to the confirmation message:



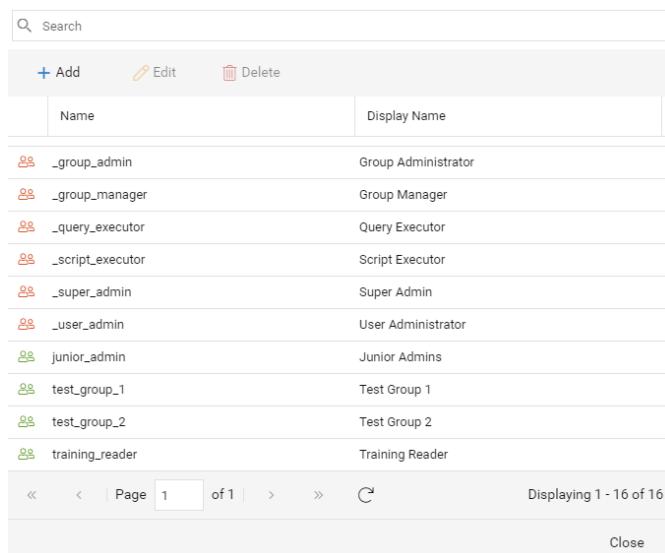
3. The token is revoked. Please note that any services associated with the token will no longer be able to connect to CARA.

3.2. Groups

Description

Groups are used throughout CARA to determine the functionality and document access available to users. Groups can be created and managed manually or automatically from an LDAP connection.

Several groups are created during installation, these system groups are identified by a name which begins with an underscore and are listed in red. User created group names cannot start with an underscore. Only members of the _admin group can modify membership of system groups.



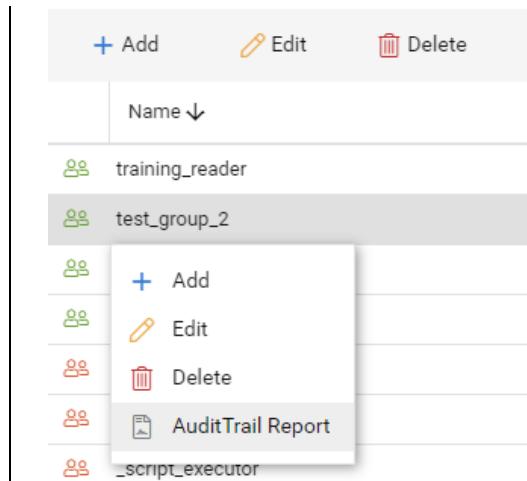
Search	
+ Add Edit Delete	
Name	Display Name
@ _group_admin	Group Administrator
@ _group_manager	Group Manager
@ _query_executor	Query Executor
@ _script_executor	Script Executor
@ _super_admin	Super Admin
@ _user_admin	User Administrator
@ junior_admin	Junior Admins
@ test_group_1	Test Group 1
@ test_group_2	Test Group 2
@ training_reader	Training Reader

<< < Page 1 of 1 | > >> Close

Displaying 1 - 16 of 16

The **Groups** window displays a list of groups alphabetically by **Name** or **Display Name**. Groups can be searched for in the search bar located at the top of the panel.

The **Audit Trail Report** for a highlighted group is available from the right-click menu:



The Audit Trail Report shows changes to the group such as members being added or removed:

A screenshot of the 'Audit Trail Report - training_reader' window. On the left is a 'Filters' sidebar with sections for 'Last Year', 'Countries' (United Kingdom), 'Events' (System: Update, System: Create), 'Modified Attribute', 'Types' (Group), and 'Users'. On the right is a table titled 'Events' with columns: Event, User Name, Time Stamp, Reason, and Signature. The table lists 13 entries, all of which are 'System: Update' events by 'Barry Prince' occurring between June 2021 and May 2022.

Event	User Name	Time Stamp	Reason	Signature
System: Update	Barry Prince	9/17/21 1:25 PM		
System: Update	Barry Prince	9/17/21 1:25 PM		
System: Update	Barry Prince	9/17/21 1:24 PM		
System: Update	Barry Prince	8/6/21 4:30 PM		
System: Update	Barry Prince	8/6/21 4:29 PM		
System: Update	Barry Prince	8/6/21 4:28 PM		
System: Update	Barry Prince	6/7/21 10:47 AM		
System: Update	Barry Prince	6/7/21 10:40 AM		
System: Update	Barry Prince	5/24/21 9:43 AM		
System: Update	Barry Prince	5/24/21 9:42 AM		
System: Update	Barry Prince	5/24/21 9:41 AM		
System: Create	Barry Prince	3/24/21 2:29 PM		

Right-click an entry and select **Details** to see further details and the changes made in a separate window:

Audit Trail – Details	
Properties Change	
Added	
Attribute	Value
/admins/0	student1
/description	Training Reader
/id	109
/name	training_reader
/source	Inline
/type	Group
/user_members/0	Barry Prince
/user_members/1	student1

Close

Creating Groups Manually

1. From the **CARA Control Panel** select **Administration > Groups**.
2. In the **Groups** panel select **+ Add**.

 Group Form

Group Name:*	<input type="text"/>
Display Name:	<input type="text"/>
Source:	Inline X ▾
User Members:	<input type="text"/> ▼ 
Group Members:	<input type="text"/> ▼ 
Administrators:	<input type="text"/> ▼ 

Save Cancel

3. In the **Group Form** window enter a unique **Group Name**. This field is required. Only lowercase letters, numbers and underscores are allowed. Group Name must start with a letter.
4. Optionally enter a group **Display Name**.

5. **Source** refers to whether the group was created and is managed manually (Inline) or automatically via an LDAP connection. For LDAP based groups the additional fields **LDAP ID** (identifier) and **LDAP DN** (Domain Name) will also be displayed.
6. **User Members** is the list of users who are members of the group. To add users click the right-hand side drop-down arrow for a list of users or the pencil icon to open the separate **User Selection** window.

Move users to the right-hand panel to add them to the group, either by double-clicking or using the central arrows. Users can be searched for using the search field at the top of the window. Click **Save** once complete.
7. Groups can be nested, that is groups can be members of other groups. **Groups Members** displays the groups which are members of the group. Add groups from the **Group Selection** window which is opened by clicking the right-hand side drop-down arrow.
8. **Administrators**. Users who are allowed to change the membership of the group.
9. **Save** the group once configured.

Primary System Groups

_super_admin. Members have access to user sessions, logs, clear cache, authentication tokens, type definitions and queues. It is the only group that can modify system group access. In a multi-tenant root environment they have access to spaces management.

_admin. Members have access to users, groups, jobs and methods. They cannot modify system group access.

_config_manager. Provides members with access to the control panel. Access within is determined by other group membership.

_analytics_manager. Provides members with access to the analytics window.

_query_executor. Provides members with access to the query editor.

_script_executor. Provides members with access to the script editor.

_user_admin. Members have access to the users section of the control panel. Members can add and edit users, however they cannot provide or modify user access to system groups.

_group_admin. Members have access to the groups section. Members can delete non-system groups. They can also create new groups and can change non-system group membership.

_group_manager. Members have access to the groups section. Members cannot delete any groups. They can create new groups and can change non-system group membership.

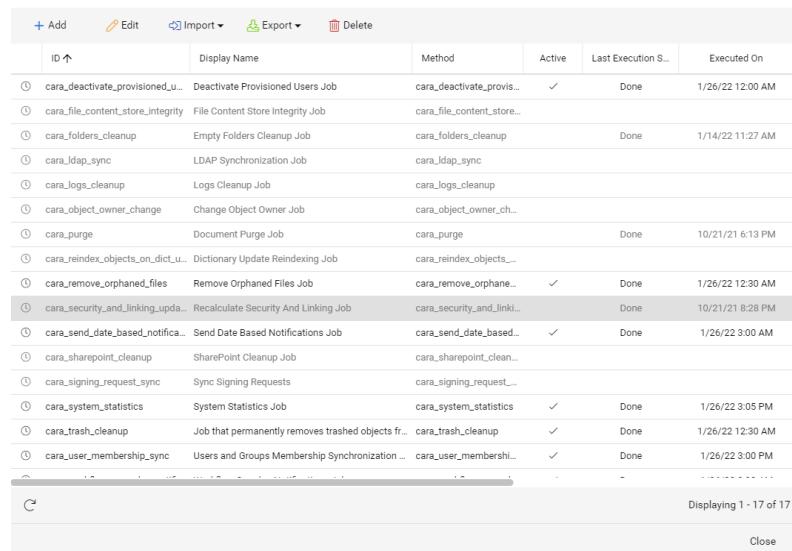
Note that the ability to update individual configuration elements such as those for type configuration, dictionaries and workspaces can be provided to workspace administrators, see the workspaces chapter for details.

3.3. Jobs

Description

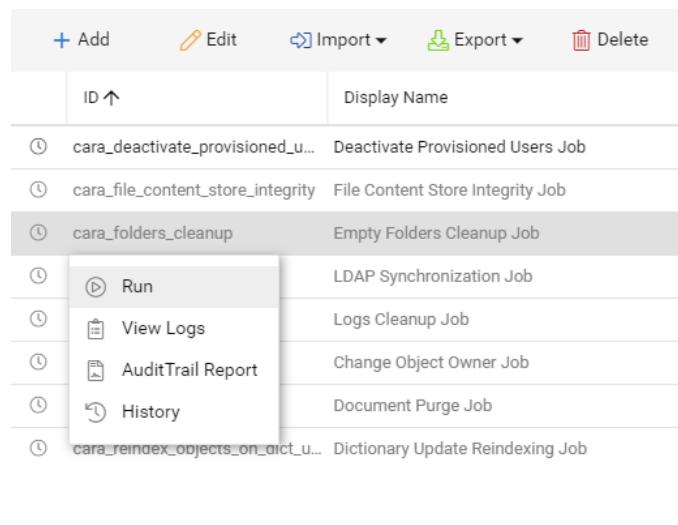
A job is a method which has been scheduled to run automatically.

The **Jobs** window lists each job with details including which jobs are active, when they were last executed, and when they are next due to be executed:



The screenshot shows a table titled "Jobs" with the following columns: ID, Display Name, Method, Active, Last Execution S..., and Executed On. There are 17 rows of data, each representing a different job. The jobs listed include: Deactivate Provisioned Users Job, File Content Store Integrity Job, Empty Folders Cleanup Job, LDAP Synchronization Job, Logs Cleanup Job, Change Object Owner Job, Document Purge Job, Dictionary Update Reindexing Job, Remove Orphaned Files Job, Recalculate Security And Linking Job, Send Date Based Notifications Job, SharePoint Cleanup Job, Sync Signing Requests, System Statistics Job, Job that permanently removes trashed objects fr..., and Users and Groups Membership Synchronization ...

Jobs which are not active and so will not run automatically are listed in grey. All jobs, including those which are inactive, can be run manually by selecting **Run** from the right-click menu:



The screenshot shows a table titled "Jobs" with the following columns: ID, Display Name. A context menu is open over the row for "cara_folders_cleanup". The menu options are: Run, View Logs, AuditTrail Report, and History. The "Run" option is highlighted. The "cara_folders_cleanup" row is also highlighted in the table.

Jobs with status queued or in progress have an additional menu option to Reset Job State, which is provided for use in cases where a job is unable to complete as expected.

Jobs for system methods are automatically present within CARA and do not need to be created manually, however their configuration options can be updated as required, for example to set when they run or which users will be notified when the job completes.

New jobs for custom methods can be created manually.

Creating and Modifying Job Configuration

1. From the CARA Control Panel select **Administration > Jobs**.
The **Jobs** panel lists all current jobs.
2. To modify the configuration of an existing job double-click the job. To create a new job select **+ Add**.

All jobs contain the following configuration settings:

The screenshot shows the 'New Job' configuration dialog box. It includes fields for ID, Display Name, Method, Status, Mode (Manual or Scheduled), Schedule (Mode: Daily, Every: Day, Starts At (UTC): 12:00 AM), Managers, Notifications (On Successful Completion, On Failure), and Parameters. At the bottom are Save and Cancel buttons.

Name	Type	Value				
ID	Text	[Empty]				
Display Name	Text	[Empty]				
Method	Dropdown	[Empty]				
Status	checkbox	Active				
Mode	radio buttons	Manual (radio) Scheduled (radio)				
Starts At (UTC)	Text	12:00 AM				
Managers	Dropdown	[Empty]				
Notifications	checkboxes	On Successful Completion (checkbox) On Failure (checkbox)				
Parameters	Table	<table border="1"><thead><tr><th>Name</th><th>Value</th></tr></thead><tbody><tr><td>[Empty]</td><td>[Empty]</td></tr></tbody></table>	Name	Value	[Empty]	[Empty]
Name	Value					
[Empty]	[Empty]					

Buttons at the bottom: Save (blue button) and Cancel.

- a. **ID.** The ID can only contain lowercase letters, numbers, underscores and hyphens. It must start with a letter.
- b. **Display Name.** The Display Name can contain mixed-case letters and spaces.
- c. **Method:** Select the method that the job will run. Once a method has been selected, clicking the pencil icon will open the method's configuration in a separate window.
- d. **Status.** Tick **Active** if the job is to run automatically.
- e. **Mode.** Select **Manual** or **Scheduled**. Only scheduled jobs run automatically.
- f. **Schedule.** Select the time interval and start point or frequency. The options available depend upon the **Mode** selected.
- g. **Managers.** Users who are selected in the Managers field are automatically sent notifications each time the job runs, whether the job has been run automatically or on demand. Users can be selected individually or as members of a group. The `_job_completed` event must be set to active within the `_notification_events` dictionary, and a notification template must be associated with the `_job_completed` event.
- h. **Notifications.** Select whether job completion notifications should be sent to the managers listed above **On Successful Completion** and/or **On Failure** of the job.
- i. **Parameters.** Parameters are job specific. For example, the `cara_trash_cleanup` job has a **threshold** parameter which determines how days a document must be in the trash bin before it is deleted:

The screenshot shows the 'cara_trash_cleanup - Jobs' configuration window. It includes the following fields:

- ID:** cara_trash_cleanup
- Display Name:** Job that permanently removes trashed objects from the system
- Method:** Trash Cleanup Method (cara_trash_cleanup)
- Status:** Active (checked)
- Mode:** Manual (radio button) and Scheduled (radio button selected)
- Schedule:**
 - Mode:** Daily
 - Every:** Day (radio button selected)
 - Starts At (UTC):** 12:30 AM
- Managers:** Barry Prince
- Notifications:** On Successful Completion (checkbox), On Failure (checkbox)
- Parameters:**

Name	Value
threshold	1

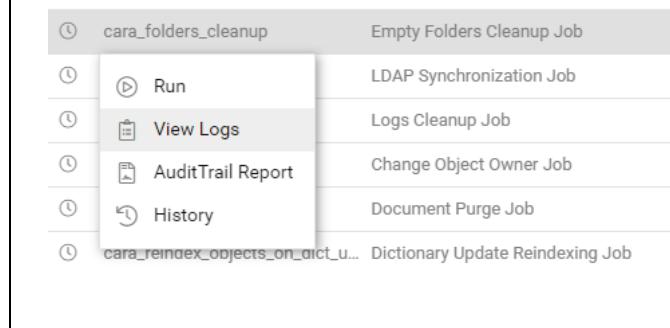
At the bottom are 'Save' and 'Cancel' buttons.

- Click **+ Add** to add a new parameter or double-click an existing parameter to modify and then click **Update** to save the change.

- Save** the job once configured.

Viewing Job Logs

The **Job Log** window is accessed by right-clicking a job and selecting **View Logs**:



The Job Logs window lists the **Status**, when the job was queued, when the job was started, ended and the user who ran the method:

Status	Queue Time	Start Time	End Time	Run By
Done	1/26/22 3:00 AM	1/26/22 3:00 AM	1/26/22 3:00 AM	_scheduler
Done	1/25/22 3:00 AM	1/25/22 3:00 AM	1/25/22 3:00 AM	_scheduler
Done	1/24/22 3:00 AM	1/24/22 3:00 AM	1/24/22 3:00 AM	_scheduler
Done	1/23/22 3:00 AM	1/23/22 3:00 AM	1/23/22 3:00 AM	_scheduler
Done	1/22/22 3:00 AM	1/22/22 3:00 AM	1/22/22 3:00 AM	_scheduler
Done	1/21/22 3:00 AM	1/21/22 3:00 AM	1/21/22 3:00 AM	_scheduler
Done	1/20/22 3:00 AM	1/20/22 3:00 AM	1/20/22 3:00 AM	_scheduler
Done	1/19/22 3:00 AM	1/19/22 3:00 AM	1/19/22 3:00 AM	_scheduler
Done	1/18/22 3:00 AM	1/18/22 3:00 AM	1/18/22 3:00 AM	_scheduler

Double-click an entry to view further details:

Job Log	
<pre>Job name: cara_folders_cleanup Method name: cara_folders_cleanup User name: Barry Prince Status: Done Queue time: 2022-01-14T11:27:17.167Z Start time: 2022-01-14T11:27:17.272Z ----- Executing folder cleanup... tmf_studentnb - removed 2 folder with IDs: [4, 6] tmf_studentnb - removed 1 folder with IDs: [3] tmf_studentnb - removed 1 folder with IDs: [2] Folder cleanup finished.</pre>	

End time:	2022-01-14T11:27:17.688Z

CARA System

Jobs

Described

CARA File Content Store Integrity.

Checks that each document content file listed in the index is present within the defined storage device. This can be useful if the content of a storage device needs to be restored from a backup. The report will list any missing files. For file system based storage devices the report can also include checksum based verification of the content files.

Parameters:

connectors. Specifies which connectors will be checked. Enter the connector's name, comma separated where there is more than one.

For example _file_system,_amazon_s3.

checksum. A Boolean value which determines if a file integrity checked is applied. Set to **true** or **false**. This option is only applicable to file system based storage devices.

CARA Folders Cleanup.

Removes folders which are empty. For object types which use automatic folder linking, folders are created automatically as needed, based on their type linking rules. Folders may be left empty if all documents within are automatically moved due to a change to the linking rules. This job checks for and removes empty folders.

This job has no parameters.

CARA LDAP Sync.

Synchronizes users and groups from a LDAP connection. LDAP connection and synchronisation details are specified in the `ldap-authentication.yml` file.

Parameters:

isFullSync. Determines if a complete set of user and group IDs are queried from the user registries in order to determine deletions. This parameter is independent of the `forceUpdate` parameter as a separate query is run.

forceUpdate. Determines if the complete set of users and groups are created and updated locally, or only those changed since the last synchronisation. When set to **true** all users and groups are queried from the user registry and updated locally. When set to **false** each source is only queried for those modified since the most recent modification date of all the objects last queried from that same source.

allowDeletions. Determines if deletions are allowed of CARA users who have been marked for deletion on the LDAP source.

CARA Logs Cleanup.

Cleans logs based on a given threshold.

Applies to cara_job_log, cara_method_log and cara_queue_item.

The parameter **threshold** is the age in months at which a log is cleaned.

Parameter:

action. Set to either **delete** (default) or **truncate**.

CARA Object Owner Change.

Changes ownership of documents.

Parameters:

current_owner. Username of the current owner. Case-sensitive.

new_owner. Username of the new owner. Case-sensitive.

CARA Purge.

Purges objects based on the purge configuration of each type. See the Purge Configuration section of this manual for details.

Parameter:

test. If the **test** parameter is **true**, no objects are purged but the job log records which objects were eligible for purging.

CARA Reindex Objects On Dictionary Update.

Updates object values for attributes which have been selected from a dictionary, when the source dictionary alias has been modified.

This job has no parameters.

CARA Remove Orphaned Files.

When documents are moved, the files are still referenced in the cara_file index and the content files remain in the associated storage location (such as filesystem or S3). This job removes both when run.

Parameter:

test. This job runs in test mode by default, producing a report of the files which are eligible to be removed. Change the **test** parameter to **false** to have the job remove the orphaned content files.

CARA Security and Linking Update

Recalculates security and folder linking rules on objects identified with a CQL query. Ordinarily security and linking rules are reapplied to objects when they are versioned or their properties are saved. This job allows updated security and linking rules to be applied to sets of objects, preventing a need to update objects individually in order to trigger the recalculation when a rule has changed.

Parameter:

cql. The job has a single parameter, the CQL query that identifies the objects for which security and linking rules are reapplied. The CQL query cannot be blank. It must be a full valid select statement. Select * can be used.

Example: select * from clinical_doc where lifecycle_state="Draft".

This would recalculate security and linking rules for draft objects of type clinical_doc.

CARA Send Date Based Notifications.

Checks for and sends active date based notifications. The associated job logs detail which notifications were triggered.

This job has no parameters.

CARA Sharepoint Cleanup.

Removes orphaned (locked) documents in Sharepoint. Documents imported into Sharepoint for collaboration cannot be automatically deleted from Sharepoint at the moment the collaboration ends as Sharepoint maintains a temporary lock on those documents.

This job has no parameters.

CARA Signing Request Sync.

Monitors AdobeSign and DocuSign services for completed signing requests and triggers the creation of related renditions within CARA.

This job has no parameters.

CARA System Statistics.

Logs cluster, index and type statistics hourly.

This job has no parameters.

CARA Trash Cleanup.

Deletes documents from trash.

Parameters:

threshold. The number of days a document must be in trash before it is deleted.

test. If the test parameter is **true**, the job does not delete any documents, but the job log records which documents are due to be deleted based on the threshold parameter. Set to false to have the job run fully.

CARA User Membership Sync.

Recalculates user and group membership when users are selected as workflow task recipients.

This job has no parameters.

CARA Workflows Overdue Notifications.

Sends out overdue workflow notifications. Each time the job runs it will send out notifications according to the **Target Duration** property of each workflow template. The channel used and the notification message sent are determined by each notification template individually.

Parameters:

dry_run. If the dry_run parameter is **true**, no notifications are sent and the job log records each overdue workflow task including the user and the number of days overdue. If set to **false** notifications are sent.

remind_on_day. Determine the days after tasks are due that notifications are sent. Entered as a comma separated list of integers. For example 1,7,14 would sent out overdue notifications on the 1st, 7th and 14th days after a task is overdue.

3.4. Methods

Description

A method is a Java class or script which performs a requested function, for example a method is used to synchronise user details via LDAP. New methods can be written to extend CARA functionality. Methods can be run manually as needed, or scheduled to run automatically via a job.

Name ↑	Display Name	Type	Status	Version
cara_deactivate_provisioned_users	Deactivate Provisioned Users	JavaClass	Done	9
cara_file_content_store_integrity	File Content Store Integrity Method	JavaClass		12
cara_folders_cleanup	Empty Folders Cleanup Method	JavaClass	Done	12
cara_ldap_sync	LDAP Synchronization Method	JavaClass		12
cara_logs_cleanup	Logs Cleanup Method	JavaClass		12
cara_object_owner_change	Change Object Owner	JavaClass		9
cara_purge	Document Purge Method	JavaClass	Done	12
cara_reindex_objects_on_dict_update	Dictionary Update Reindexing Method	JavaClass		12
cara_remove_orphaned_files	Remove Orphaned Files Method	JavaClass	Done	12
cara_security_and_linking_update	Recalculate Security And Linking On ObjectsJob and	JavaClass	Done	9
cara_send_date_based_notifications	Send Date Based Notifications	JavaClass	Done	12
cara_sharepoint_cleanup	SharePoint Cleanup Method	JavaClass		12
cara_signing_request_sync	Sync Signing Requests	JavaClass		6
cara_system_statistics	System Statistics Method	JavaClass	Done	12

Adding a Method

1. From the CARA Control Panel select **Administration > Methods**.
2. Select **+ Add** to add a new method, the **New Method** window opens, complete the following information:

The 'New Method' dialog box shows the following fields:

- ID:*
- Display Name:*
- Type: JavaClass
- Class Path:

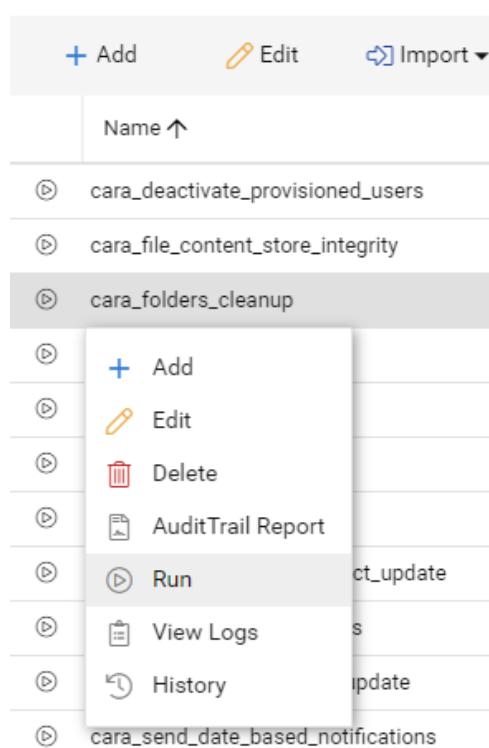
Buttons at the bottom:

- Save
- Cancel

- a. **ID.** The ID can only contain lowercase letters, numbers, underscores and hyphens. It must start with a letter.
- b. **Display Name.** The Display Name can contain mixed-case letters and spaces.
- c. **Type:** Select the method type, either **JavaClass** or **Script**.
- d. **Class Path.** Enter the full class path to the method.
- e. **Save.**

Running a Method

1. Right-click the method and select **Run**:



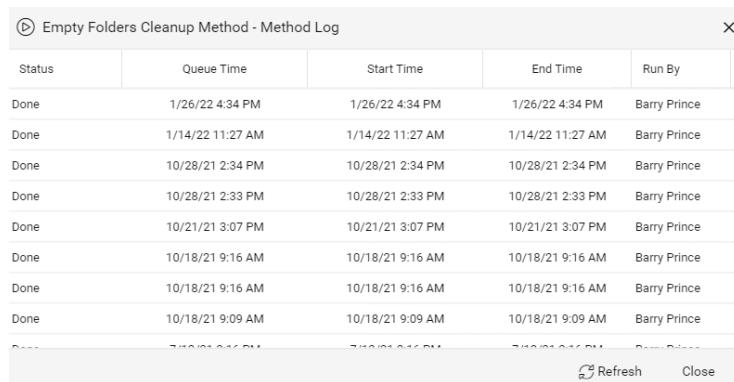
2. Click **Yes** to the confirmation window:
3. The **Info** window confirms that the method has been queued for execution and includes a link to the method logs window. The method logs window can also be accessed from the right-click menu within the Methods window.

Viewing Method Logs

4. **Close** the Info window.

A link to the **Method Log** window is provided when a method is run. It can also be accessed by right-clicking a method and selecting **View Logs**:

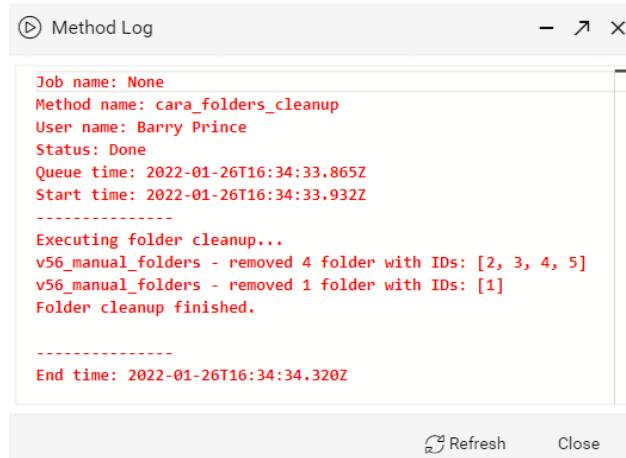
The Method Logs window lists the **Status**, when the method was queued, when the method was started, ended and the user who ran the method:



The screenshot shows a table titled "Empty Folders Cleanup Method - Method Log". The columns are labeled "Status", "Queue Time", "Start Time", "End Time", and "Run By". There are nine rows, each representing a completed task. The "Status" column shows "Done" for all entries. The "Run By" column consistently shows "Barry Prince". The "Start Time" and "End Time" columns show various dates and times from January 18, 2022, to January 26, 2022.

Status	Queue Time	Start Time	End Time	Run By
Done	1/26/22 4:34 PM	1/26/22 4:34 PM	1/26/22 4:34 PM	Barry Prince
Done	1/14/22 11:27 AM	1/14/22 11:27 AM	1/14/22 11:27 AM	Barry Prince
Done	10/28/21 2:34 PM	10/28/21 2:34 PM	10/28/21 2:34 PM	Barry Prince
Done	10/28/21 2:33 PM	10/28/21 2:33 PM	10/28/21 2:33 PM	Barry Prince
Done	10/21/21 3:07 PM	10/21/21 3:07 PM	10/21/21 3:07 PM	Barry Prince
Done	10/18/21 9:16 AM	10/18/21 9:16 AM	10/18/21 9:16 AM	Barry Prince
Done	10/18/21 9:16 AM	10/18/21 9:16 AM	10/18/21 9:16 AM	Barry Prince
Done	10/18/21 9:09 AM	10/18/21 9:09 AM	10/18/21 9:09 AM	Barry Prince
Done	7/10/21 2:16 PM	7/10/21 2:16 PM	7/10/21 2:16 PM	Barry Prince

Double-click an entry to view further details:



The screenshot shows a window titled "Method Log" with a single entry. The log output is as follows:

```

Job name: None
Method name: cara_folders_cleanup
User name: Barry Prince
Status: Done
Queue time: 2022-01-26T16:34:33.865Z
Start time: 2022-01-26T16:34:33.932Z
-----
Executing folder cleanup...
v56_manual_folders - removed 4 folder with IDs: [2, 3, 4, 5]
v56_manual_folders - removed 1 folder with IDs: [1]
Folder cleanup finished.

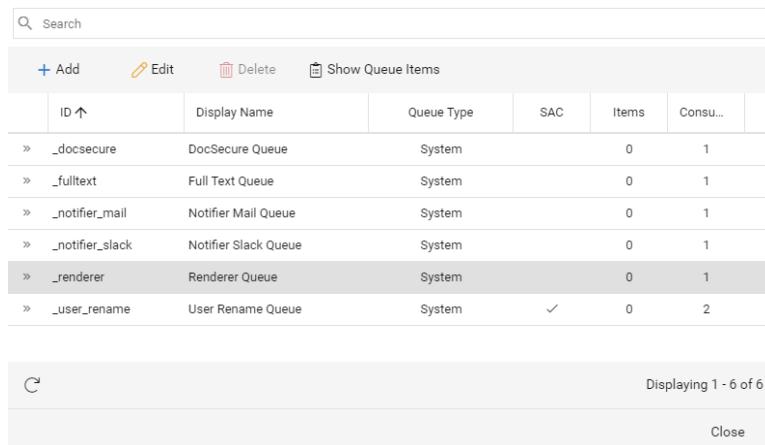
-----
End time: 2022-01-26T16:34:34.320Z

```

3.5. Queues

Description

Queues pass requests to services. Typically a request is generated by an action performed on an object and then placed in a queue. For example, a new document being created can result in a request for a new rendition being placed in the rendition queue, which is then processed by the rendition service. The services which access queues can either be internal to CARA or external via a 3rd party API using the CARA Rest Client.



ID ↑	Display Name	Queue Type	SAC	Items	Consumers
» _docsecure	DocSecure Queue	System	0	1	
» _fulltext	Full Text Queue	System	0	1	
» _notifier_mail	Notifier Mail Queue	System	0	1	
» _notifier_slack	Notifier Slack Queue	System	0	1	
» _renderer	Renderer Queue	System	0	1	
» _user_rename	User Rename Queue	System	✓	0	2

Displaying 1 - 6 of 6

Close

By default CARA includes the following system queues for internal services:

_docsecure. The DocSecure queue processes requests for additional layers on PDF files, such as watermarking, electronic signature pages and property overlays.

_fulltext. The full text queue processes requests for the indexing of document content files, enabling full text searching.

_notifier_mail. The mail notifier queue processes notification requests for the mail channel, typically related to document change notifications and workflow tasks.

_notifier_slack. The slack notifier queue processes notification requests for the slack channel, typically related to document change notifications and workflow tasks.

_renderer. Renderer queue. The renderer queue processes requests for the creation of document renditions, commonly in a PDF format and triggered by documents being created or updated.

_user_rename. The user rename queue processes requests for users to be renamed. Unlike other queues, the user rename queue is set to single active consumer by default and this setting cannot be changed.

Other queues, such as for external services, can be created as required.

The requests which are placed into the queues and their trigger conditions are configured separately per type. Please see the Type Queues chapter for details.

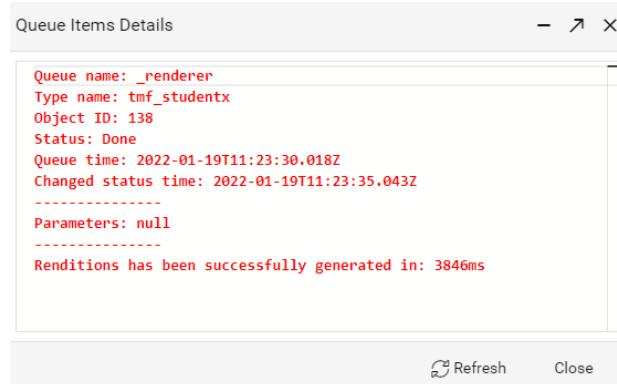
Queue items for a highlighted queue can be viewed by selecting **Show Queue Items** from the right-click menu:

Status	Start Time ↑	Status Change Time	Related Object	User
Done	1/19/22 11:24 AM	1/19/22 11:25 AM	tmf_studentx:141	Barry Prince
Done	1/19/22 11:24 AM	1/19/22 11:24 AM	tmf_studentx:140	Barry Prince
Done	1/19/22 11:24 AM	1/19/22 11:24 AM	tmf_studentx:139	Barry Prince
Done	1/19/22 11:23 AM	1/19/22 11:23 AM	tmf_studentx:138	Barry Prince
Done	1/18/22 3:11 PM	1/18/22 3:11 PM	type_v56:5	Barry Prince
Done	1/18/22 1:27 PM	1/18/22 1:27 PM	type_v56:4	Barry Prince
Done	1/18/22 11:16 AM	1/18/22 11:16 AM	tmf_studentx:137	Barry Prince
Done	1/14/22 2:31 PM	1/14/22 2:31 PM	tmf_studentx:136	Barry Prince
Done	1/14/22 11:35 AM	1/14/22 11:35 AM	tmf_studentx:135	Barry Prince
Done	1/14/22 11:34 AM	1/14/22 11:34 AM	tmf_studentx:134	Barry Prince
Done	1/4/22 4:39 PM	1/4/22 4:39 PM	tmf_studentnb:13	naomi
Done	1/4/22 4:39 PM	1/4/22 4:39 PM	tmf_studentnb:17	naomi

Displaying 1 - 25 of 193

Close

Details of individual queue requests are available by highlighting an item and selecting **Details**:



Selecting **Locate** will search for and highlight the related document in the main view panel.

The `_notifier_mail` and `_notifier_slack` queues have the additional option to **Re-Queue Items** individually, available from the toolbar and right-click menu:

<code>_notifier_mail - Queue Items</code>				
Status	Start Time ↑	Status Change Time	Related Object	User
Done	1/25/22 4:35 PM	1/25/22 4:35 PM	<code>_workflow:studentx_workflow_21</code>	Barry Prince
Done	1/24/22 9:48 AM	1/24/22 9:48 AM	<code>_workflow:studentx_workflow_21</code>	Barry Prince
Done	1/21/22 10:42 AM	1/21/22 10:42 AM	<code>_workflow:studentx_workflow_20</code>	Barry Prince
[Details] [Locate] [Re-Queue Items]		1/21/22 10:41 AM	<code>_workflow:studentx_workflow_15</code>	Barry Prince
[Details] [Locate] [Re-Queue Items]		1/21/22 10:16 AM	<code>_workflow:studentx_workflow_19</code>	John Smith
[Details] [Locate] [Re-Queue Items]		1/21/22 10:16 AM	<code>_workflow:studentx_workflow_18</code>	John Smith
Done	1/21/22 9:40 AM	1/21/22 9:40 AM	<code>_workflow:studentx_workflow_17</code>	Barry Prince

Creating Queues

The internal CARA system queues described above are present by default and so do not need to be created manually. Other queues, for 3rd party services or internal processing by script, can be created:

1. Within the CARA Control Panel select **Queues** from the **General** panel.
2. Click **+ Add**, the **New Queue** window opens:

The screenshot shows the 'New Queue' configuration window. It includes fields for 'ID:' (with a placeholder box), 'Display Name:' (with a dropdown menu and edit icon), 'Options:' (checkbox for 'Single Active Consumer' with a tooltip), and 'Groups Allowed to Access the Queue:' (dropdown menu with edit icon). At the bottom are 'Save' and 'Cancel' buttons.

3. Enter a unique **ID** for the queue. The ID can only contain letters, numbers and underscores and must start with a letter.
4. Enter a **Display Name** for the queue. The display name can be entered manually or chosen from the drop-down list linked to the `_display_labels` dictionary.
5. **Single Active Consumer**. Select if the queue should allow only a single consumer at a time to be processing requests. The `_user_rename` queue requires this setting to be active, for other queues it is optional.
6. Select **Groups Allowed to Access the Queue**. These are the groups allowed to connect to the queue from the REST API in order to process requests. Click the drop-down arrow to open the **User and Groups Selection** window:

Select users and groups as appropriate and **Save**, returning to the **New Queue** window.
7. **Save** the new queue configuration.

Deleting Queues

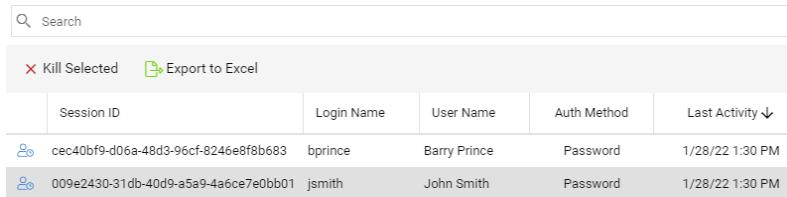
Non-system queues can be deleted if the service is no longer required. Highlight the queue in the **Queues** panel and select **Delete**. Click **Yes** to the subsequent confirmation window.

3.6. Sessions

Description

Active user sessions are listed in the **Sessions** window.

The Session ID, Login Name, User Name, Authentication Method, Last Activity, Login Date and IP Address are listed.



Sessions				
Session ID		Login Name	User Name	Auth Method
Last Activity				
	cec40bf9-d06a-48d3-96cf-8246e8f8b683	bprince	Barry Prince	Password
	009e2430-31db-40d9-a5a9-4a6ce7e0bb01	jsmith	John Smith	Password

Sessions can be killed by highlighting them and clicking **Kill Session**. Click **Yes** on the Confirmation window:

The current list of sessions can be exported by clicking **Export to Excel**.

3.7. Spaces and Multi-Tenancy

Description	CARA can be deployed in two modes: single instance and multi-tenancy.
	Single instance is the most common and allows the use of CARA by a single organisation or customer.
	Multi-tenancy enables the creation of multiple tenants, called spaces. Every space within a multi-tenancy environment has its own domain and visibility only of its own data. Spaces within a multi-tenancy environment share the underlying Elasticsearch platform.
	Where a multi-tenant environment is required, it is enabled at installation time. In a multi-tenant installation there is always a root space which is used to manage other tenants/spaces. The root space should not be used to store data.
	A dedicated section of the CARA Control Panel is used to create and update spaces. The Spaces section of the Control panel is not present in single instance environments.
	Multi-tenancy introduces a new access level that allows precise control over the system: SuperAdmin. The following table illustrates the different levels of access for Admin and SuperAdmin in a single instance vs. multi-tenant environment.

Section	Single Instance	Multi-Tenant Root	Tenant
Spaces Management	-	SA	-
Sessions	SA	SA	-
Logs	SA	SA	-
Clear Cache	SA	SA	-
Users	A	A	A
Groups	A	A	A

System Group Update	SA	SA	SA
Auth Tokens	SA	SA	SA
Jobs	A	A	A
Methods	A	A	A
Types	SA	SA	SA
Queue Logs	SA	SA	SA
Queues Management	SA	SA	-
Analytics	Analytics Manager	Analytics Manager	Analytics Manager
Query Executor	Query Executor	Query Executor	Query Executor
Script Executor	Script Executor	Script Executor	Script Executor

Key:

SA - SuperAdmin

A - Admin

" " Not Available

Multi-tenancy Considerations:

Super Admin access. Customers in spaces are not given Super Admin access, which would allow them to create other SuperAdmins and scripts. Customers in spaces cannot create and update scripts by themselves.

SSO/SAML. It is currently not possible to use SSO/SAML authentication in a Mult-tenant environment.

Types. Only SuperAdmins can create types. Every type uses an index and has a direct impact on the cluster size.

Sessions, Logs and Clear Cache. System tools such as Sessions, Logs and Clear Cache are only available on the root space for personnel with SuperAdmin access.

Creating Spaces

To create a space in a CARA multi-tenancy environment:

1. From the CARA Control Panel select **Spaces** within the **Administration** section.

Space Name ↑	Display Label	Host Name	Users	Last Login
mt1	Tenant 1	develop2-mt-1.generis.cloud	96	2/11/22 2:06 PM
mt2	Tenant 2	develop2-mt-2.generis.cloud	87	2/11/22 1:55 PM
mt3	Tenant 3	develop2-mt-3.generis.cloud	2	2/11/22 11:33 AM
new	new	new	1	
qa	QA	qa	1	
tasteTenant	atest	naasd	1	2/11/22 10:15 AM
test	testaa	test	1	2/4/22 10:31 AM

Displaying 1 - 7 of 7

2. In the Spaces panel click Add to open the New Spaces window.

General

Name:*

Display Label:

Host Name:

Status: Active

Space Logo
For best visualization, only SVG-format vector images are allowed with logo size up to 520x100 or a similar ratio

Light Mode:

Dark Mode:

Save **Cancel**

3. Enter a Name and Display Label for the space.
4. Enter the Host Name of the space.
5. Status. Select if the space will be Active. Spaces which are inactive cannot be accessed but retain their data.

6. Select a logo for the space. Only SVG format vector images are allowed. Separate logos can be selected for light and dark mode.

General

Name*: mt2

Display Label*: Tenant 2

Host Name*: develop2-mt-2.generis.cloud

Status: Active

Space Logo

For best visualization, only SVG-format vector images are allowed with logo size up to 520x100 or a similar ratio

Light Mode:  Browse...

Dark Mode:  Browse...

7. Save the space configuration.

3.8. Type Definition

Description

Types define the objects that users will work with, and that will be managed by CARA. Primarily users work with documents, that is objects which have content and can be versioned, however other object types can also be defined and managed.

Each organisation may have many different types of documents or objects. Departments within an organisation may define objects which are specific to them and which only their users will work with. Types are defined and managed from the Types Definition section of the control panel.

The Type Definition section lists the currently defined types and includes their **Name** and current **Object Count**. Other properties, including the user who last updated the type definition, are also displayed.

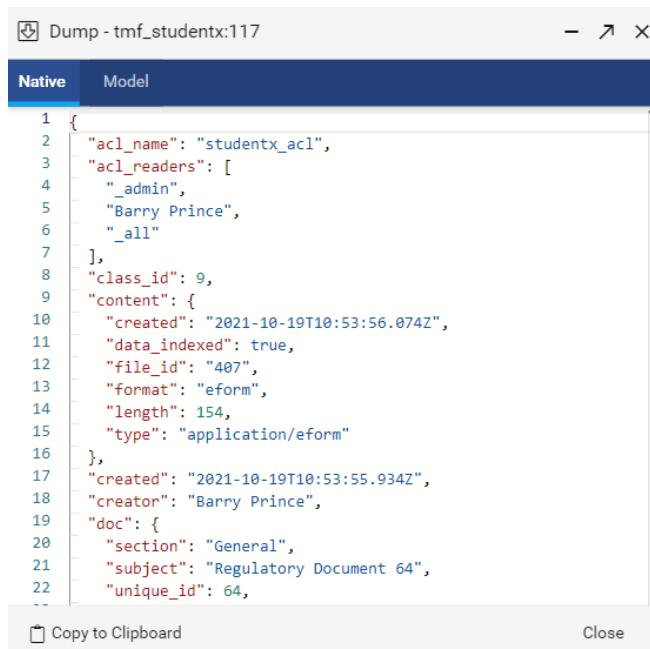
Name	Display Name	Object Count	Content?
data_type1 (v1)	Data Type1	16	
tmf_studentx (v1)	TMF Studentx	109	✓
type_v56 (v1)	Type V56	4	✓

CARA version 5 uses Elasticsearch as the primary platform, which is a non-relational database, it does not use a hierarchy and inheritance based object model. However there are a core set of properties available to all types by default, such as `object_name` and `version_label`.

Custom attributes are defined for each object type. It is common for multiple object types to have attributes with the same name, for example 2 or more types can have an attribute named `country` or

region. Custom attributes are listed with the prefix doc. for example a custom attribute named product becomes doc.product.

Each document (and each version of the document) has a corresponding JSON file which contains core properties, custom properties and content text. This JSON file can be viewed directly from the CARA user interface by selecting Dump. When viewed as a dump the content text, which can be very large, is omitted.



The screenshot shows a 'Dump' window titled 'Dump - tmf_studentx:117'. The window has tabs for 'Native' and 'Model', with 'Native' selected. The content area displays a JSON object with numbered lines 1 through 22. Lines 1-16 show standard document properties like acl_name, acl_readers, class_id, content, created, data_indexed, file_id, format, length, and type. Lines 17-22 show custom properties under the doc section, including section, subject, and unique_id. At the bottom of the window are 'Copy to Clipboard' and 'Close' buttons.

```

1  {
2    "acl_name": "studentx_acl",
3    "acl_readers": [
4      "_admin",
5      "Barry Prince",
6      "_all"
7    ],
8    "class_id": 9,
9    "content": {
10      "created": "2021-10-19T10:53:56.074Z",
11      "data_indexed": true,
12      "file_id": "407",
13      "format": "eform",
14      "length": 154,
15      "type": "application/eform"
16    },
17    "created": "2021-10-19T10:53:55.934Z",
18    "creator": "Barry Prince",
19    "doc": {
20      "section": "General",
21      "subject": "Regulatory Document 64",
22      "unique_id": 64,
...

```

Documents and objects are uniquely identified by a combination of their type and ID. The ID attribute is a per type sequential identifier. In the example above the document would be uniquely identified as tmf_studentx (the type name): 117 (the ID), as shown in the **Dump** window title bar.

In the example above the custom properties for the type tmf_studentx are grouped together under the “**doc**” section.

Core Object

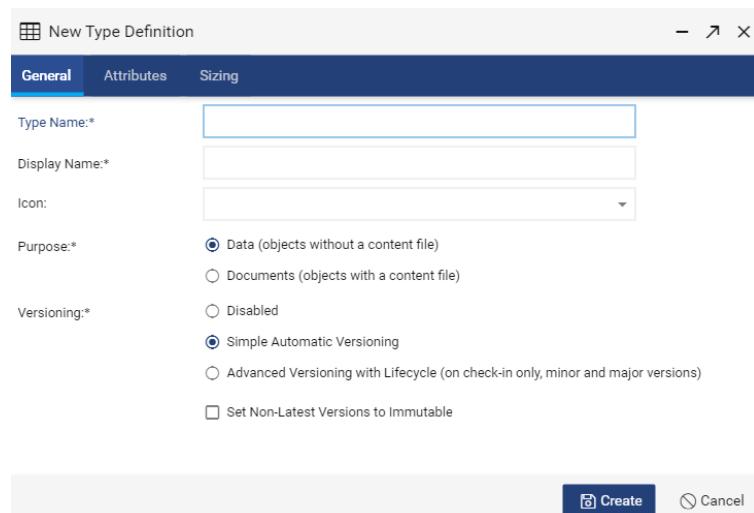
Properties

- object_id
- object_name
- owner
- created

- creator
- modified
- modifier
- lock_owner
- lock_date
- in_transaction
- is_template
- template_id
- lifecycle_name
- lifecycle_state
- acl_name
- acl_readers
- root_version_id
- previous_version_id
- version_no
- version_label
- is_latest
- status
- deleted
- content
- type
- format
- length
- checksum
- data
- data_indexed
- renditions
- type
- format
- length
- checksum
- identifier
- title

Creating Type Definitions – General Tab

1. From the CARA Control Panel select **Type Definition** within the **Administration** section, the Type Definition window opens.
2. Click **+ Add**, the **New Type Definition** window opens:



3. In the **General** tab enter a unique **Type Name**. Type names can only contain letters, numbers and underscores. They must start with a letter.
4. **Display Name**. Enter a display name for the type. End users will not typically see the display name, it is used within administration windows.
5. **Icon**. Select an icon by either scrolling through the drop-down list or entering part of an icon's name.
6. **Purpose**. Select the purpose as either **Data** or **Documents**. Data objects contain only metadata, they do not have associated content files.
7. If **Documents** is selected as the **Purpose**, the following additional options are displayed:

New Type Definition

General Attributes Sizing

Type Name:*

Display Name:*

Icon:

Purpose:*

Data (objects without a content file)
 Documents (objects with a content file)

Content Storage:*

CARA FileSystem Connector (_file_system)

Content Processing:

Indexed for Search Render PDF automatically

Versioning:*

Simple Automatic Versioning
 Advanced Versioning with Lifecycle (on check-in only, minor and major versions)
 Set Non-Latest Versions to Immutable

Folders:*

Disabled
 Automatic Folder Linking Based on Rules
 Manual Folder Linking

Create **Cancel**

- a. **Content Storage.** Select the connector to be used for the storage of content files. CARA supports multiple content storage methods such as cloud, local file system and 3rd party applications such as Box. Each method to be used is defined as a separate connector in the Connectors section of the control panel.
- b. **Content Processing: Indexed for Search.** Add text contained within the document's content file to the searchable full text index.
- c. **Content Processing: Render PDF Automatically.** When documents of the type are created or versioned, automatically render a PDF rendition.
- d. **Folders.** Place documents in a folder hierarchy. Documents can be automatically placed in a folder location based on the type's linking rules or they can be placed in a folder selected by a user manually.

When **Automatic Folder Linking Based on Rules** is enabled, folders are created automatically when required, based on the type-specific linking rules. When the folder

location is set to **Manual** folders will need to be created manually by users.

When **Manual Folder Linking** is enabled, folders need to be created and managed manually. There are separate type capabilities that can be granted to users allowing them to create root folders, create folders, rename folders, remove folders, move folders and delete folders:

Type Capabilities	
Capability	Default
Create Folder	<input checked="" type="checkbox"/>
Create Root Folder	<input checked="" type="checkbox"/>
Delete Folder	<input checked="" type="checkbox"/>
Folder Navigation	<input checked="" type="checkbox"/>
Folder Restrictions	<input checked="" type="checkbox"/>
Move Folder	<input checked="" type="checkbox"/>
Rename Folder	<input checked="" type="checkbox"/>

[+ Add Role](#)
 [Remove Role ▾](#)
 [Save](#)

When **Manual Folder Linking** is enabled, users with write permissions can drag and drop documents between folders.

The creation and use of folders can also be set to **Disabled**.

8. **Versioning.** For both **Data** and **Document** object types the following versioning options are available:
 - a. **Simple Automatic Versioning.** Only integer (major) version numbers are applied. The version is incremented only upon check-in. The version label attribute only uses the LATEST label, in order to identify the latest version. Other descriptive labels such as Draft or Approved are not used.
 - b. **Advanced Versioning with Version Labels.** Advanced versioning includes minor version increments such as 0.1, 0.2,

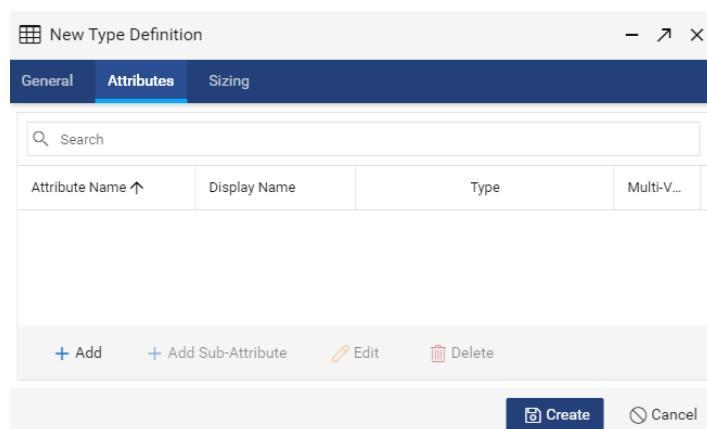
0.3 and allows version labels to include descriptive text such as Draft or Approved.

- c. **Set Non-Latest Versions to Immutable.** All versions other than the latest are set to immutable, they cannot be versioned, deleted or have their properties changed. They can have annotations added to a rendition.

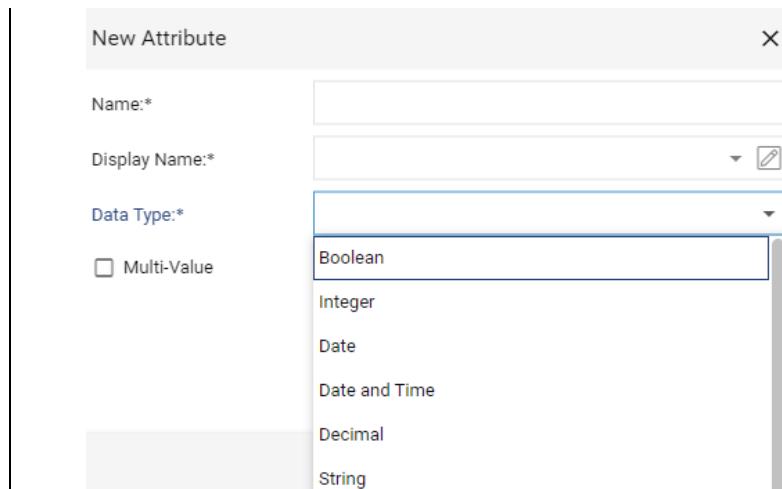
Please note that the options above cannot be changed after the initial type definition has been saved. Type attributes can be added at any time, as detailed in the following section. Type attributes cannot be removed if they are in use.

Creating Type Definitions - Attributes Tab

1. To add custom attributes to a type click **+ Add** in the **Attributes** tab.



2. **Name.** Enter the attribute's name, which can contain only lowercase letters, numbers and underscores and must start with a letter.
3. **Display Name.** Enter a display name manually or select a value from the _display_labels dictionary.
4. **Data Type.** Select the data type:



The available data types are:

- a. **Boolean.**
- b. **Integer.**
- c. **Date.**
- d. **Date & Time.**
- e. **Decimal.**
- f. **String.**
- g. **DictionaryKey.**
- h. **User/Group Reference.** Allows selecting an existing user or group.
- i. **GeoPoint.** Specifies a geographical location in latitude and longitude with a comma separator, for example 51.483, -0.604. Required for map filter panels.
- j. **ID (Object Reference).** ID objects are used to display the properties of other objects, such as their name. For example

an email and its attachments could be imported into CARA as separate documents, but reference each other so that users can easily see they are related. The properties form can display multiple properties of the referenced documents/objects.

ID (Object Reference) attributes must specify the type of object they can reference, which will usually be its own type.

Object *List* form controls are used to display multi-value ID (Object Reference) attributes on a properties form. See the forms chapter for details.

Object *Field* form controls are used to display single-value ID (Object Reference) attributes on a properties form. See the forms chapter for details

New Attribute	
Name:*	attachment X
Display Name:*	Attachment X <input type="button" value="edit"/>
Data Type:*	ID (Object Reference) X <input type="button" value="edit"/>
<input type="checkbox"/> Multi-Value	
Referenced Type:*	TMF Studentx (tmf_studentx) X <input type="button" value="edit"/>
Binding:*	<input type="radio"/> Selected Version <input checked="" type="radio"/> Latest Version
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

Binding refers to whether the referenced object will be the version that was originally **selected** by the user or the **latest** version where subsequent versions have been created.

- k. **Object.** Object attributes can have related sub-attributes. For example attribute Product might have sub-attributes for Product Name, Price and Code. When highlighting a nested

object attribute in the **New Type Definition** window + **Add Sub-Attribute** becomes available:

Attribute Name ↑	Display Name	Type	Multi-V...
product	Product	Object	✓
product_name	Product Name	String	
product_code	Product Code	String	
product_price	Product Price	Decimal	

Define a Name, Description and Data Type for Sub-attributes as for other attributes.

5. By default new attributes are single value. If required select **Multi - Value**. Single value attributes have only a single attribute value per object instance. Multi value attributes can have multiple values for a single attribute. For example, a book might have only 1 title (a single value attribute) but multiple authors (a multi-value attribute).
6. **Save** the new attribute definition.

Creating Type Definitions – Sizing Tab

Type sizing and the distribution of object data across the Elasticsearch platform is based on an estimated document (object) count and the underlying Elasticsearch cluster size.

When a cluster has just one node, all types will have just one primary shard and no replicas.

When a cluster has multiple nodes, types with an estimated document (object) count of greater than 100,000 will have as many shards as nodes and one replica. Types with an estimated document

(object) count of less than 100,000 will have one primary shard and one replica.

Replication ensures redundancy, prevents data loss and helps boost search performance. Replication should always be enabled in production environments and is enabled by default, however it can be disabled during migration to improve import speeds

New Type Definition

General Attributes Sizing

Estimated Document Count*: 50000

Data Replication: Disable Replication of Data
Replication ensures redundancy, prevents data loss and helps boost search performance. Replication should always be enabled in production environments and is enabled by default, however it can be disabled during migration to improve import speeds.

Create Cancel

1. Enter the **Estimated Document Count**.
2. Optionally **Disable Replication**.
3. Click **Create** to confirm creation of the new type.

Type Definition - Indexes Tab

Elasticsearch automatically creates an index for a type when it is first defined and saved. The index arranges the distribution of data for the type across available nodes and shards within a named cluster.

Because the index is only created once the type has been initially defined and created, the **Indexes** tab is not available until after initially saving and then returning to the type definition.

Details of the index, including the number of objects, size, number of (primary) shards and replicas, can be seen in the Indexes tab when selecting to Edit an existing type:

type_v56 - Edit Type

General Attributes Sizing **Indexes**

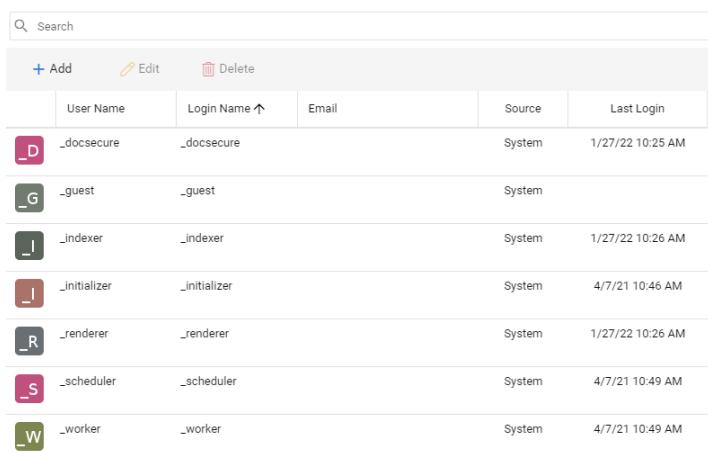
Index Name	Index Status	Object Count	Size	Shards
cara_data_type_v56-1	Active	5	260.3 KB	1

Refresh Save Cancel

3.9. Users

Description

Users can be created and managed manually within the system or imported and updated automatically from an LDAP connection. General user properties such as login name and email address are set within the Administration panel. Separately users are granted System Capabilities and specific Type Capabilities which determine the functionality available to them, these are covered in later sections.

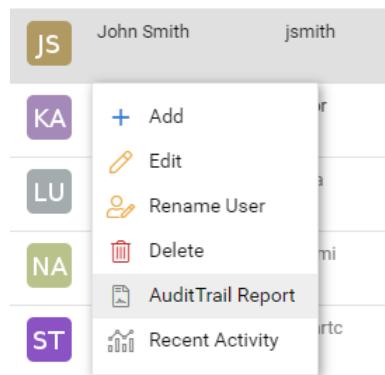


The screenshot shows a table with columns: User Name, Login Name, Email, Source, and Last Login. The data is as follows:

	User Name	Login Name	Email	Source	Last Login
	_D	_docsecure	_docsecure	System	1/27/22 10:25 AM
	_G	_guest	_guest	System	
	_I	_indexer	_indexer	System	1/27/22 10:26 AM
	_I	_initializer	_initializer	System	4/7/21 10:46 AM
	_R	_renderer	_renderer	System	1/27/22 10:26 AM
	_S	_scheduler	_scheduler	System	4/7/21 10:49 AM
	_W	_worker	_worker	System	4/7/21 10:49 AM

The **Users** window displays a list of users by **Name**, **Login Name**, **Email**, **Source** and **Last Login**. Users can be searched for in the search bar located at the top of the panel.

The **Audit Trail Report** and **Recent Activity** reports are available for a highlighted user from the right-click menu:



Please note that the **Users** window lists system generated users as well as organisation specific users. System users have a User Name and Login Name prefixed with an underscore and their Source = System.

Throughout CARA system generated configuration items, such as users, groups and dictionaries are indicated with an underscore prefix. Typically these items cannot be deleted and the Delete button is greyed out when they are selected.

Creating Users Manually

1. From the **CARA Control Panel** select **Administration > Users**, The **Users** window opens.
2. Select **+ Add**, the **New User** window opens. Fields marked with an asterisk are required. Complete the following fields:

The screenshot shows the 'New User' window with the 'General' tab selected. The window has tabs for General, Membership, Notifications, and Settings. The General tab contains fields for Status (Active, Locked), User Name, Login Name, First Name, Last Name, Display Name, Source (LDAP), LDAP ID, LDAP DN, and Email Address. There is also a placeholder for a user photo with the instruction 'Drop your photo here or browse your computer'. At the bottom are Save and Cancel buttons.

General Tab:

3. **Status.** Only users who are marked as **Active** can log into the system. Inactive users are listed in grey within the Users window, are unable to login and are not sent notifications.
Users who have exceeded the maximum number of password attempts are automatically **Locked** out. Locked users are listed with a strikethrough in the Users window.
4. **User Name.** The display name used throughout the system.
5. **Login Name.** The name the user enters when logging in, this does not need to be the same as the User Name. Only lowercase letters, numbers, underscores, hyphens, dots and @ are allowed. If @ is used it must be within a valid email address format. Login name must start with a letter.
6. **First Name, Last Name** and **Display Name** are optional.
7. **Source.** Refers to the user's password type:
 - **Inline.** Manually entered and stored in the system in an encrypted format. High password complexity is enforced when setting the password.
 - **LDAP.** Automatically synchronised from LDAP and not stored within the system.
 - **System.** No manual login, used for services connecting with authentication tokens.
 - **OKTA.** When using SAML based single sign-on.
8. **Password** and **Repeat Password.** These fields are only displayed when the user's password Source is set to **Inline**.
Passwords for inline users must be a minimum of 8 characters and include a mix of uppercase, lowercase and numbers. They must be at least medium complexity.
9. **LDAP ID.** LDAP user identification, only present when Source is set to **LDAP**.

10. **LDAP DN.** LDAP domain name, only present when Source is set to LDAP.
11. **Email Address.** Email addresses are used for notifications such as tasks and documents being updated. A valid email address format must be entered.
12. **Photo/Image.** Users can have an image displayed beside their username within the system. Either drag and drop a graphic or open the file upload window by clicking within the circle.

The following information is also displayed for existing users:

Last Login. Date.

Failed Login Attempts. Number of.

Membership Tab:

13. **Groups.** A list of the groups the user belongs to. Group membership determines user access level for documents and the functions available within the system.

Notifications Tab:

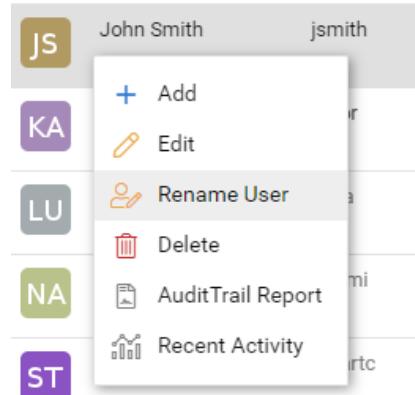
14. **Preferred Channel.** Set the channel used for notifications including those related to tasks and subscriptions. Users can also set and update their preferred channel.
15. **Locale.** Language locale, used to display appropriate language labels within the system.

Settings Tab:

16. The settings tab displays 5 custom user attributes. These attributes are read-only within the user interface, they can only be set by a script.

Renaming Users

User Name and Login Name can be modified after the user has been initially saved. Highlight a user in the **Users** window and select **Rename User** from the right click menu:



Enter the New User Name and New Login Name and click Rename:

Rename User	
User Name:	John Smith
Login Name:	jsmith
New User Name:*	<input type="text"/>
New Login Name:*	jsmith <input type="button" value="X"/>
<input type="button" value="Rename"/> <input type="button" value="Cancel"/>	

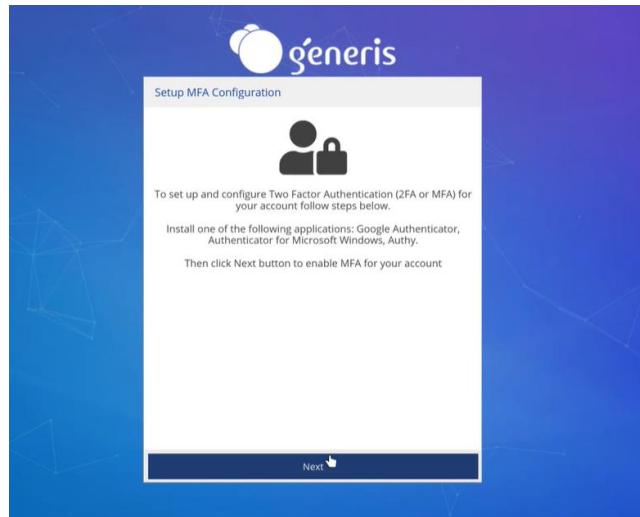
Renaming of the user is recorded in the audit trail, with the old and new name values listed.

Multi-Factor Authentication (MFA)

CARA supports multi-factor authentication (MFA) for both user login and user electronic signatures for audited events. Both features can be enabled independently of the other. If both are enabled, a user only needs to setup MFA once and the same authentication device can then be used for both.

MFA is enabled on the CARA server in the **application-dev.yml** file. There are 3 applicable settings in the **authentication** section of the file:

mfa_enabled: true or false. When set to true MFA is enabled for login. When a user logs in for the first time they are guided through the setup of a device for MFA, during which they are provided with a unique setup code which they enter into the authentication application on their device. Each subsequent time they login they are asked to provide a One Time Code (OTC) generated by the authentication application on their device.



user_signature_method: password or otc. When set to otc, for audited events which require an electronic signature, users are asked to provide a One Time Code (OTC). The first time a user is asked to provide signoff for an audited event they are guided through the setup of a device for MFA, during which they are provided with a unique setup code which they enter into the authentication application on their device. Subsequently when signing-off an audited event they are asked to provide an OTC generated by their device. The audit trail report shows whether a traditional username and password or OTC was provided for signoff:

Audit Trail Report - 5113.docx						
Filters		Events				
Last Month		ID	Event	Version	User Name	Time Stamp
Countries	No matches found	34034	System Update	0.3	Karen Bryd	5/27/2021 2:44 PM
Events	No matches found	34035	Update	0.3	Karen Bryd	5/27/2021 2:44 PM
Modified Attribute	No matches found	33954	System Update	0.3	Karen Bryd	5/26/2021 11:19 AM
Types	No matches found	33955	Update	0.3	Karen Bryd	5/26/2021 11:19 AM
Users		33768	System Update	0.3	Karen Bryd	5/24/2021 3:46 PM
		33769	Update	0.3	Karen Bryd	5/24/2021 3:46 PM
		33775	System Update	0.3	Karen Bryd	5/24/2021 3:55 PM
		33776	Update	0.3	Karen Bryd	5/24/2021 3:55 PM
		33779	System Update	0.3	Karen Bryd	5/24/2021 2:51 PM
		33780	Update	0.3	Karen Bryd	5/24/2021 2:51 PM
		33785	System Update	0.3	Karen Bryd	5/24/2021 2:50 PM
		33786	Update	0.3	Karen Bryd	5/24/2021 2:50 PM
		33487	System: Add Rendition	0.3	.renderer	5/21/2021 3:50 PM

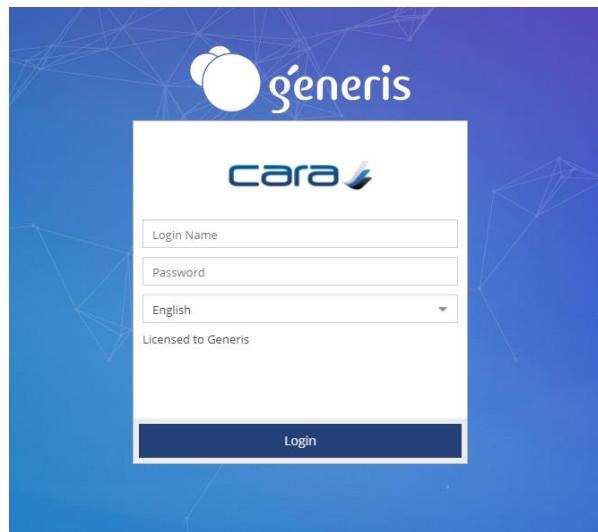
**User Setup of
Multi-Factor
Authentication
(MFA)**

exclusions: users entered here will not be required to provide OTC for either of the above enabled features.

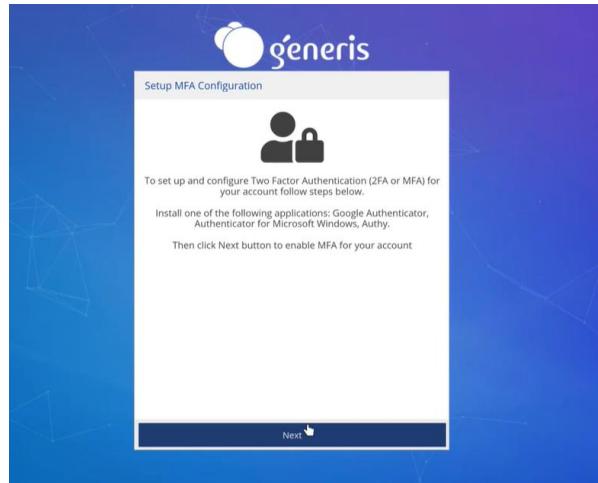
When MFA has been enabled for login, users initially login into CARA using their username and password and are then guided through the steps detailed below to setup MFA for CARA on their authentication device.

The MFA setup steps required for users when One Time Code (OTC) use is enabled for electronic signatures are the same as for MFA on login, except the setup is initiated when providing an electronic signature for the first time.

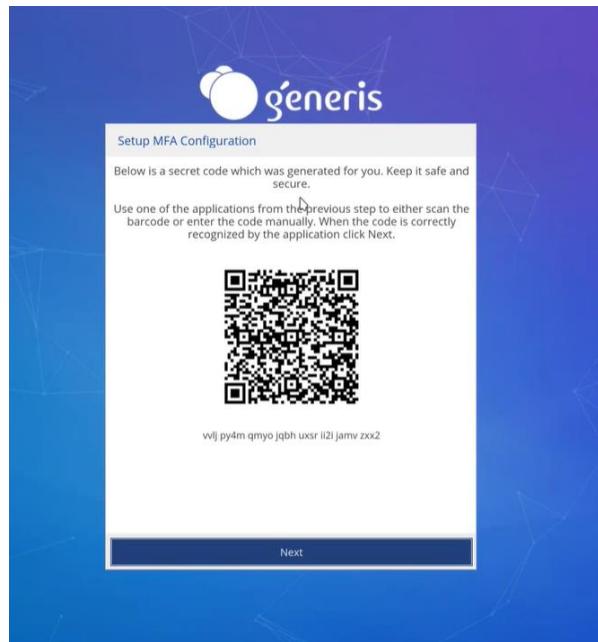
1. The user logs into CARA with their traditional username and password combination:



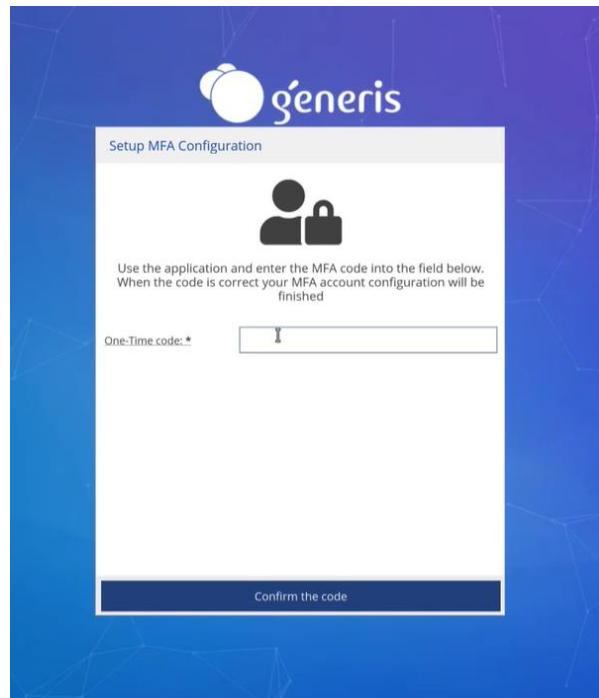
2. The Setup MFA Authentication window advises users of which authentication applications are supported:



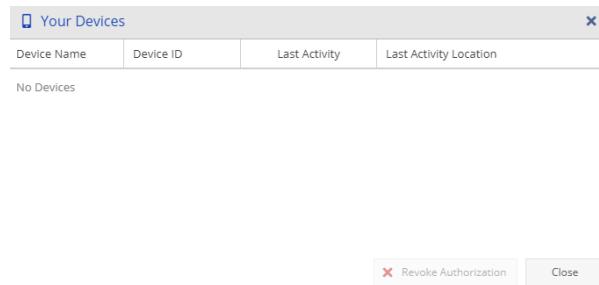
3. The user is provided with a unique setup code to be entered into the authentication application on their device. The code can be scanned as a QR code or manually entered as text:



4. The user is asked to enter the One Time Code (OTC) subsequently generated by their authentication application:



5. The user is logged in. From then on when the user logs in they are asked to provide the OTC generated by their device.
6. The option to Reset MFA Configuration is available under the User Profile menu in the top-right corner of the CARA user interface, which clears all user devices and again steps the user through the setup steps. Alternatively users can revoke the authorisation of individual devices from the Your Devices option under the User Profile menu:



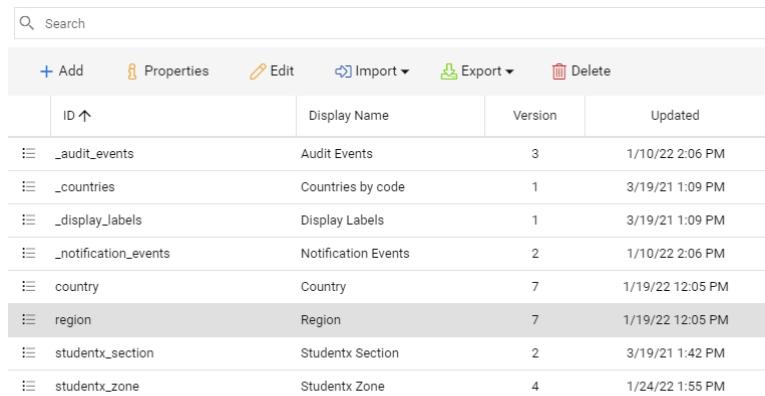
4. Data

4.1. Dictionaries

Description

A dictionary is a list of values, typically presented to users as a drop-down list on a property form. Dictionaries are intended to remove the need for users to enter data manually, which can result in mistakes or inconsistent data. Typical examples of dictionaries are for properties such as region or product.

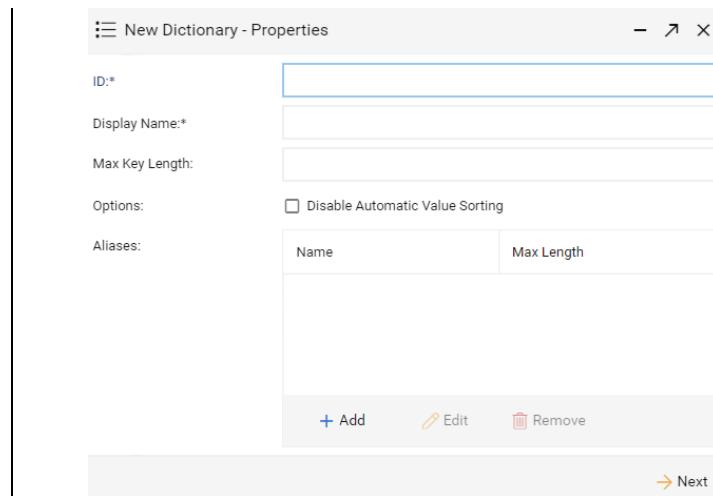
Dictionaries can be created and updated manually within CARA, or via 3rd party applications and then imported into CARA. Dictionaries can either be standalone or linked together within a taxonomy. Dictionaries are versioned each time they are updated.



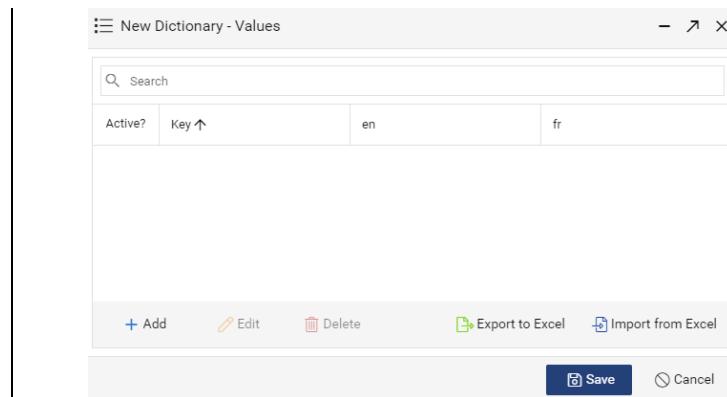
ID ↑	Display Name	Version	Updated
_audit_events	Audit Events	3	1/10/22 2:06 PM
_countries	Countries by code	1	3/19/21 1:09 PM
_display_labels	Display Labels	1	3/19/21 1:09 PM
_notification_events	Notification Events	2	1/10/22 2:06 PM
country	Country	7	1/19/22 12:05 PM
region	Region	7	1/19/22 12:05 PM
studentx_section	Studentx Section	2	3/19/21 1:42 PM
studentx_zone	Studentx Zone	4	1/24/22 1:55 PM

Creating a Dictionary

1. From the CARA Control Panel select **Data > Dictionaries**.
2. Select **Add** to create a new dictionary, the **New Dictionary** window opens:



3. Provide a unique **ID** (name) for the dictionary. ID's can only contain letters, numbers and underscores. They must start with a letter. System created dictionaries, which cannot be deleted, start with an underscore.
4. Provide a **Display Name** for the dictionary.
5. Specify the **Max Key Length**, which is the maximum length of a value.
6. When displayed to users, such as on a document properties form, the values within a dictionary are by default automatically sorted. Optionally **Disable Automatic Value Sorting**.
7. **Aliases** can be added to the dictionary. They are optional but are used to display user friendly labels for key values based on the user's language or other criteria. Click **Add** to create an alias, for each alias specify a **Name** and **Max Length**.
8. Click **Next**, the **New Dictionary - Values** window opens:



9. Click **+ Add**, the **New Value** window opens:

10. By default new values will be **Active**, that is displayed to users.

Values can be included within a dictionary but not be displayed to users by unticking **Active**. If an existing value should no longer be displayed to users, it is often preferable to make the value inactive rather than delete it.

11. Enter a **Key** value. If no aliases have been created, only a Key value is required for each entry. If aliases have been added, for example for different languages, enter an alias value for each Key value.

12. **Save** the New Value, returning to the **Values** window.

13. **Add** additional values as needed. The order of entries can be modified via click and drag:

Active?	Key	en	fr
<input checked="" type="checkbox"/>	uk	United Kingdom	Le United Kingdom
<input checked="" type="checkbox"/>	france	France	
<input checked="" type="checkbox"/>	germany	Germany	Deutschland
<input checked="" type="checkbox"/>	usa	USA	
<input checked="" type="checkbox"/>	canada	Canada	
<input checked="" type="checkbox"/>	japan	Japan	

[+ Add](#) [Edit](#) [Delete](#) [Export to Excel](#) [Import from Excel](#)

[Save](#) [Cancel](#)

14. Click **Save** once all values have been entered.

Updating Dictionaries

To modify a dictionary, for example to add a new Alias, in the **Dictionaries** window highlight the dictionary and select **Properties**:

ID:	studentx_zone						
Display Name:*	Studentx Zone						
Max Key Length:							
Options:	<input checked="" type="checkbox"/> Disable Automatic Value Sorting						
Aliases:	<table border="1"> <thead> <tr> <th>Name</th> <th>Max Length</th> </tr> </thead> <tbody> <tr> <td>en</td> <td>32</td> </tr> <tr> <td>fr</td> <td>32</td> </tr> </tbody> </table>	Name	Max Length	en	32	fr	32
Name	Max Length						
en	32						
fr	32						

[+ Add](#) [Edit](#) [Remove](#)

[Save](#) [Cancel](#)

To add, edit or delete the values within a dictionary, in the **Dictionaries** window highlight the dictionary and select **Edit**:

studentx_zone - Values - studentx_zone - Dictionaries			
Active?	Key	en	fr
<input checked="" type="checkbox"/>	Regulatory	Regulatory	
<input checked="" type="checkbox"/>	Site Management	Site Management	
<input checked="" type="checkbox"/>	Trial Management	Trial Management	

+ Add Edit Delete Export to Excel Import from Excel
 Save Cancel

Deleting Dictionaries

Non-system dictionaries can be deleted from the Dictionaries window by highlighting them and selecting **Delete**. The Delete button is greyed-out when system Dictionaries are selected as these cannot be deleted.

Dictionaries which are in use, such as within a taxonomy or properties form, cannot be deleted and a warning message is displayed when trying to delete them:



System Dictionaries

The following system dictionaries are created automatically:

- _audit_events
- _display_labels
- _notification_events
- _countries

The **_audit_events** dictionary lists events which can be recorded in the audit history. User friendly labels for audit events can be added and updated as preferred.

:_audit_events - Values - _audit_events - Dictionaries			
<input type="text"/> Search			
Active?	Key ↑	en	
<input checked="" type="checkbox"/>	add_relation	Add Relation	
<input checked="" type="checkbox"/>	add_rendition	Add Rendition	
<input checked="" type="checkbox"/>	cancel_checkout	Cancel check-out	
<input checked="" type="checkbox"/>	checkin	Check-in	
<input checked="" type="checkbox"/>	checkout	Check-out	
<input checked="" type="checkbox"/>	controlled_print	Controlled Print	
<input checked="" type="checkbox"/>	delete	Delete	
<input checked="" type="checkbox"/>	edit	Edit	

+ Add  Edit  Delete  Export to Excel  Import from Excel

 Save  Cancel

The **_display_labels** dictionary provides a list of values which can be selected when configuring user friendly labels in type and other view configuration. Entering and then selecting values from the **_display_labels** dictionary, instead of re-entering a label repeatedly across configuration windows, aids with consistency of labels.

:_display_labels - Values - _display_labels - Dictionaries			
<input type="text"/> Search			
Active?	Key ↑	en	

+ Add  Edit  Delete  Export to Excel  Import from Excel

 Save  Cancel

The **_notification_events** dictionary list the events that can trigger an automatic notification, such as when the task sent event triggers an email to the task recipients informing them of their task.

≡ _notification_events - Values - _notification_events - Dictionaries		
<input type="text"/> Search		
Active?	Key ↑	en
<input checked="" type="checkbox"/>	_create	Create
<input checked="" type="checkbox"/>	_job_completed	Job Completed
<input checked="" type="checkbox"/>	_state_change	State Change
<input checked="" type="checkbox"/>	_task_completed	Task Completed
<input checked="" type="checkbox"/>	_task_delegated	Task Delegated
<input checked="" type="checkbox"/>	_task_overdue	Task Overdue
<input checked="" type="checkbox"/>	_task_reassigned	Task Reassigned
<input checked="" type="checkbox"/>	_task_sent	Task Sent

The **_countries** dictionary provides a list of countries with the ISO English short names and corresponding alpha 2 codes.

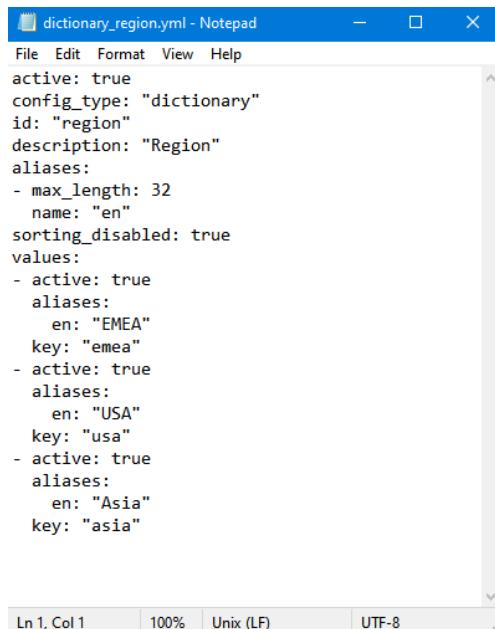
≡ _countries - Values - _countries - Dictionaries		
<input type="text"/> Search		
Active?	Key ↑	en
<input checked="" type="checkbox"/>	AD	Andorra
<input checked="" type="checkbox"/>	AE	United Arab Emirates
<input checked="" type="checkbox"/>	AF	Afghanistan
<input checked="" type="checkbox"/>	AG	Antigua and Barbuda
<input checked="" type="checkbox"/>	AI	Anguilla
<input checked="" type="checkbox"/>	AL	Albania
<input checked="" type="checkbox"/>	AM	Armenia
<input checked="" type="checkbox"/>	AO	Angola

Managing Dictionaries Externally

Dictionaries can be managed outside of CARA.

1. To export a dictionary, in the **Dictionaries** window highlight a dictionary and select **Export > Export to File**.
2. Select a location to export the .yml file to, or leave the default to save to your browser's default downloads location.

3. The .yml format file can be opened in a text editor such as Notepad or other 3rd party application:



A screenshot of a Windows Notepad window titled "dictionary_region.yml - Notepad". The window displays a YAML configuration file. The content includes fields like "active: true", "config_type: 'dictionary'", "id: 'region'", "description: 'Region'", and "aliases" which map "en" to "EMEA" and "USA" to "Asia". The Notepad interface shows standard controls at the top and bottom, including tabs for "Ln 1, Col 1", "100%", "Unix (LF)", and "UTF-8".

```
active: true
config_type: "dictionary"
id: "region"
description: "Region"
aliases:
- max_length: 32
  name: "en"
sorting_disabled: true
values:
- active: true
  aliases:
    en: "EMEA"
    key: "emea"
- active: true
  aliases:
    en: "USA"
    key: "usa"
- active: true
  aliases:
    en: "Asia"
    key: "asia"
```

4. Make changes as necessary and save.
5. To import a dictionary, with the correct dictionary highlighted in the CARA **Dictionaries** window select **Import From File**.
6. In the **File Upload** window locate and select the .yml format file to be imported and click **Open**.
7. The dictionary is updated and a new version created.

In addition to exporting and importing complete dictionaries in a .yml file format, there is a separate option to manage the values within a dictionary via export to and import from Excel:

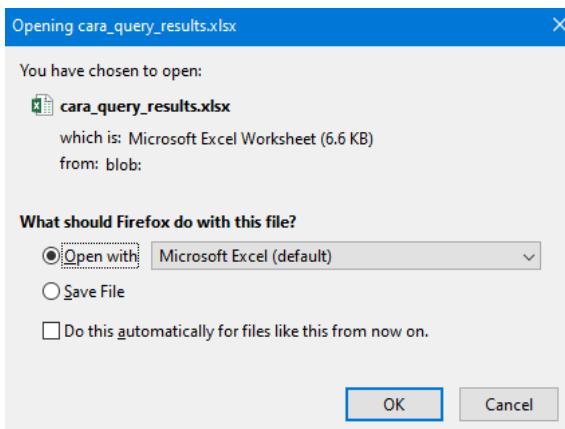
1. Within the **Values** window for a dictionary select **Export to Excel**:

The screenshot shows a software window titled '_countries - Values - _countries - Dictionaries'. At the top, there is a search bar with the placeholder 'Search' and a dropdown menu for 'Active?' and 'Key ↑' set to 'en'. Below this is a table listing countries:

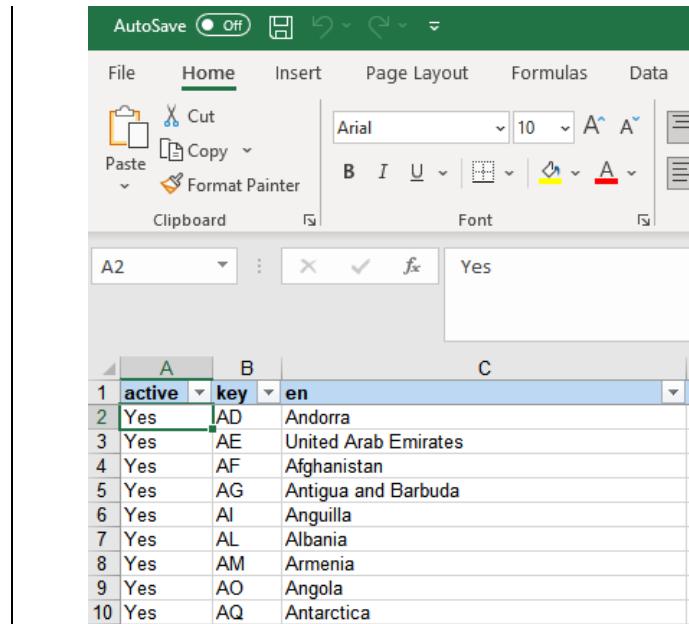
Active?	Key ↑	Value
<input checked="" type="checkbox"/>	AD	Andorra
<input checked="" type="checkbox"/>	AE	United Arab Emirates
<input checked="" type="checkbox"/>	AF	Afghanistan
<input checked="" type="checkbox"/>	AG	Antigua and Barbuda
<input checked="" type="checkbox"/>	AI	Anguilla
<input checked="" type="checkbox"/>	AL	Albania
<input checked="" type="checkbox"/>	AM	Armenia
<input checked="" type="checkbox"/>	AO	Angola

At the bottom of the window are buttons for '+ Add', 'Edit', 'Delete', 'Export to Excel' (highlighted in green), 'Import from Excel', 'Save' (highlighted in blue), and 'Cancel'.

2. Save the export file or open immediately as per your browser's instructions:



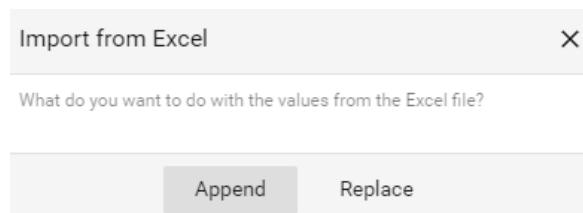
3. Add and update entries as appropriate and then save the exported file:



A screenshot of a Microsoft Excel spreadsheet. The table has three columns: A, B, and C. Column A contains the value 'active'. Column B contains country codes: AD, AE, AF, AG, AI, AL, AM, AO, and AQ. Column C contains country names: Andorra, United Arab Emirates, Afghanistan, Antigua and Barbuda, Anguilla, Albania, Armenia, Angola, and Antarctica. Row 1 is a header row with columns A, B, and C. Row 2 is the first data row.

A	B	C
1 active	en	
2 Yes	AD	Andorra
3 Yes	AE	United Arab Emirates
4 Yes	AF	Afghanistan
5 Yes	AG	Antigua and Barbuda
6 Yes	AI	Anguilla
7 Yes	AL	Albania
8 Yes	AM	Armenia
9 Yes	AO	Angola
10 Yes	AQ	Antarctica

4. To import an Excel spreadsheet containing dictionary values, within the **Values** window select **Import from Excel**. Locate the Excel format file and click **Open**.
5. The user is prompted to **Append** or **Replace** the existing values:



6. The updated list of values is displayed within the **Values** window, select **Save** to keep the imported changes.

4.2. Taxonomies

Description

A taxonomy is where two or more dictionaries are linked together, that is the choice a user makes in one dictionary determines the choices available in subsequent dictionaries.

Example: A dictionary for region and a dictionary for country could be linked so that when a user selects a region e.g. Europe they can then only choose countries from within that region e.g. UK, France or Germany.

Most dictionaries are standalone and do not need to be placed in a taxonomy, therefore taxonomies are optional, with one exception. A taxonomy needs to be created for the main classification choices presented to users when they are creating a new object or document.

Taxonomies can be up to 7 levels deep, that is they can connect up to 7 dictionaries in a hierarchy. One dictionary can be used within multiple taxonomies. Taxonomies are versioned each time they are updated.

ID	ID ↑	Display Name	Version	Updated
region_country	region_country	Region and Country	1	5/26/21 4:20 PM
studentx_main_classification	studentx_main_classification	Studentx Main Classification	2	3/19/21 1:43 PM

Creating a Taxonomy

1. Within the CARA Control Panel select **Taxonomies** from the **Data** section.
2. Select **+ Add**.
3. Enter a unique **ID** (name) for the taxonomy.
4. Enter a **Display Name**.

5. For each level of the taxonomy select the dictionary to be used.
Taxonomies can have up to 7 levels.

The screenshot shows a configuration interface for a taxonomy named 'region_country'. The 'Display Name' field contains 'Region and Country'. The 'Level 1' dropdown is set to 'Region (region)'. The 'Level 2' dropdown is set to 'Country (country)'. The 'Level 3' through 'Level 7' dropdowns are currently empty. At the bottom of the window are 'Save' and 'Cancel' buttons.

6. Click **Next**, the **New Taxonomy - Values** window opens:

The screenshot shows a 'New Taxonomy - Values' window. It includes a search bar, a table with columns for 'Node' and 'Active?', and a footer with buttons for adding items, deleting, exporting, importing, and saving.

7. Select **+ Add Item** to add items from the top level dictionary.
8. In the **Add Items** window either double-click entries to the included or highlight and use the arrows to move them across to the right-hand panel.
9. Click **Save** once complete.
10. Back in the **Values** window highlight an individual top level entry and click **Add Sub-item**.

The screenshot shows a list of taxonomy nodes under the heading "region_country - Values - region_country - Taxonomies". There is a search bar at the top. Below it is a table with columns for "Node ↑" and "Active?". Three items are listed: "asia" (active), "emea" (active), and "usa" (active). At the bottom are buttons for "+ Add Item", "+ Add Sub-item", "Delete", "Export to Excel", "Save", and "Cancel".

11. Select the entries from the sub-level dictionary to be available for that choice within the taxonomy.
12. Repeat until each level of the taxonomy has the appropriate sub-level dictionary entries added:

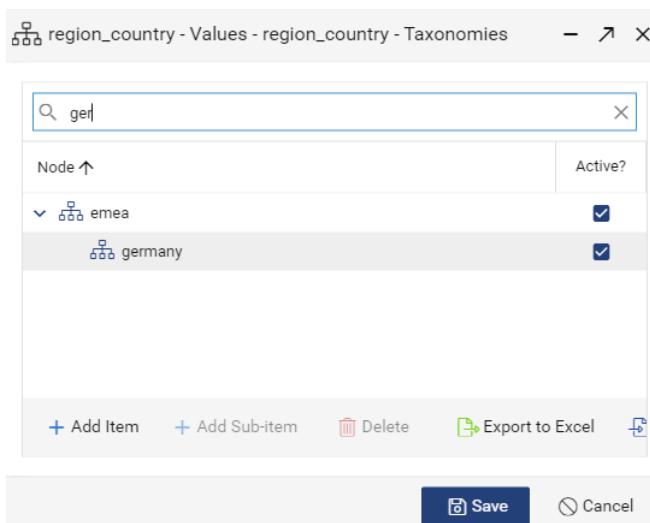
The screenshot shows a more detailed taxonomy structure. The "asia" node has a child node "japan". The "emea" node has children "france" and "germany". The "usa" node has children "uk", "canada", and "usa". All nodes under "asia", "emea", and "usa" are marked as active. The interface is identical to the one in the previous screenshot, with a search bar, a table for selecting active items, and buttons for adding, deleting, saving, and canceling.

13. **Save** the taxonomy

Editing an Existing Taxonomy

To add additional dictionaries to the taxonomy, or to remove or replace the existing dictionaries, highlight a taxonomy and select **Properties**. To modify the items and sub-items within a taxonomy, highlight a select **Edit**.

A search field is located at the top of the **Taxonomy Values** window, which can be used to quickly locate values contained anywhere within the taxonomy hierarchy. The search is non-case sensitive and uses the contains operator:



Deleting or Making Entries Inactive

It is recommended that dictionary items or sub-items which are no longer to be selectable by users are made inactive instead of deleted. **Edit** the taxonomy, highlight the items or sub-items to be removed and unselect **Active** in the right-hand column.

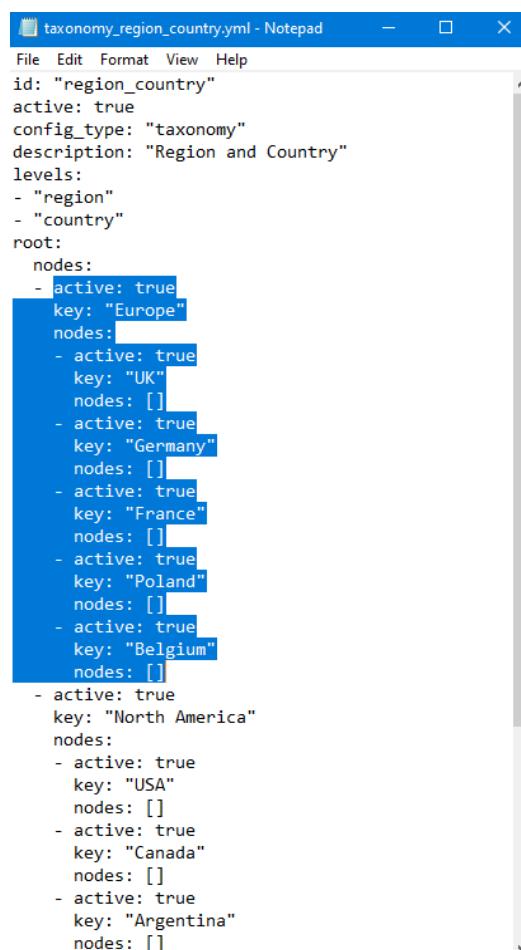
Items and sub-items can be deleted from a taxonomy if required, select the item or sub-item and select **Delete** and then **Yes** to the confirmation message. Items and sub-items deleted from a taxonomy are not deleted from their parent dictionary.

Multiple items can be activated, inactivated or deleted by holding the Ctrl or Shift keys while selecting.

Managing Taxonomies Externally

Taxonomies can be managed outside of CARA.

1. To export a taxonomy, in the **Taxonomies** window highlight a dictionary and select **Export > Export to File**.
2. Select a location to export the .yml file to, or leave the default to save to your browser's default downloads location.
3. The .yml format file can be opened in a text editor such as Notepad or other 3rd party application:



A screenshot of a Windows Notepad window titled "taxonomy_region_country.yml - Notepad". The window contains a YAML configuration file for a taxonomy. The code defines a root node with two levels: "region" and "country". The "region" level has nodes for "Europe", "North America", and "South America". The "Europe" node has children for "UK", "Germany", "France", "Poland", and "Belgium". The "North America" node has children for "USA" and "Canada". The "South America" node has a child for "Argentina". All nodes are active.

```
id: "region_country"
active: true
config_type: "taxonomy"
description: "Region and Country"
levels:
- "region"
- "country"
root:
  nodes:
    - active: true
      key: "Europe"
      nodes:
        - active: true
          key: "UK"
          nodes: []
        - active: true
          key: "Germany"
          nodes: []
        - active: true
          key: "France"
          nodes: []
        - active: true
          key: "Poland"
          nodes: []
        - active: true
          key: "Belgium"
          nodes: []
    - active: true
      key: "North America"
      nodes:
        - active: true
          key: "USA"
          nodes: []
        - active: true
          key: "Canada"
          nodes: []
    - active: true
      key: "South America"
      nodes:
        - active: true
          key: "Argentina"
          nodes: []
```

4. Make changes as necessary and save.
5. To import a taxonomy, with the correct taxonomy highlighted in the CARA **Taxonomies** window select **Import From File**.

6. In the **File Upload** window locate and select the .yml format file to be imported and click **Open**.

7. The taxonomy is updated and a new version created.

In addition to exporting and importing complete taxonomies in a .yml file format, there is a separate option to manage the values within a taxonomy via export to and import from Excel:

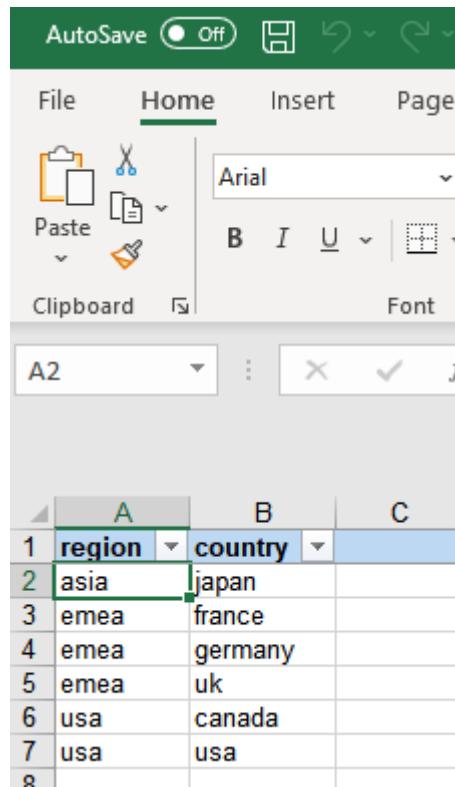
1. Within the **Values** window for a taxonomy select **Export to Excel**:

The screenshot shows a software interface titled 'region_country - Values - region_country - Taxonomies'. It displays a list of taxonomy values under the heading 'Node ↑'. The values listed are 'asia', 'emea', and 'usa', each with a checked 'Active?' checkbox. At the bottom of the window, there are buttons for '+ Add Item', '+ Add Sub-item', 'Delete', 'Export to Excel' (which is highlighted with a blue border), and 'Save' or 'Cancel' buttons.

2. Save the export file or open immediately as per your browser's instructions:

The screenshot shows a Firefox dialog box titled 'Opening cara_query_results.xlsx'. It displays information about the file: 'cara_query_results.xlsx', 'Microsoft Excel Worksheet (6.6 KB)', and 'from: blob:'. Below this, it asks 'What should Firefox do with this file?'. There are two radio button options: 'Open with Microsoft Excel (default)' (selected) and 'Save File'. A checkbox below says 'Do this automatically for files like this from now on.' At the bottom are 'OK' and 'Cancel' buttons.

3. Add and update entries as appropriate and then save the exported file:



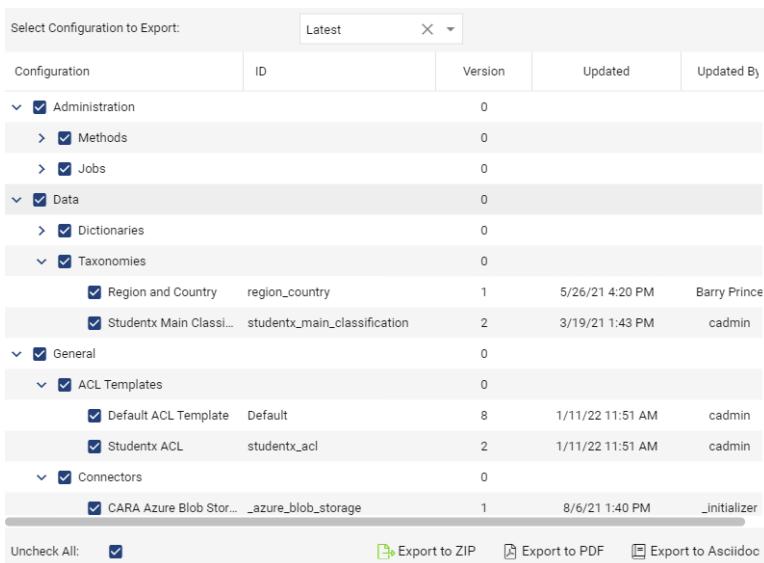
	A	B	C
1	region	country	
2	asia	japan	
3	emea	france	
4	emea	germany	
5	emea	uk	
6	usa	canada	
7	usa	usa	
8			

4. To import an Excel spreadsheet containing taxonomy values, within the **Values** window select **Import from Excel**. Locate the Excel format file and click **Open**. The updated list of values is displayed within the **Values** window, select **Save** to keep the imported changes:

5. Maintenance

5.1. Configuration Export

Description	Configuration export enables the export of CARA environment configuration in .yml format. The latest version of the configuration or specific tagged versions can be exported.
-------------	--



The export can be used as a configuration backup, if problems are encountered the configuration can be re-imported to return to a previous configuration point. The export can also be used as a method of creating additional test environments, configuration can be exported and subsequently imported into an alternative CARA test environment.

Configuration export is not designed or intended to create copies of user generated metadata or content, it cannot be used to export individual documents or objects.

Configuration export does not include users and groups or type definitions. Type definitions can be exported separately as individual .yml files from the Type Definition window.

Configuration export can be selective, all of an environment's configuration can be exported, or just parts of it. Selection of what to export extends to individual configuration elements of a configuration type, for example specific dictionaries could be exported while others are excluded.

It is important to be aware of dependencies when exporting and importing either within the same environment or between different environments. Most configuration is interrelated and therefore dependent upon other elements. For example a form may make use of specific named dictionaries and refer to those dictionaries in its configuration. If the form were exported and imported without the accompanying dictionary it would result in errors.

The exported yml files are divided into folders by configuration group and type and wrapped up into a zip file for convenience.

Configuration can be edited in a yml format from the yml editor built directly into the CARA Control Panel user interface. Directly editing CARA configuration export files outside of CARA is not supported.

Creating a Configuration Export

1. Within the CARA Control Panel select **Configuration Export** from the **Maintenance** section:

Configuration	ID	Version	Updated	Updated By
✓ Administration	0			
✓ Methods	0			
✓ Change Object Owner	cara_object_owner_change	10	1/27/22 10:25 AM	_initializer
✓ Deactivate Provisione...	cara_deactivate_provisioned_u...	10	1/27/22 10:25 AM	_initializer
✓ Dictionary Update Rei...	cara_reindex_objects_on_dict_u...	13	1/27/22 10:25 AM	_initializer
✓ Document Purge Met...	cara_purge	13	1/27/22 10:25 AM	_initializer
✓ Empty Folders Cleanu...	cara_folders_cleanup	13	1/27/22 10:25 AM	_initializer
✓ File Content Store Inte...	cara_file_content_store_integrity	13	1/27/22 10:25 AM	_initializer
✓ LDAP Synchronization...	cara_ldap_sync	13	1/27/22 10:25 AM	_initializer
✓ Logs Cleanup Method	cara_logs_cleanup	13	1/27/22 10:25 AM	_initializer
✓ Recalculate Security ...	cara_security_and_linking_upda...	10	1/27/22 10:25 AM	_initializer
✓ Remove Orphaned Fil...	cara_remove_orphaned_files	13	1/27/22 10:25 AM	_initializer
✓ Send Date Based Noti...	cara_send_date_based_notifica...	13	1/27/22 10:25 AM	_initializer
✓ SharePoint Cleanup M...	cara_sharepoint_cleanup	13	1/27/22 10:25 AM	_initializer

Uncheck All: Export to ZIP Export to PDF Export to AsciiDoc

2. The **Configuration Export** panel lists each configuration group, type and individual element separately, with a corresponding tick box indicating whether the group, type or element will be included or exported from the export. For example the whole of the Administration configuration group can be excluded or included with a single click.
3. The **Configuration Export** window displays the configuration element name, **ID**, **Version**, who the element was last **Updated By** and any **Tags** applied.
4. At the top of the window choose whether the **Latest** version of the configuration is to be exported or a specific tagged version. Note that all elements can be quickly unselected using the **Uncheck All** option.
5. Select the export format, click **Export to Zip**, **Export to PDF** or **Export to AsciiDoc**. Select Export to Zip if the configuration is to be reimported into a CARA environment.
6. The CARA_export.zip file is saved locally, to your default downloads directory.
7. It is recommended changing the name of the .zip file to indicate the tag, state or date of the export.
8. Opening the .zip export file shows that the export is divided into folders for each configuration group, then type, with individual yml files for each configuration element within:

Name	Date modified	Type
_audit_events.yml	31/01/2022 09:51	YML File
_countries.yml	31/01/2022 09:51	YML File
_display_labels.yml	31/01/2022 09:51	YML File
_notification_events.yml	31/01/2022 09:51	YML File
country.yml	31/01/2022 09:51	YML File
region.yml	31/01/2022 09:51	YML File
studentx_section.yml	31/01/2022 09:51	YML File
studentx_zone.yml	31/01/2022 09:51	YML File

5.2. Configuration Import

Description

Configuration import is the import of a previously exported CARA configuration. Configuration export can be used to return to a previous configuration state or when creating a new test environment.

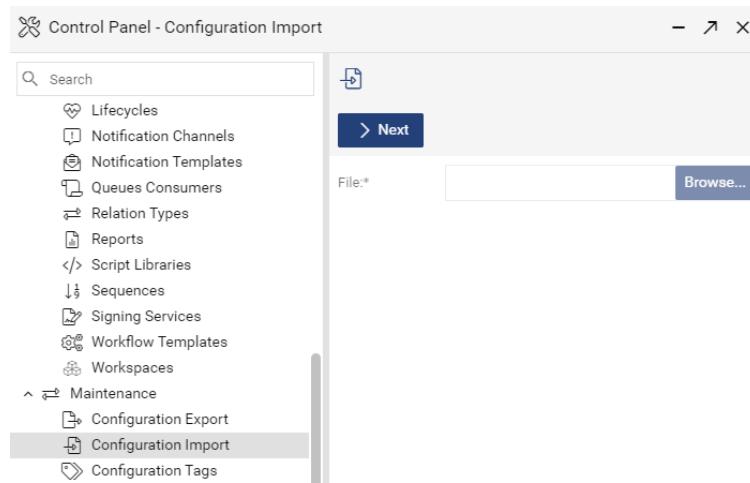
It is important to note:

1. Most CARA configuration is interrelated.
2. CARA configuration exports can be partial or complete.
3. CARA configuration imports can be partial or complete.

It is very important to ensure that the state and completeness of the configuration export is understood before carrying out a configuration import. Importing configuration that is dependent upon configuration elements which are not present in the target environment will result in errors and potential data loss.

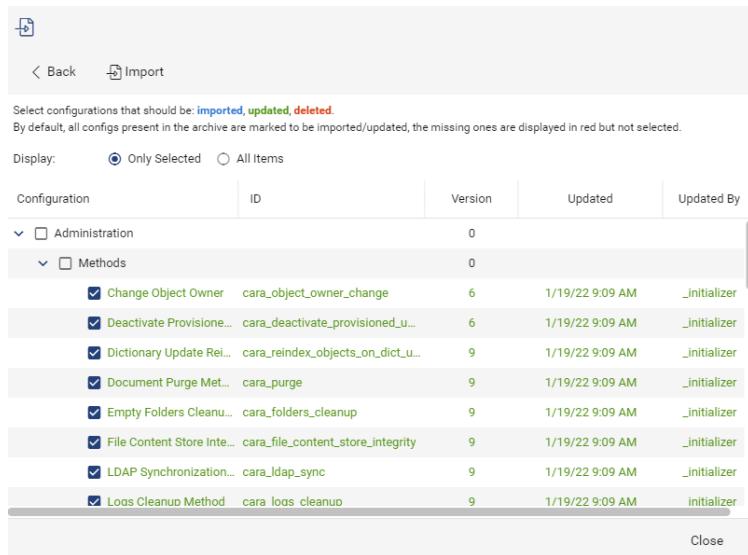
Importing a Configuration

1. Within the CARA Control Panel select **Configuration Import** from the **Maintenance** section.



2. Click **Browse**, locate and highlight the CARA export zip file to be imported and then click **Save**.

3. Click **Next**. The **Configuration Import** panel opens:

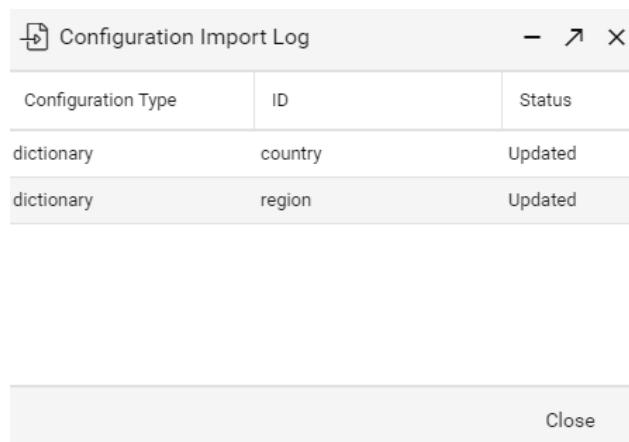


4. **Display**. The configuration import panel initially lists only the configuration elements referenced within the import file, **Only Selected** is enabled by default.

To list all configuration elements of the destination environment in the configuration import panel select **All Items**.

5. Each configuration element can be ignored (unselected), **imported**, **updated** or **deleted** as appropriate.
6. By default configuration elements which are present in the configuration archive but not present in the destination configuration will be marked to be **Imported**. They are listed in blue. Unselect those which should instead be ignored and not imported.
7. By default configuration elements which are present in the configuration archive and also present in the destination configuration will be **Updated**. They are listed in green. Unselect those which should instead be ignored and not updated.

8. By default configuration elements which are not present in the configuration archive to be imported but are present in the destination configuration are displayed in red but are **not** selected, no action will be performed on these elements. Selecting these elements will cause them to be deleted from the destination configuration.
9. Review the list of configuration elements to be ignored, imported, updated or deleted. Click **Import** when ready.
10. Once the import has completed the **Configuration Import Log** window opens:



Configuration Type	ID	Status
dictionary	country	Updated
dictionary	region	Updated

Close

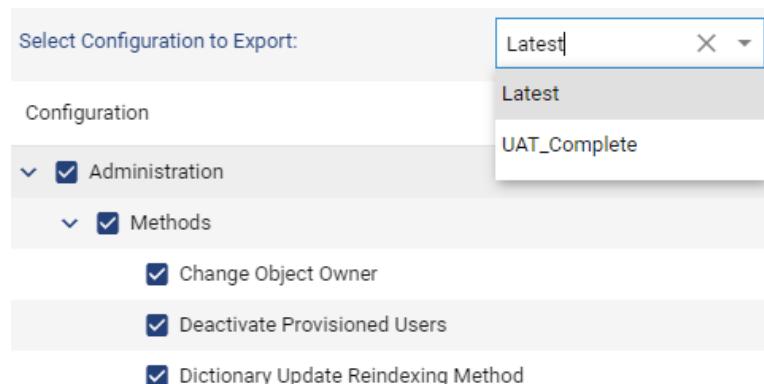
11. For each configuration element where an action was carried out, the **Configuration Import Log** window lists the status as **Updated**, **Added**, or **Deleted**.

Note. Configuration export does not include type definitions. Type definitions must be present in the target system before configuration import. Type definitions are exported and imported separately from the Type Definitions window.

5.3. Configuration Tags

Description

Tags are descriptive labels applied to sets of configuration. They are primarily used when exporting and importing configuration. When exporting configuration the choice is provided to export the configuration as it currently is (Latest), or as it was when a specified tag was applied:



Tags can be used to identify milestones in CARA configuration, for example UAT complete, go-live or “Department x added”.

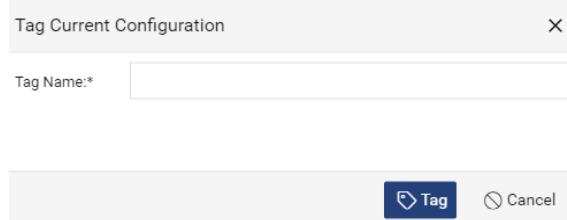
In addition to being selectable within the Configuration Export window, tags are displayed in many of the configuration windows within the CARA Control Panel. The tags applied to a configuration element are shown in a separate column:

ID ↑	Display Name	Version	Updated	Updated By	Tags
_audit_events	Audit Events	3	1/10/22 2:06 PM	_initializer	
_countries	Countries by code	1	3/19/21 1:09 PM	_initializer	UAT_Complete
_display_labels	Display Labels	1	3/19/21 1:09 PM	_initializer	UAT_Complete
_notification_events	Notification Events	2	1/10/22 2:06 PM	_initializer	
country	Country	7	1/19/22 12:05 PM	Barry Prince	
region	Region	7	1/19/22 12:05 PM	Barry Prince	
studentx_section	Studentx Section	2	3/19/21 1:42 PM	cadmin	UAT_Complete
studentx_zone	Studentx Zone	3	3/19/21 1:43 PM	cadmin	UAT_Complete

Creating a Configuration Tag

1. Within the CARA Control Panel select **Configuration Tags** from the **Maintenance** section. The **Latest** tag is present by default.

2. Click **Tag Configuration**, the **Tag Current Configuration** window opens:



3. Enter a **Tag Name** and click **Tag**.
4. The new tag is applied and listed in the Configuration Tags panel.

Please note that there is no GUI based option to remove or edit a tag once applied.

6. General Configuration

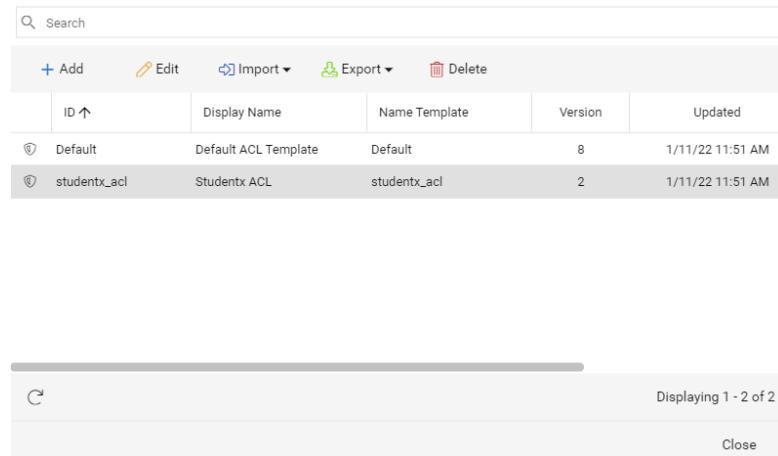
6.1. ACL Templates

Description

ACL (Access Control List) templates determine the security applied to documents and objects. An ACL consists of a list of users and/or groups and the permissions they are granted by the ACL.

Each document or object will have a single ACL, applied initially when it is created. Type specific security configuration determines which ACL applies to which documents. The ACL applied may be updated or changed by a lifecycle or other process.

Documents and objects can share a common ACL or have an ACL which is unique to them.



The screenshot shows a list of ACL templates. At the top, there is a search bar and several action buttons: '+ Add', 'Edit', 'Import', 'Export', and 'Delete'. Below this is a table with columns: ID, Display Name, Name Template, Version, and Updated. There are two entries:

ID	Display Name	Name Template	Version	Updated
Default	Default ACL Template	Default	8	1/11/22 11:51 AM
studentx_acl	Studentx ACL	studentx_acl	2	1/11/22 11:51 AM

At the bottom of the list, there is a footer with a 'Close' button.

Permission

Levels

The following is the list of permissions which can be granted to a user or group within an ACL. These permissions are hierarchical, for example a user with Read Content permissions can also Read Properties.

1. **None.** The object will not be visible to the user at all, it will not appear in search results or in folders when browsed.

2. **Read Properties.** A user can view the properties of an object but not the associated content file.
3. **Read Content.** View the content file. Please note a user may separately be limited to only viewing a particular rendition format such as .pdf
4. **Annotate.** A user can add comments/annotations to a document (usually a .pdf format rendition) but cannot edit the content file.
5. **Version.** A user can edit and check-in a document, creating a new version of the document.
6. **Save Properties.** A user can edit the properties of a document.
7. **Save Content.** A user can edit and save the content of a document.
8. **Delete.** A user can delete the specific version of the document selected.

Creating an ACL Template

1. Within the CARA Control Panel select **ACL Templates** from the **General** section.
2. Select **+ Add**, the **New ACL Template** window opens:

The screenshot shows the 'New ACL Template' configuration screen. At the top, there are fields for 'ID:' and 'Display Name:', both marked with asterisks indicating they are required. Below these is a 'Type:' section with three options: 'General - Not Based on Properties' (selected), 'Property Based - Properties Used in the Name Can Then Be Used in the Definition', and 'Object Based - All Object Properties Can Be Used'. A note below the 'Object Based' option states: 'Selecting this option will result in an ACL instance per object'. The main area is a table titled 'User/Group' with columns 'User/Group' and 'Permissions'. It contains three rows: '_admin' with 'Default: Delete', '_owner' with 'Default: Delete', and '_all' with 'Default: Read Properties'. At the bottom of the table are buttons for '+ Add User/Group', 'Edit', and 'Remove'. Below the table are 'Save' and 'Cancel' buttons.

3. Enter a unique **ID** (name) for the ACL template.
4. Enter a **Display Name**.
5. Select the **ACL Type**:
 - a. **General**. A simple static ACL that does not make use of properties in its name or user/group permissions. This is the most common type of ACL. Multiple documents can use the same general ACL
 - b. **Property Based**. A dynamic ACL that can use property values to determine both user/group names and their permission levels. Multiple documents can use the same property based ACL.
 - c. **Object Based**. A dynamic ACL that can use property values to determine both user/group names and their permission levels. A separate ACL is created for each document/object instance.
6. For property based and object based ACL's select the **Object Type**. This option is not required for general ACL's.
7. **Property** based ACL's require a **Name Template** to be defined. The name template must include any properties that are to be

used when determining the users/groups listed or any conditional permissions. **Object** based properties will automatically have a unique name generated which includes the object ID and do not need a name template configured:

New ACL Template

ID:*

Display Name:*

Type:

- General - Not Based on Properties
- Property Based - Properties Used in the Name Can Then Be Used in the Definition
- Object Based - All Object Properties Can Be Used

⚠ Selecting this option will result in an ACL instance per object

Object Type:*

Name Template:*

User/Group Permissions

_admin	Default: Delete
_owner	Default: Delete
_all	Default: Read Properties

+ Add User/Group Edit Remove

Save Cancel

- a. Click **+ Add Element**. The **New Name Template** window opens:

New Element

Element Type:

- Static Text
- Attribute Value

Text:*

Save Cancel

- b. Select the **Element Type** as either **Static Text** or **Attribute Value**.

- c. If selecting **Static Text** enter a value in the **Text** field as it should be displayed, including capital letters if required.
- d. If selecting **Attribute Value**, select an attribute from the drop-down list. **Subsequent options are attribute type dependant.**

Enter an **Unspecified Value**, this will be used in the ACL name if no valid value for the specified attribute is available.

If the **Attribute** selected was a **text** based attribute:

The screenshot shows the 'New Element' dialog box with the following fields filled in:

- Element Type:** Attribute Value (radio button selected)
- Attribute:** Zone - doc.zone (String)
- Unspecified Value:** unknown
- Dictionary:** (empty dropdown)
- Dictionary Alias:** (empty dropdown)
- Case:** Regular (radio button selected)
- Truncate After:** Char(s) (empty input), Replace Characters: (empty input), With: (empty input)

At the bottom right of the dialog are two buttons: **Save** (dark blue) and **Cancel**.

Optionally select a **Dictionary** and whether a **Dictionary Alias** should use a specified fixed alias (such as English, French, German etc..) or an alias based on the user's locale. This is useful where users may use multiple languages but a single language is used in the ACL name for consistency.

Select the **Case** for the ACL Name Template element as either **Regular** (the text case will be the same as that entered, including mixed-case), all **Upper-case** or all **Lower-case**.

Optionally select to **Truncate** after x number of characters. This can be useful where attributes values are potentially long but the ACL name length should be kept within a specified limit.

Optionally where attributes values are a fixed format, characters in a specified position can be replaced, enter a **Replace Characters** value and the alternative values to replace them **With**.

If the **Attribute** selected was a **date** based attribute enter a **Date Format**.

The screenshot shows the 'New Element' dialog box. The 'Element Type:' section has 'Attribute Value' selected. The 'Attribute:' field contains 'Created - created (DateTime)'. The 'Unspecified Value:' field contains 'unknown'. The 'Date Format:' field is empty. At the bottom right are 'Save' and 'Cancel' buttons.

If the **Attribute** selected was a **numeric** based attribute enter a **Number Format**.

The screenshot shows the 'New Element' dialog box. The 'Element Type:' section has 'Attribute Value' selected. The 'Attribute:' field contains 'Sequence No - sequence_no (Integer)'. The 'Unspecified Value:' field contains 'unknown'. The 'Number Format:' field is empty. At the bottom right are 'Save' and 'Cancel' buttons.

- e. **Save** the new name template element. Back in the **New ACL Template** window add additional elements to the Name Template as required. Name template elements can be edited, removed and their order changed as needed.

8. Specify the **User/Groups** and their **Permissions** within the ACL.

Three system groups are included by default and cannot be removed, however the permissions granted to them by default can be modified:

- a. **_admin**. The system administrator group. Granted delete by default.
- b. **_owner**. The user marked as the document's owner, typically the user that created the document, granted delete by default.
- c. **_all**. A special group containing all users in the system. Used to grant default permissions. Granted read properties by default.

Other users and groups can be added as required. For **general** ACL's static users and groups are selected. For **property based** and **object based** ACL's users and groups can also be dynamically generated based on attributes.

9. To add a user or group click **+ Add User/Group**, the **Add User/Group** window opens:

The screenshot shows the 'Add User/Group' dialog box. At the top, there are two radio buttons: 'Static' (selected) and 'Dynamic'. Below this are fields for 'User/Group:' and 'Default Permission:'. Under 'Default Additional Permissions', there are checkboxes for 'Annotate' and 'Relate'. A table below these fields has columns for 'Permission' and 'Condition'. At the bottom of the dialog are buttons for '+ Add', 'Edit', and 'Remove'. At the very bottom are 'Save' and 'Cancel' buttons.

10. To add a **Static** user or group, scroll through the **User/Group** drop-down list or type a name into the field. Entering 3 or more characters from a name will automatically filter the list.
11. To add a **Dynamic** user or group, select Source: Dynamic and click **+Add Element** beside the **Name Template** field, opening the **New Name Template** window. The options presented are the same as those when creating the ACL's name template.
12. Once a static or dynamic user/group has been added, in the **Add User/Group** window select a **Default Permission** from the **Default Permission** drop-down list:

The screenshot shows the 'Add User/Group' dialog box. At the top, there are two radio buttons: 'Static' (selected) and 'Dynamic'. Below that is a text input field labeled 'User/Group:' containing 'training_reader'. Underneath is another text input field labeled 'Default Permission:' containing 'Read Content'. Below these fields is a section for 'Default Additional Permissions' with checkboxes for 'Annotate' and 'Relate'. At the bottom of the dialog is a table with two columns: 'Permission' and 'Condition'. Below the table are three buttons: '+ Add' (highlighted in blue), 'Edit', and 'Remove'. At the very bottom are 'Save' and 'Cancel' buttons.

13. Conditions can be added, which if returned as true would grant alternative permissions, overriding the default permission. Click **+Add** to open the **New Condition** window:

The screenshot shows the 'New Condition' dialog box. It has three main input fields: 'Property:' with 'doc.zone', 'Value:' with 'Regulatory', and 'Permission:' with 'Read Properties'. Below these is a section for 'Additional Permissions' with checkboxes for 'Annotate' and 'Relate'. At the bottom are 'OK' and 'Cancel' buttons.

- a. Select a **Property** that the condition is to be based on.
- b. Enter the **Value** required for the condition to return as true.
- c. Select the **Permission** granted if the condition is true and if the **Additional Permissions** of **Annotate** and **Relate** will also be granted.
- d. Click **OK**, returning to the **Add User/Group** window.
- e. Further conditional permissions can be added as required.
- f. **Save** the new user/group permissions, returning to the **New ACL Template** window:
 14. Add further users or groups to the ACL template and set permissions as required.
 15. **Save the New ACL Template.**

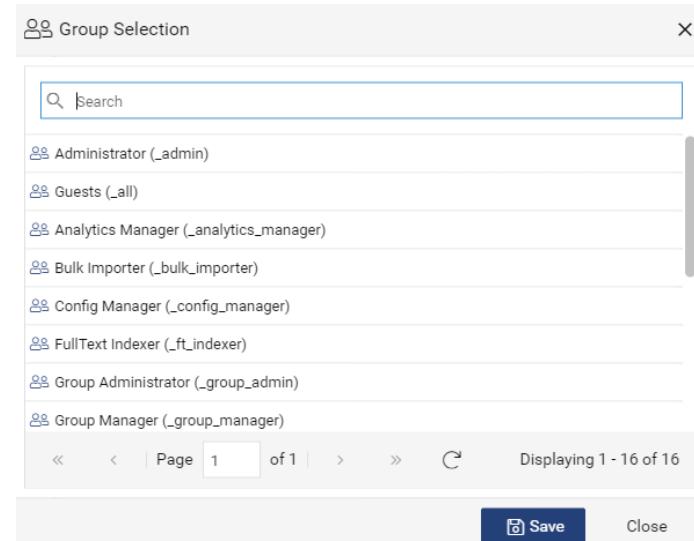
6.2. Capabilities

Description	Capabilities determine the system wide functions which groups of users are permitted, such as accessing tasks and copying cell or row content. These are different from the type specific capabilities which are configured from the Type Configuration panel and which are detailed in a separate section.
-------------	---

Capability	Default	_admin	_config_manager
Copy Cell Content to Clipboard	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Copy Row Content to Clipboard	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Signing Requests	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Tasks - Inbox & Completed Tasks	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Tasks - Inbox - Multiple Tasks Processing	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Tasks - Related Workflows	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Tasks - Send Workflow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Tasks - Workflow Reporting	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

List of System Capabilities	<ol style="list-style-type: none"> 1. Copy Cell Content to Clipboard. 2. Copy Row Content to Clipboard. 3. Signing Requests. 4. Tasks - Inbox & Completed Tasks. 5. Tasks - Inbox Multiple Tasks Processing. 6. Tasks - Related Workflows. 7. Tasks - Send Workflow. Each workflow template's configuration specifies which users are allowed to initiate its tasks. 8. Tasks - Workflow Reporting.
-----------------------------	---

Adding a Group to Capabilities	<ol style="list-style-type: none"> 1. Within the CARA Control Panel select Capabilities from the General section. 2. Select + Add Role, the Group Selection window opens.
--------------------------------	---



3. Highlight a group. Groups can be located by use of the scroll bar or entering a partial or full name in the search field.
4. Click **Save**, returning to the **Capabilities** panel.

Setting Group Capabilities

Newly added groups initially have no capabilities.

1. In the **Capabilities** window locate the appropriate group column and click once in the cell corresponding to the required capability, a tick will be displayed.

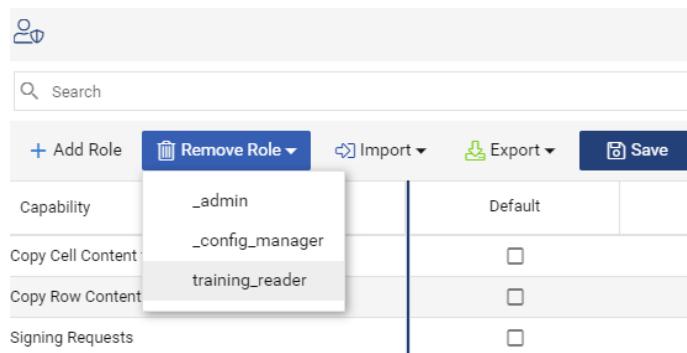
Capability	Default	_admin	_config_manager	training_reader
Copy Cell Content to Clipboard	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Copy Row Content to Clipboard	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Signing Requests	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Tasks - Inbox & Completed Tasks	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Tasks - Inbox - Multiple Tasks Processing	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Tasks - Related Workflows	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Tasks - Send Workflow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Tasks - Workflow Reporting	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

2. Each time a cell is clicked it will toggle between enabling and disabling a capability, set capabilities as appropriate.
3. **Save** once complete.

Removing Capabilities from Groups

To remove capabilities from a group, untick the capability from the corresponding cell in group column and **Save**.

To entirely remove a group from capabilities, select the group from the **Remove Role** drop-down list and then **Save**:



Changing the Default Capability

A **Default** role is present within capabilities by default and cannot be removed. It represents the capabilities available to all users. Add and remove capabilities to the default role as required. No capabilities are required to be available to end users.

6.3. Collaboration Services

Description	<p>Collaboration services allow multiple users to simultaneously update documents managed within CARA via a 3rd party collaboration service. CARA supports Google Docs and Microsoft SharePoint/Office365 as collaboration services.</p> <p>Please note use of collaboration services requires users to have accounts with the same email address enabled in both CARA and the collaboration service to be used.</p> <p>Details of how CARA will connect to the chosen collaboration service must be configured in advance.</p>
--------------------	--

ID ↑	Display Name	Service Type	Version
Generis-Google	Google	GoogleDocs	1
Generis-Office365	Generis-Office365 Collaboration	Office365	2

Creating a Collaboration Service Configuration

1. Within the CARA Control Panel select **Collaboration Services** from the **General** panel.
2. Select **Add**, the **New Collaboration Service** window opens:

The screenshot shows a configuration dialog for a 'New Collaboration Service'. The 'General' tab is active. The 'ID:' field is empty. The 'Display Name:' field is empty. The 'Service Type:' dropdown is empty. The 'Source Formats:' list is empty. The 'Object Name Pattern:' and 'Unique Name Pattern:' lists are empty. There is an unchecked checkbox for 'Generate Unique Names'. At the bottom are 'Save' and 'Cancel' buttons.

3. **ID.** Enter a unique name for the collaboration service configuration. The ID can only contain letters, numbers and underscores and hyphens. It must start with a letter.
4. **Display Name.** Enter a user friendly display name.
5. **Service Type.** Select the 3rd party collaboration service to be used from the drop-down list. Google Docs and Office 365 are supported. Office 365 requires additional configuration as detailed below.
6. **Source Formats.** Optionally specify which documents formats will be supported in the collaboration, separated by a comma. For example doc, docx.
7. **Object Name Pattern.** Enter the pattern to be used for object/document names when they are imported into the collaboration service. Object properties can be included as variables, for example \${object_name}.
8. **Unique Name Pattern.** If Generate Unique Names is selected, enter the pattern to be appended to the object name pattern.

9. **Generate Unique Names.** Select to have unique names generated for objects within the collaboration service, which appends the unique name pattern to the object name pattern.

Additional Fields for Office 365:

ID:	Generis-Office365
Display Name:*	Generis-Office365 Collaboration
Service Type:*	Office 365
Source Formats: ?*	doc,docx
Object Name Pattern: ?*	\$(object_id)_(object_name)
Unique Name Pattern: ?*	'(dd-MMM-yyyy HH-mm-ss z)'
<input checked="" type="checkbox"/> Generate Unique Names ?	
URL:*	
User Name:*	
Password:*
Site Name:*	
Target Path:*	
Options:	<input checked="" type="checkbox"/> Enable Recycle Bin <input checked="" type="checkbox"/> Enable Change Tracking on Export <input checked="" type="checkbox"/> Disable Change Tracking on Import
Tracking Password:
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

10. **URL.** Enter the URL used to connect to the Office 365 service.

11. **User Name.** Enter the username to be used by CARA when connecting to the Office 365 service.

12. **Password.** Enter the password to be used by CARA when connecting to the Office 365 service.

13. **Site Name.** Enter the name of the Office 365 instance to be connected to.

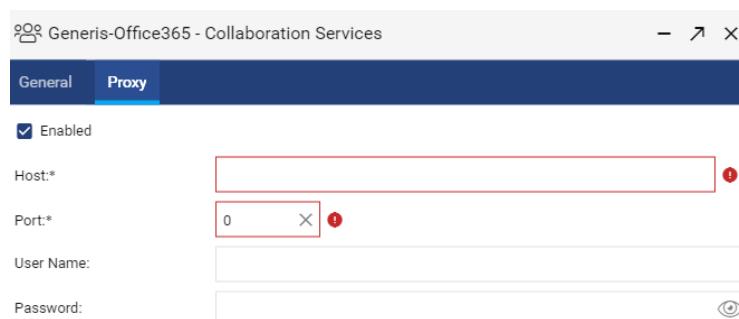
14. Target Path. Enter the root target path for documents within the Office 365 instance.

15. Enable Recycle Bin. Tick to enable recycle bin functionality within the collaboration service.

16. Enable Tracking. Tick to automatically enable document tracking when documents are imported into Office365. When selected, an additional **Tracking Password** field is displayed. The tracking password is subsequently used to turn tacking on and off within documents that are sent for collaboration.

Proxy Settings.

If the collaboration service connection is to be made via a proxy server, enter the following details:



1. **Enabled.** Tick to have the collaboration service connection made via proxy, using the details provide below.
2. **Host.** Enter the hostname of the proxy server.
3. **Port.** Enter the port number used to connect to the proxy server.
4. **User Name.** Enter the user name to be used when connecting to the proxy server.

5. **Password.** Enter the password to be used when connecting to the proxy server.

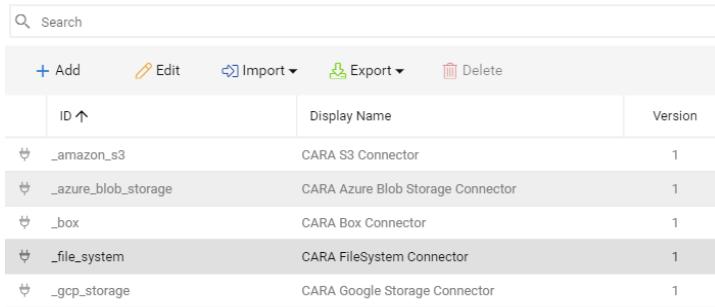
A Note About Google Docs:

Additional settings for how CARA connects to Google Docs as a collaboration service are set within a dedicated JSON file on the server, unlike for Office365 where all connection settings are set within the user interface windows shown above.

6.4. Connectors

Description

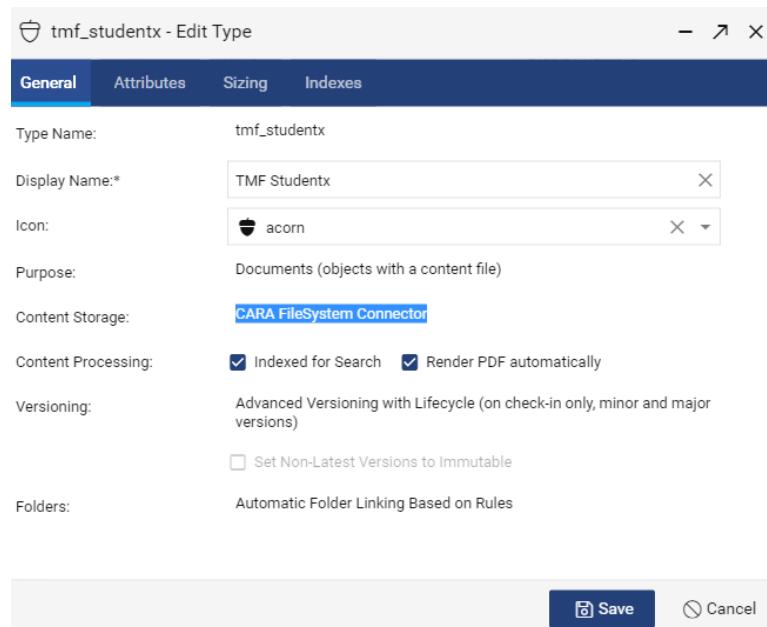
CARA supports multiple services and locations for content storage, such as simple file shares, Amazon S3, Box and Azure blob storage. Each connector represents a storage service and/or device:



A screenshot of a web-based application interface for managing connectors. At the top, there is a search bar and a toolbar with buttons for 'Add' (+), 'Edit' (pencil), 'Import' (cloud), 'Export' (file), and 'Delete' (trash). Below the toolbar is a table with three columns: 'ID ↑', 'Display Name', and 'Version'. The table contains five rows, each representing a connector:

ID ↑	Display Name	Version
_amazon_s3	CARA S3 Connector	1
_azure_blob_storage	CARA Azure Blob Storage Connector	1
_box	CARA Box Connector	1
_file_system	CARA FileSystem Connector	1
_gcp_storage	CARA Google Storage Connector	1

In the configuration for a type, when Purpose is set as Documents, a connector is selected which determines for the storage used for associated content files:



A screenshot of a 'tmf_studentx - Edit Type' configuration dialog. The dialog has tabs for 'General', 'Attributes', 'Sizing', and 'Indexes', with 'General' being the active tab. The configuration fields are as follows:

- Type Name: tmf_studentx
- Display Name: TMF Studentx
- Icon: acorn
- Purpose: Documents (objects with a content file)
- Content Storage: CARA FileSystem Connector
- Content Processing: Indexed for Search Render PDF automatically
- Versioning: Advanced Versioning with Lifecycle (on check-in only, minor and major versions)
 - Set Non-Latest Versions to Immutable
- Folders: Automatic Folder Linking Based on Rules

At the bottom right are 'Save' and 'Cancel' buttons.

For each type of connector being used, a corresponding connector.yml file is configured and placed on the application

server. The connector configuration in the control panel points to that file.

The Configuration File Examples chapter in this manual includes examples of the following connector.yml files:

File System: cara_connector-fs.yml

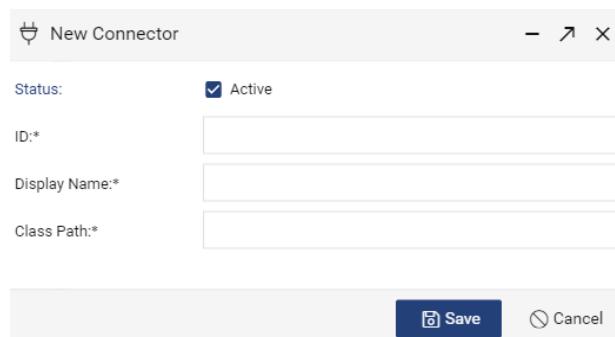
Amazon S3: cara_connector-s3.yml

Box: cara_connector-box.yml

Azure Blob Storage: cara_connector-azure-blob-storage.yml

Creating a Connector

1. Within the CARA Control Panel select **Connectors** from the **General** section.
2. Select **Add**, the **New Connector** window opens:



3. **Status.** The connector must be **Active** if it is to be selectable within type configuration.
4. **ID.** Enter a unique ID. The ID can only contain lowercase letters, numbers, underscores and hyphens. It must start with a letter.
5. **Display Name.** The Display Name can contain mixed-case letters and spaces.
6. **Class Path:** Enter the class path for the connector's configuration file. Each connector implementation has its own

configuration files which are deployed on the server. The configuration file sets connector specific options such as the type of encryption used, storage path or device. The configuration files are .yml format.

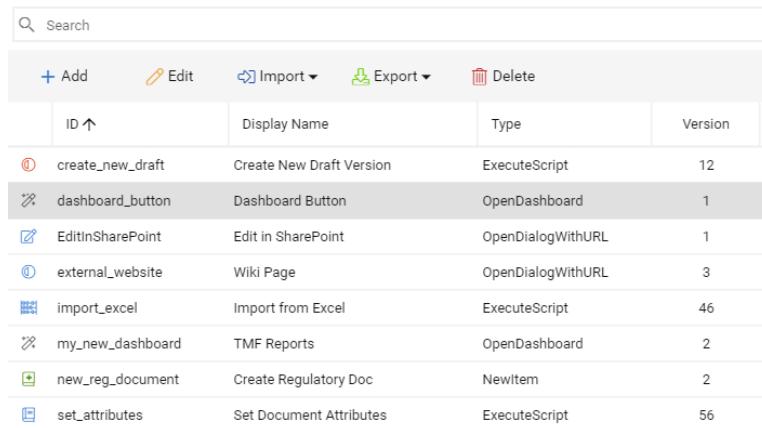
6.5. Custom Actions

Description

Custom actions provide a method of extending the functionality of CARA. They can be based on various technologies and are primarily deployed as menu options or toolbar buttons for users to select, which then launch custom actions applied to currently selected documents.

As part of a custom action users may be presented with forms with variables to enter or update. Custom actions can temporarily provide users with additional permissions required to carry out an action without the need to separately alter the object's ACL.

Custom actions are created and listed within the **Custom Actions** window within the Control Panel, which displays their **ID**, **Description**, **Type**, **Version**, date **Last Updated** and who the custom action was last **Updated By**.



ID ↑	Display Name	Type	Version
create_new_draft	Create New Draft Version	ExecuteScript	12
dashboard_button	Dashboard Button	OpenDashboard	1
EditInSharePoint	Edit in SharePoint	OpenDialogWithURL	1
external_website	Wiki Page	OpenDialogWithURL	3
import_excel	Import from Excel	ExecuteScript	46
my_new_dashboard	TMF Reports	OpenDashboard	2
new_reg_document	Create Regulatory Doc	NewItem	2
set_attributes	Set Document Attributes	ExecuteScript	56

**Creating
Custom Actions**

1. Within the CARA Control Panel select **Custom Actions** from the **General** panel.
2. Click **Add**, the **New Custom Action** window opens:

The screenshot shows the 'New Custom Action' configuration window with the 'General' tab selected. The window has a title bar with icons for minimize, maximize, and close. Below the title bar is a navigation bar with tabs: General (selected), Definition, Variables, Restrictions, and Options. The main area contains fields for 'ID:' (with placeholder 'Internal Name of the Custom Action'), 'Display Name:' (placeholder 'Internal Name of the Custom Action'), 'Label:' (dropdown menu with placeholder 'Button and Menu Item Label'), 'Icon:' (dropdown menu with placeholder 'Button and Menu Item Icon'), 'Icon Color:' (color picker with 'Default' option selected), 'Main Menu Location:' (dropdown menu with 'Tools' selected), and 'Status:' (checkbox 'Active' checked). At the bottom right are 'Save' and 'Cancel' buttons.

3. The **Custom Action** window is divided into the following tabs: General, Definition, Variables, Restrictions and Options. The configuration options presented on many of the tabs will vary according to the **Action Type** selected on the **Definition** tab.

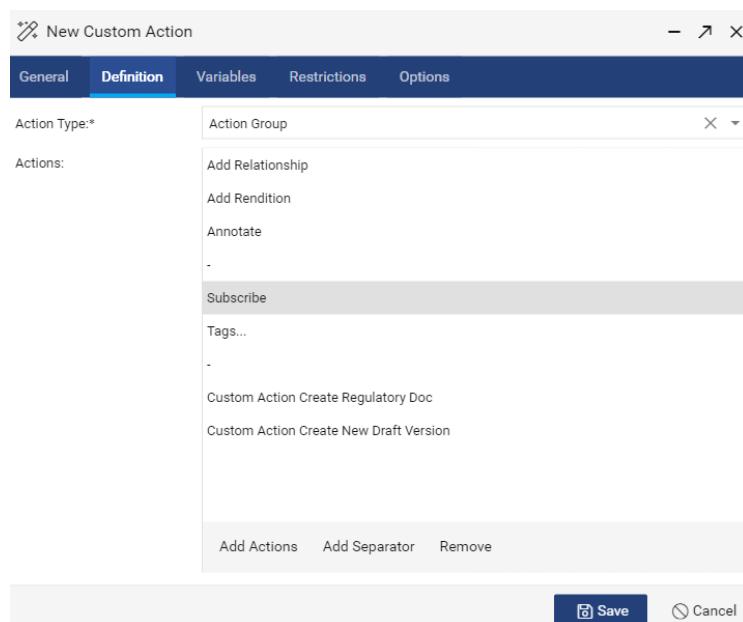
General Tab

1. Enter a unique **ID**.
2. Enter a **Description**.
3. Enter a user friendly **Label** which can be displayed as a toolbar button and menu item. This can either be entered manually or an entry can be chosen from the _display_labels dictionary by clicking the right-hand down arrow.
4. Select an **Icon** and **Icon Color** from the drop-down lists.
5. **Main Menu Location**. Select if the custom action should appear under the **Tools**, **Settings** or **Help** menus. If the field is cleared the custom action will not appear under any main menu, though it can still be added as a view action.
6. Select if the custom action will be **Active**.

Definition Tab

1. Select the **Action Type** from the drop-down list:

2. **Action Group**. Used to group together a set of actions (both standard and custom) and have them available to be displayed within a toolbar or menu bar. The **Actions** field displays:



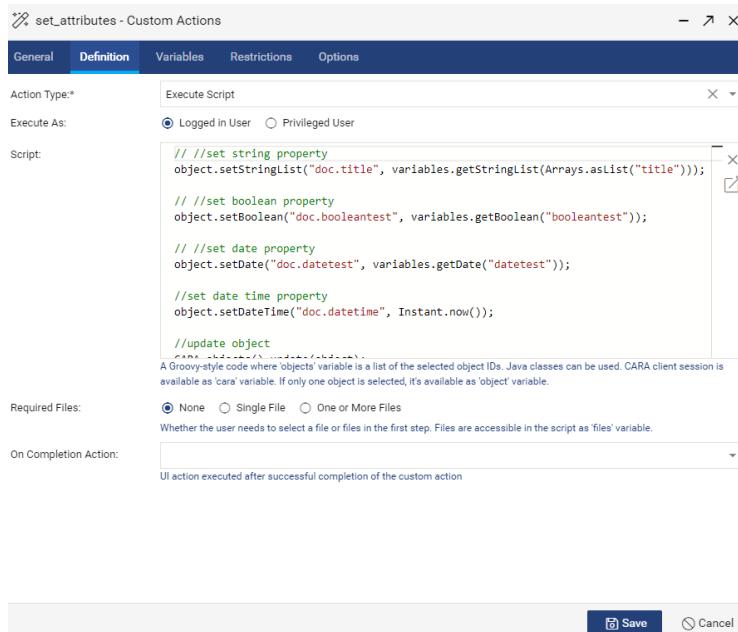
- a. Click **Add Actions** to open the **Available Selection** window. Move the actions to be included in the action group to the right-hand side panel either by click and drag or using the central arrows.

- b. Re-order the selected actions in the right-hand panel as preferred by click and drag or using the central arrows.

- c. Save the **Action Selection**, returning to the main custom action window.

- d. Optionally **Add Separators** to visually group set of actions within the Action Group. The separator will be added beneath the selected action:

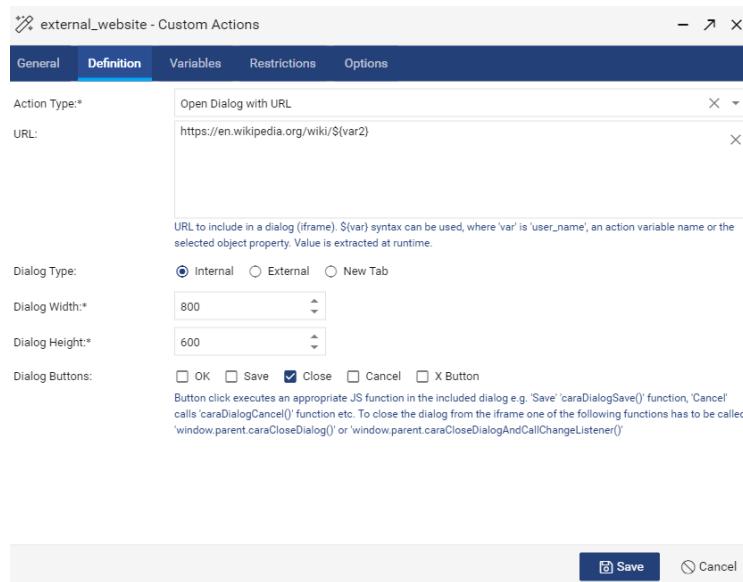
3. **Execute Script.** In this context a script is a Groovy-style code where the 'objects' variable is a list of the selected object IDs. Java classes can be used. The CARA client session is available as a 'cara' variable. If only one object is selected, it's available as an 'object' variable.



- Select whether the script should **Execute As** the currently **Logged in User** or a **Privileged User**. Selecting Privileged User allows users to perform an action that they would not normally be allowed to carry out. For example if it was necessary for users to update one specific attribute, when they normally have read-only access.
- Enter the script in the **Script** panel. Scripts can include variables, which are defined on the subsequent **Variables** tab.
- Required Files.** Select if the user needs to select a **Single File** or **One or More Files** in the first step, or if no file selection is required (**None**).
- On Completion Action.** Select a user interface action to be executed after successful completion of the action. All options

below apart from File Download require specifying a variable which identifies the object to be acted upon:

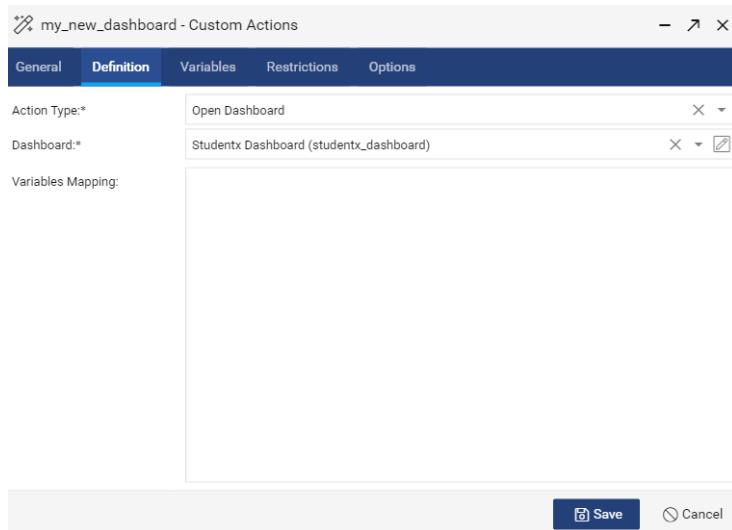
- i. **Open Properties.**
 - ii. **View.**
 - iii. **View PDF.**
 - iv. **Export.**
 - v. **File Download.** Additionally specify the following:
File Bucket ID Variable: variable name holding the transferId. The variable must first be defined from the **Variables** tab of the custom action.
File Action: Either open or save the file.
 - vi. **Navigate To.** Open the object specified by Object ID variable.
4. **Open Dialog with URL.** Opens a URL either in a new web page or internally within the CARA application and can be passed variables such as properties of the currently selected object.



- a. **URL.** Enter the URL to be opened. This can include variables using the syntax \${variable_name}.

- b. **Dialog Type.** Select whether the URL should be opened in an **Internal** or **External** web page.
- c. **Dialog Width.** Specify the web page **Width** in pixels.
- d. **Dialog Height.** Specify the web page **Height** in pixels.
- e. If the **Dialog Type** was set to **Internal**, select which **Dialog Buttons** are to be available.
- f. If the **Dialog Type** was set to **External**, enter a **Window Name**, which specifies the target attribute or the name of the window. The following values are supported:
 - _blank.** The URL is loaded into a new window, the default.
 - _parent.** The URL is loaded into the parent frame.
 - _self.** The URL replaces the current page.
 - _top.** The URL replaces any framesets that may be loaded.
 - name.** The name of the window (note: the name does not specify the title of the new window).
- g. If the **Dialog Type** was set to **External**, enter a **Window Spec**, which is a comma-separated list of items. The following settings are supported:
 - fullscreen
 - location
 - menubar
 - resizable
 - scrollbars
 - status
 - titlebar
 - toolbar
 - top
 - left

5. **Open Dashboard.** Reports are grouped together for display within dashboards. Dashboards can be added as toolbar or menu bar button **Action** for a type **View** once they have been assigned to an **Open Dashboard** custom action.



Please see the separate chapter for details of creating and configuring reports.

- a. **Dashboard.** Select the Dashboard to be made available. Once selected, the dashboard's configuration can be opened in a separate window by clicking the pencil icon.
 - b. **Variables Mapping.** Optionally enter initial variable values which will be passed to the reports contained within the chosen dashboard. For example, a report might expect a user to select a status of documents to be reported on, the variables mapping could pass an initial value such as Draft or Approved in order that a default report can be displayed before the user has made a selection.
6. **New Item.** Provide users with a toolbar or menu button for quickly creating new documents with a specific Creation Method/Template and Classification. This is useful where users repeatedly create similar documents. Once the custom action has

been defined as a **New Item**, it can be added as an **Action** within the related type's **View** configuration.

- a. **Object Type.** Select the object type of the new item from the drop-down list.
- b. **Creation Method.** Select the content creation method.
- c. **Classification.** Select the classification of the new object.
- d. **Template or eForm.** This field is only displayed if **From Template** or **From eForm** were selected as **Creation Method**. Select the specific content **Template** or **eForm** to be copied.
- e. **Object Attribute.** Initial property values can be set. Select an attribute and enter the initial value. Multiple attributes can be set.

Variables Tab

1. Custom action types such as **Execute Script** can make use of variables. Where one or more variables are required they are

defined in the **Variables** tab. If users are expected to provide a value for one or more variables these are presented to users on a custom **Variables Form**.

ID ↑	Display Name	Type	Multi-V...
var2	var2	String	

+ Add + Add Sub-Attribute Edit Delete Edit Variables Form

Custom Save Button Label:

Variables Initialization Script:

Save **Cancel**

2. Click **Add** to open the **New Variable** window:

Name:*

Display Name:*

Data Type:*

Multi-Value

Source Object Property to Copy as Default:

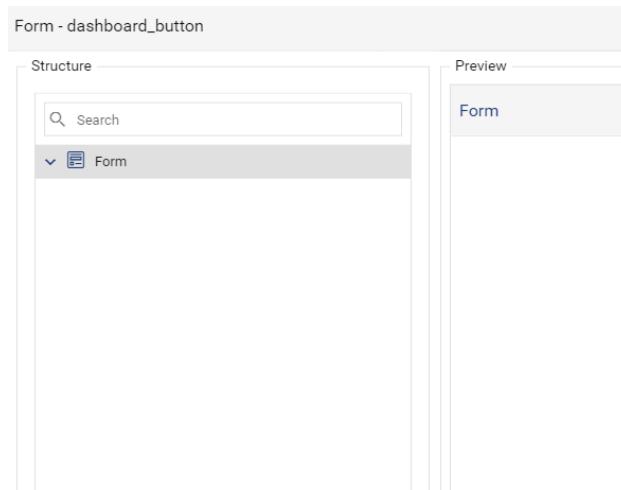
If the action will only be performed on one object, the value of the attribute will be copied as the default value of the variable.

Save **Cancel**

3. Enter a unique **Name** for the variable.

4. Enter a **Description** for the variable.

5. Select the **Data Type** from the drop-down list.
6. Tick **Multi Value** if the variable can accept multiple values, for example documents may have multiple authors.
7. Optionally enter a **Source Object Property to Copy as Default**, meaning an initial value for the variable can be passed to the script, allowing it to execute and return results as soon as selected, instead of waiting for input from the user.
8. **Save** the new variable, returning to the **Variables** tab.
9. **Add** additional new variables if required.
10. If values for the variables are to be entered by the user, a custom variables form needs to be defined for the purpose. Click **Edit Variables Form**, launching the form creator window. The options available are the same as for standard document property forms, see the separate Property Forms chapter for details.



11. Please note that when adding fields onto a custom action variable form, only the attributes defined as variables can be added:

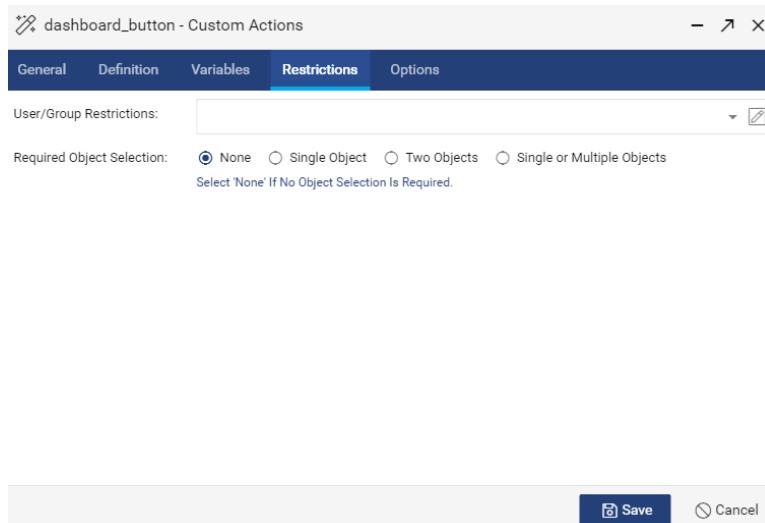
Save the variables form once complete, returning to the **Variables** tab.

12. **Custom Save Button Label.** Optionally enter an alternative user friendly label for the **Save** button which appears at the bottom of the variables form.
13. **Variables Initialization Script.** A script can be added to dynamically set the initial variable values.

Restrictions Tab

In the **User/Group Restrictions** field, select which users or groups will have access to the custom action. Click the drop-down arrow in the corner of the **User/Group Restrictions** field to open the **User and Groups Selection** window.

Please note access to custom actions is further restricted by users only seeing the custom actions which have been explicitly made available to them as toolbar or menu buttons in a **Type** based **View** which they have been given access to.



Select the **Required Object Selection**.

If the custom action does not perform any actions, such as when defined as an Action Group, or does not require the user to select an object for the action to be performed on, select **None**.

If the custom action performs an action on a user selected object or objects, such as when executing a script that updates attribute values, select **Single Object**, **Two Objects** or **Single or Multiple Objects** and complete the subsequent options:

The screenshot shows the 'Restrictions' tab of a custom action configuration. The 'User/Group Restrictions' field contains 'training_reader'. The 'Required Object Selection' section has a radio button selected for 'Single Object'. The 'Audit Event' field is set to 'Update'. The 'Object Type' field is set to 'tmf_studentx'. The 'Required Permission' field is set to 'Save Properties'. The 'Availability Condition' field is empty. At the bottom right are 'Save' and 'Cancel' buttons.

Audit Event. The event type to be recorded in the audit history on the selected object.

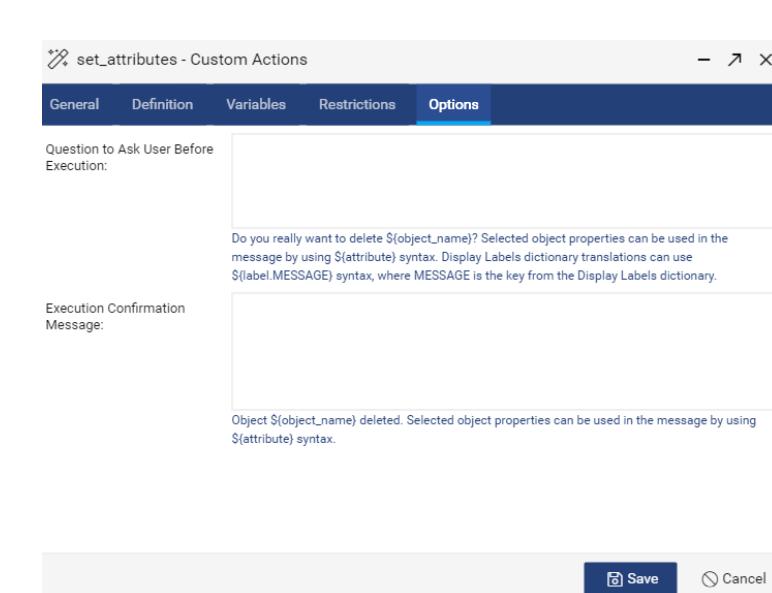
Object Type. The type of object which must be selected.

Required Permission. Permission which the user must have on the selected object.

Availability Condition. A CQL condition, applied to the selected object, which must return as true.

Options Tab

Where a custom action executes an action based on user input, a question can be asked before execution, and a message displayed after execution:



1. Optionally enter a **Question to Ask User Before Execution**. Object properties can be used in the message by using the \${attribute} syntax. Display labels dictionary translations can be used by using the \${label.MESSAGE} syntax, where MESSAGE is the key value from the **_display_labels** dictionary.
2. Optionally enter an **Execution Confirmation Message**. As for Questions to Ask a User Before Execution, object properties can be included in the message using the \${attribute} syntax.

Shortcut to Edit Custom Actions

In addition to editing custom actions from the custom action section of the control panel, a shortcut is available by holding Shift and left clicking an individual custom action from the main user interface **Tools** menu.

6.6. Dashboards

Description

Dashboards group together reports and make them available to users. Each report becomes a tab within the dashboard. Dashboards perform a similar function to workspaces for views.

The screenshot shows the 'Studentx Dashboard' interface. At the top, there are three tabs: 'Authors Report', 'Status Report', and 'Author Report v06'. Below the tabs is a 'Report Filter' section with a dropdown for 'Document Author' set to 'Barry Prince'. On the right, there is a 'Run Report' button. The main area is divided into two sections: 'Results' and 'Preview'. The 'Results' section contains a table with columns: Document Name, Author, Created, and Zone. The table lists various documents such as 'test.doc', 'Test Document 1205a Import Test', 'Test Email 2010b', 'Test Email Import 1205b', 'Test Student14', 'TR Document 1910b', 'TR Document 1910c', '11101a', 'TR012a Lorem ipsum DocumentH', '11101b', and '11104d'. The 'Zone' column shows categories like 'Regulatory'. The 'Preview' section on the right displays a small atomic model icon and some sample text from a document.

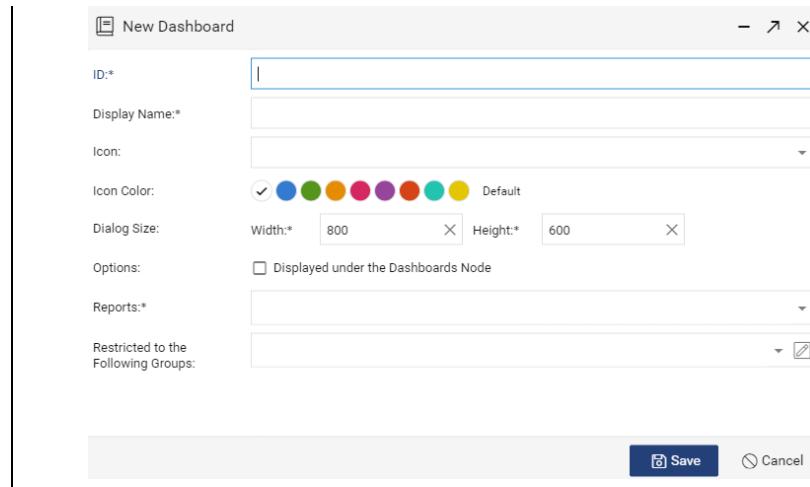
Users can have access to multiple dashboards and one report can be made available in multiple dashboards.

Creating Dashboards

1. Within the CARA Control Panel select **Dashboards** from the **General** section.

The screenshot shows the 'Dashboards' list in the CARA Control Panel. At the top, there is a search bar and a toolbar with buttons for '+ Add', 'Edit', 'Import', 'Export', and 'Delete'. Below the toolbar is a table with columns: ID, Display Name, Version, and Updated. Two dashboards are listed: 'studentx_dashboard' (Version 4, Updated 1/19/22 3:27 PM) and 'test_dashboard' (Version 3, Updated 10/21/21 8:18 PM).

2. Click **+ Add** to open the **New Dashboard** window:



3. Enter a unique internal **ID**.
4. Enter a user friendly **Description**, this will be displayed in the dashboard window header.
5. Select an **Icon** and icon colour.
6. In the **Dialog Size** fields specify the default window **Width** and **Height** in pixels.
7. **Options**. Tick **Display in Toolbar** to have the dashboard available as a toolbar button. Please remember that the dashboard will need to be added to a custom action and then the custom action added to the Type Configuration > View > Actions.
8. In the **Reports** panel, click the right-hand drop-down arrow to open the **Select Reports** window and choose which reports will be available within the dashboard.
9. **Save** the report selection, returning to the **New Dashboard** window.
10. In the **Restricted to the Following Groups** panel, select the user groups the dashboard will be available to. If no groups are selected the dashboard will be potentially available to all users.

However please note the dashboard must be added to a custom action, and custom actions themselves can be restricted to certain groups of users.

11. **Save** the dashboard.

6.7. eForms

Description

eForms provide structured document content.

Document types typically have attributes defined and then a dedicated form is created and presented to users for displaying and entering those attributes. Most documents also have an unstructured content file such as MS Word, however an eform is an alternative to the unstructured content file, an eform can be provided for users to fill out additional attributes which are then merged into an MS Word template and rendered as a PDF file.

eForms can be particularly useful when organisations have large numbers of different forms which users need to work with but which share core attributes. Instead of defining a new type for every form, a base type with core properties is defined and then additionally eForms are defined, each with unique properties and layout for specific business cases.

ID ↑	Display Name	Type	Multi-V...
product	Product	String	
quantity	Quantity	Integer	

Users can select **From eForm** as the **Creation Method** when creating new documents:

The screenshot shows the 'Create' window for eForm creation. It includes fields for 'Classification' (Zone: Regulatory, Section: General), 'Creation Method' (Create: From eForm, eForm: Training eForm (tr_efortm1)), and a 'Next' button.

The creation of eForms share much with the creation of document property forms and use the same form builder tools.

eForms can be restricted to specific types and groups of users.

Creating eForms

1. Within the CARA Control Panel select **eForms** from the **General** section.
2. The **eForms** panel opens showing any previously created eForms, along with their **Description**, **Version**, when they were last **Updated** and who they were last **Updated by**.

ID ↑	Display Name	Version	Updated
tr_efortm1	Training eForm	2	10/19/21 11:55 AM

3. Click **+ Add**, the **New eForm** window opens:

New eForm

General Restrictions

ID:*

Display Name:*

Search

ID ↑	Display Name	Type	Multi-V...

+ Add + Add Sub-Attribute Edit Delete Edit Variables Form

Template:*

MS Word template file that will be rendered to a PDF with the eForm values

Upload Export

Save Cancel

4. Enter a unique **ID**.
5. Enter a **Display Name**.
6. For each **Variable** to be added to the eForm click **+ Add** to open the **New Variable** window:

New Variable

Name:*

product_code

Display Name:*

Product Code

Data Type:*

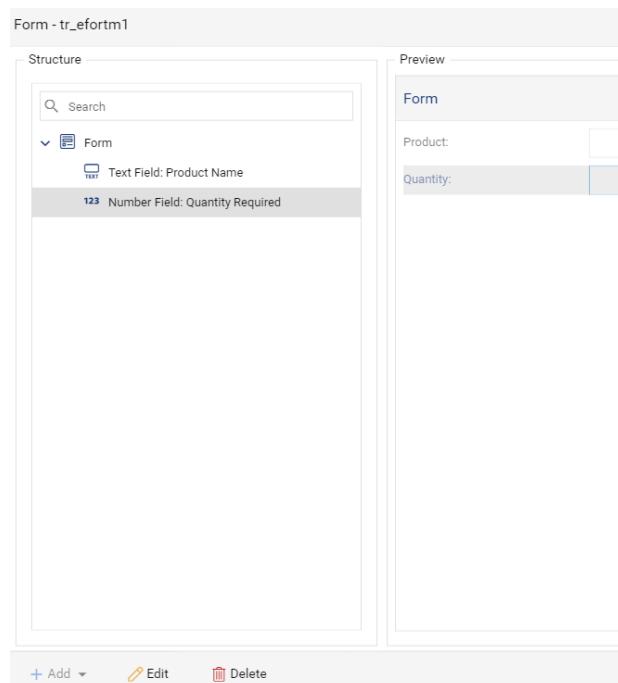
String

Multi-Value

Save Cancel

7. Enter a **Name**.
8. Enter a **Display Name**.
9. Select a **Data Type**.

10. Variables are single value by default, alternatively tick **Multi-value** if the variable should accept multiple values.
11. Optionally enter a **Source Object Property to Copy as Default**.
12. **Save** the variable, returning to the **New eForm** window.
13. Once all variables have been defined click **Edit Variables Form** to launch the form builder window:

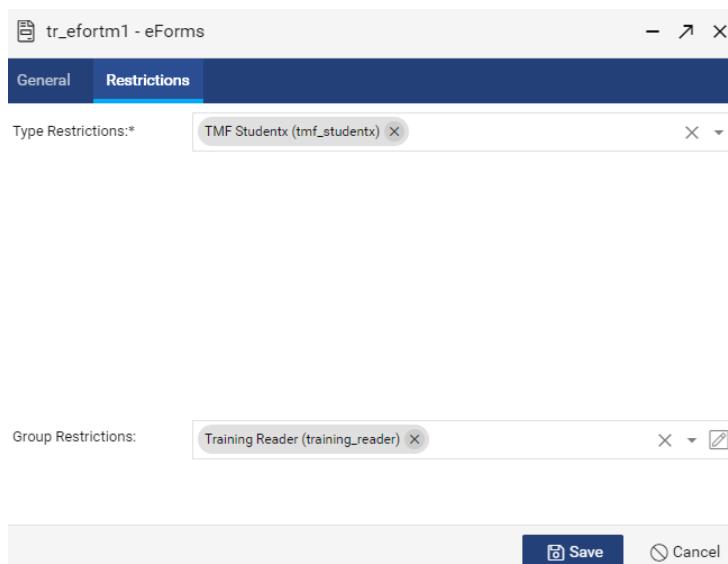


14. When creating the eform layout the form builder window has the same design options available as when creating document property forms. Please see the Forms chapter for details.
15. On the form builder window add the variables defined in the previous steps. Please note that only the variables defined specifically for the eform can be added. Other attributes such as those defined for object types are not available to be added as input fields.

16. **Save** the form once completed, returning to the **New eForm** window.
17. **Template.** Click **Upload** and locate the MS Word document the eform will be merged with when new documents are created. The MS Word document must contain merge fields which match the Name/ID of the variables above.



18. In the **Restrictions** tab, within the **Type Restrictions** panel select which types the eform will be available for:



19. In the **Restrictions** tab, within the **Group Restrictions** panel select which groups of users the eform will be available for when creating new documents. If left blank the eform will be available to all users, however users require the **Type Capability: Create From eForm** in order to create documents from any eForm.

20. **Save** the eform.

6.8. Lifecycles

Description	<p>A lifecycle describes a series of states which a document may pass through such as draft, reviewed, approved. Documents can be both promoted forward and demoted back to previous states as required. The progression of states within a CARA lifecycle does not need to be linear.</p> <p>Each state of a lifecycle can modify properties of the document attached to it, for example the security might be updated so that a document becomes read only when it enters an approved state. Attributes are typically updated each time a document enters a new state, for example the status attribute being set to Approved or Obsolete or Superseded etc.</p> <p>In CARA lifecycles can be applied to documents automatically at the time they are created, and moved between lifecycle states either manually by users or automatically as part of workflow.</p> <p>Lifecycles are created and listed within the Lifecycles window within the Control Panel, which displays their ID, Description, Version, date Last Updated and who the lifecycle was last Updated By.</p>
--------------------	--

Search				
+ Add Edit Import Export Delete				
ID ↑	Display Name	Version		
studentnb_lifecycle	Studentnb Lifecycle	1	12/5	
studentx_lifecycle	Studentx Lifecycle	4	9/20	

Creating Lifecycles

1. Within the CARA Control Panel select **Lifecycles** from the **General** panel.
2. Click **Add**, the **New Lifecycle** window opens:

The screenshot shows the 'New Lifecycle' dialog box. At the top, there are fields for 'ID*' and 'Display Name*'. Below these is a checkbox for 'Add State Name as a Version Label'. The interface has two tabs: 'States' (which is selected) and 'Transitions'. The 'States' tab contains a table with columns 'State Name', 'Actions?', and 'Notifications'. At the bottom of the dialog are buttons for '+ Add', 'Edit', and 'Remove', along with 'Save' and 'Cancel' buttons.

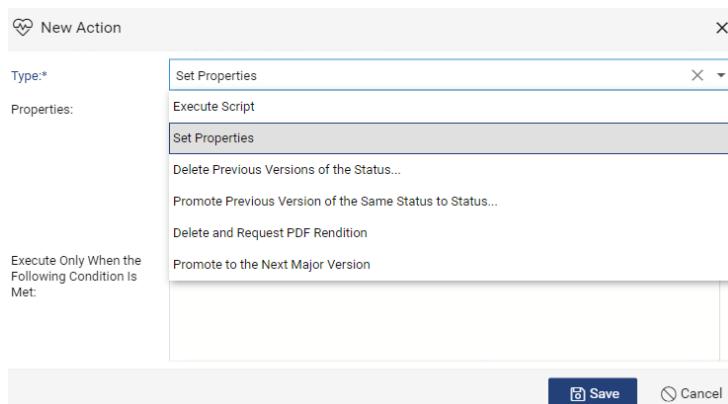
3. Enter a unique **ID**.
4. Enter a **Display Name**.
5. Optionally tick **Add State Name as a Version Label** to have the version label attribute automatically updated with the lifecycle state.
6. Click **Add** within the **States** tab to open the **New Lifecycle State** window:

The screenshot shows the 'New Lifecycle State' dialog box. It has a 'Name:' field and an 'Options:' section with checkboxes for 'Attachable' and 'Reset to Base on Versioning'. The interface has two tabs: 'Actions' (selected) and 'Notifications'. The 'Actions' tab contains a table with columns 'Type' and 'Parameters'. At the bottom of the dialog are buttons for '+ Add', 'Edit', and 'Remove', along with 'Save' and 'Cancel' buttons.

7. Enter a **Name** for the state.

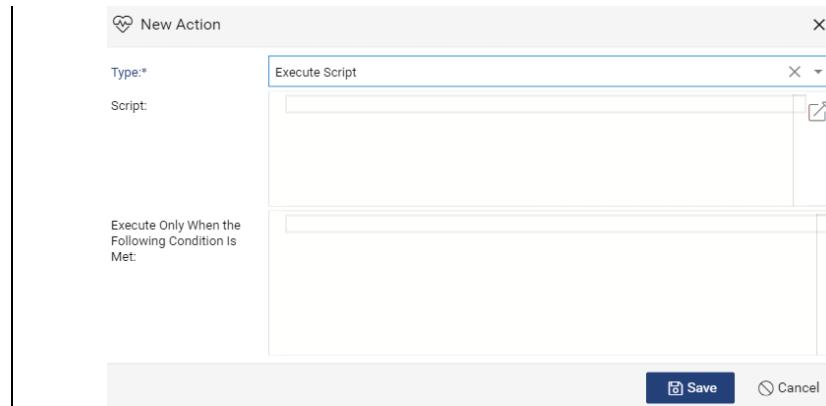
8. Select if the state will be **Attachable**, that is can a document start in this state when attached to the lifecycle. Typically a document would begin in a draft state, which would therefore be attachable. However it may be appropriate for other states to also be attachable, for example documents which are imported may have already been approved and so such documents would be placed straight into the approved state, without having to enter or pass through other earlier states first.
9. Select if the state should force the document to **Reset to Base on Versioning**. For example, if an approved document is versioned, the new version will not have passed through an approval process, it will be a new draft version and so should automatically be placed back in the base draft state.

10. Click **Add** to open the **New Action** window:

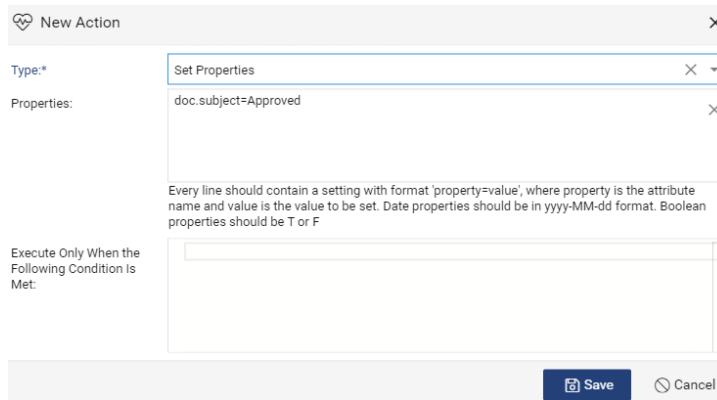


11. Select an action **Type** from the drop-down list:

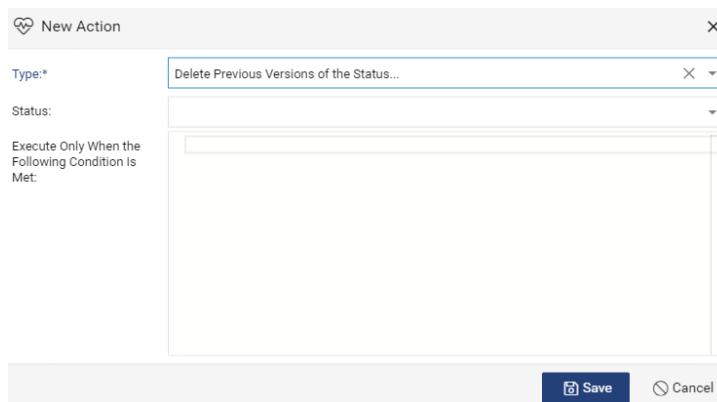
- a. **Execute Script.** Enter the **Script** and optionally a condition that is required to be met before the script can be executed:



- b. **Set Properties.** Set document properties. Multiple properties can be set, with each listed on a new line. Strings do not need to be enclosed in quotes. Type specific properties are prefixed with **doc.** as in the example below:

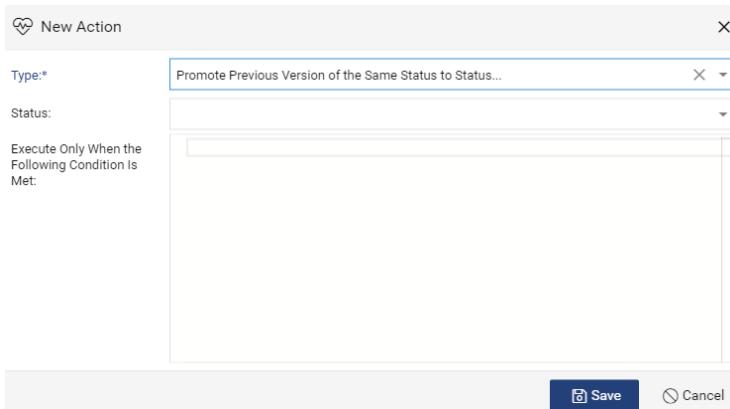


- c. **Delete Previous Versions of the Status:**



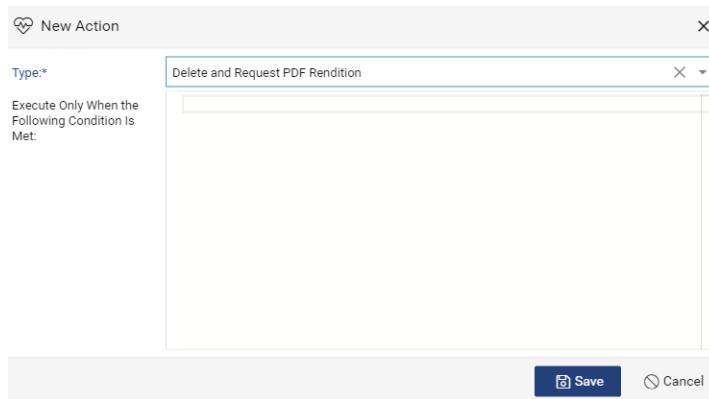
Select or manually enter the **Status** of documents which will be deleted. For example if draft versions do not need to be retained, all draft versions could be deleted when the document enters an approved state. This action type can have a condition specified prior to execution.

d. Promote Previous Versions of the Same Status to Status:



Select or manually enter the **Status** of documents which will be promoted. For example all previously approved versions might be considered superseded when the current version is approved. This action type can have a condition specified prior to execution.

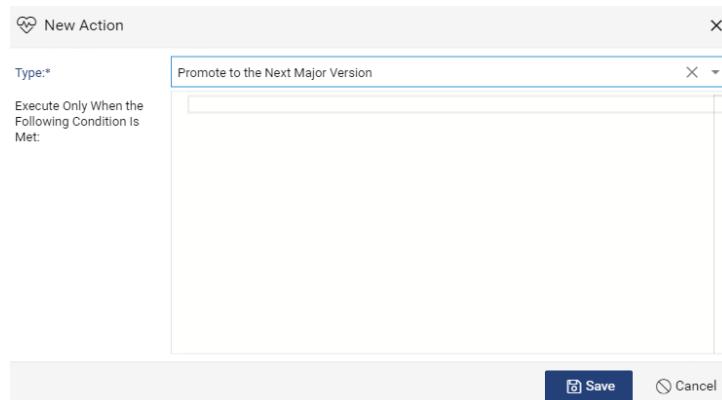
e. Delete and Request PDF Rendition:



This can be useful in scenarios such as if documents which enter a superseded state do not require the retention of a source

document because they will no longer be eligible for editing, but a rendition is to be kept for historical reference. This action type can have a condition specified prior to execution.

f. **Promote to the Next Major Version:**



Promote the document to the next major version, for example draft version 1.4 becomes version 2.0 when it moves to the approved state. Note the version being promoted becomes the next major version, rather than an additional version being created, in the example above there would no longer be a version 1.4.

12. Configuring the **Notifications** tab allows selected groups of users to be automatically notified by email when a document enters the specific lifecycle state.
13. Select a **Notifications Template** from the drop-down list.
14. Select **Restricted Accessors**, that is the individuals or group of users who will receive the notifications.
15. Click **Save** and return to the **New Lifecycle** window.
16. **Add** further states and their actions and notifications as needed.

17. Once all states have been defined click the **Transitions** tab and click **Add** to open the **New Transition** window:

The screenshot shows the 'New Transition' dialog box. At the top left is a heart icon and the text 'New Transition'. At the top right is a close button ('X'). Below the title are four dropdown menus: 'Label*', 'Source State*', 'Target State*', and 'Groups Allowed to Perform the Transition'. Under 'Options', there are two checkboxes: 'Allow Manual Transition' and 'PDF Rendition Required'. Below these is a large text area for 'Validation Script'. At the bottom are 'Save' and 'Cancel' buttons.

18. Enter a **Label** that describes the transition between states, for example "Promote to Reviewed".

19. Select a **Source State** from the drop-down list.

20. Select a **Target State** from the drop-down list.

21. **Version Before State Change.** If appropriate, the document can be automatically versioned before it enters the target state.

22. **Allow Manual Transition.** Choose if users can be allowed to select when the document transitions between the specified states.

23. **PDF Rendition Required.** If selected, documents will only be allowed to transition between the specified states if a PDF rendition is present.

24. **Groups Allowed to Perform the Transition.** If manual transitions have been allowed, select which groups of users are allowed to perform the transition.
25. **Validation Script.** A validation script can be entered which must return as true before the transition is allowed.
26. Click **Save** once complete and return to the **New Lifecycle** window.
27. Create further transitions as appropriate. There will typically be transitions which promote a document forward and others which demote to previous states. Transitions do not need to be linear, for example it may sometimes be appropriate to demote a document from approved to draft or promote from draft straight to approved.
28. When all states and transitions have been defined click **Save** to return to the **Lifecycles** window.

6.9. Notification Channels

Description

Notification channels are used to pass notifications to users, such as when a document has been updated. Notification templates are configured for each event and then a notification channel is selected.

The most common notification channel is Mail, where users receive an external email via their email client such as MS Outlook. Multiple notification channels can be configured of each channel type, for example if 2 or more SMTP servers are available or a different sender name is to be used. Each notification template is tied to a specific notification channel instance.

Creating Mail Notification Channels

1. Within the CARA Control Panel select **Notification Channels** from the **General** panel.
2. Click **Add**, the **New Notification Channel** window opens.
3. Select **Mail** from the **Channel** drop-down list.

The screenshot shows the 'New Notification Channel' configuration window for a Mail notification channel. The window has a title bar 'New Notification Channel' with standard window controls. It contains several input fields and dropdown menus:

- Channel:** A dropdown menu set to 'Mail'.
- Display Name:** An empty text input field.
- SMTP** section:
 - Host:** An empty text input field.
 - Encryption:** Radio buttons for 'None' (selected), 'SSL', and 'TLS'.
 - Port:** A dropdown menu set to '25'.
 - User Name:** An empty text input field.
 - Password:** An empty text input field.
 - Sender Name:** An empty text input field.
- Send Emails on Behalf of the Original User:** A checkbox with a note: 'The mail server needs to be configured to allow.' (unchecked).
- URL** section:
 - Application URL:** An empty text input field with a note: 'Full URL of the application that will be used in emails as a base link.'
- Action Buttons:** 'Save' and 'Cancel' buttons at the bottom right.

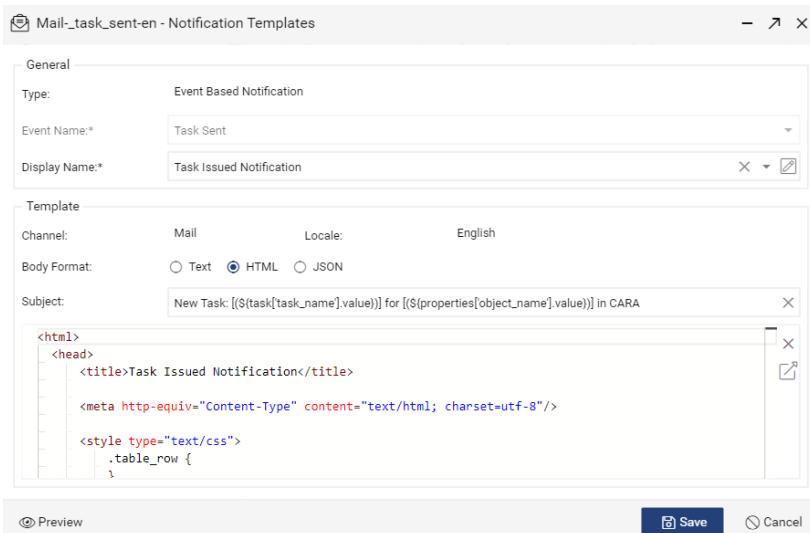
4. Enter the **Host** name of the **SMTP** email sever which will be used to send the external emails.
5. Select the **Encryption** method for the connection between CARA and the SMTP server, either **SSL** or **TLS**. If the connection is not encrypted select **None**.
6. Enter the **Port** to be used for the connection. By default:
Non-encrypted SMTP uses port 25.
SSL SMTP uses port 465.
TLS SMTP uses port 587.
7. Enter the **User Name** that CARA will use for the connection.
8. Enter the **Password** that CARA will use for the connection.
9. Enter the **Sender Name** for the notification emails sent by CARA. Alternatively tick **Send email on Behalf of the Original User**, however the mail server will need to be configured to allow for this.
10. Enter the **Application URL**, that is the full URL of the CARA application that will be used in emails as a base link. This is organisation specific and provides the functionality where a link to the related document is included within the body of the email notification.
11. **Save** the mail notification channel settings.

6.10. Notification Templates

Description

Notification templates are the basis of email and other notifications, which can be configured to be sent to users automatically based on default events, such as a document being created or versioned, or custom events.

Notification templates primarily consist of a subject line and body of text created in either a plain text, HTML or JSON format. Variables such as the user name or the document name can be included.



Notification templates can be created for a specific locale, so that users receive notifications in their appropriate language.

Different channels can be used to send the notification to the user. The most common type of channel is email, however others such as internal message and Slack notification are also available. The content from a single notification can be copied and pasted for reuse with multiple channels.

The **Notification Templates** window in the CARA Control Panel lists template **Type**, **Event/Name**, **Display Name**, **Channel**, **Locale**, **Version** and **Updated** date:

Type	Event/Name ↑	Display Name	Channel
EventBased	Job Completed	Job completed	Mail
EventBased	State Change	State Change Notification	Mail
EventBased	Task Delegated	Task Delegated Notification	Mail
EventBased	Task Reassigned	Task Reassigned Notification	Mail
EventBased	Task Sent	Task Issued Notification	Mail
EventBased	Workflow Cancelled	Workflow Cancelled Notification	Mail
EventBased	Workflow Completed	Workflow Completed Notification	Mail
EventBased	Workflow Overdue	Workflow Overdue Notification	Mail
EventBased	Workflow Step Completed	Workflow Step Completed	Mail

Creating Notification Templates

1. Within the CARA Control Panel select **Notification Templates** from the **General** panel.
2. Click **+Add** to open the **New Template** window:

3. Select the notification template **Type**:

Event Based Notifications are those triggered by the standard events, such as document creation or workflow completion, which are listed in the `_notification_events` dictionary. Only events which are marked as active in the dictionary are listed. For event based notifications select the trigger **Event Name** from the drop down list.

Custom Notifications are those triggered by events other than the standard events, for example custom date based notifications can be created per type. Please see the Date Based Notifications chapter for further details. For custom notifications enter a **Name** for the template.

4. **Display Name.** Enter a user friendly display name.
5. **Channel.** Select a channel from the drop-down list. The available channels are **Internal**, **Mail** and **Slack**. Only configured channels are selectable.
6. **Locale.** Select the locale the template will be associated with from the drop-down list.
7. **Body Format.** Select whether the notification's main body format will be **Text**, **HTML** or **JSON**. Please note that Slack does not recognise HTML.
8. **Subject.** Enter the notification subject. The subject can include variables.
9. **Main Body.** Enter the notification's main body in the format selected above.
10. **Save** the notification template, returning to the notification templates window.

Options to edit, delete, view the audit trail and view the history of notification templates are available from the right-click menu of the Notification Templates window:

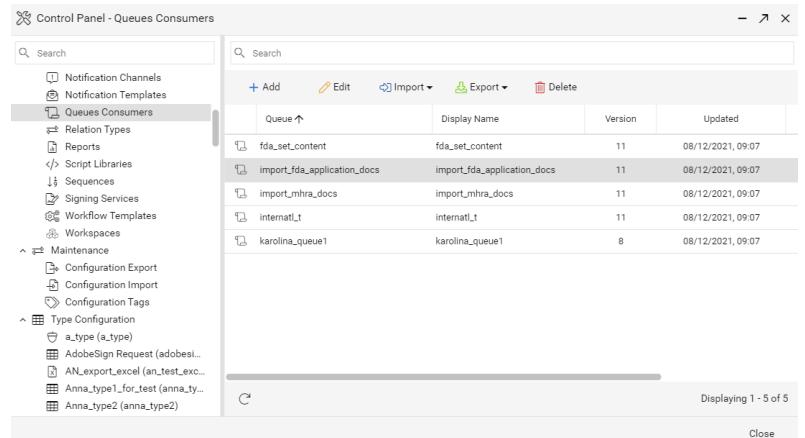
<input type="text"/> Search				
+ Add Edit Import Export Delete				
Type	Event/Name ↑	Display Name	Channel	
EventBased	Job Completed	Job completed	Mail	
	+ Add	State Change	State Change Notification	Mail
	 Edit	Task Delegated	Task Delegated Notification	Mail
	 Delete	Task Reassigned	Task Reassigned Notification	Mail
	 AuditTrail Report	Task Sent	Task Issued Notification	Mail
	 History	Workflow Cancelled	Workflow Cancelled Notification	Mail
	EventBased	Workflow Completed	Workflow Completed Notification	Mail
	EventBased	Workflow Overdue	Workflow Overdue Notification	Mail
	EventBased	Workflow Step Completed	Workflow Step Completed	Mail

Notification templates which are in use can be edited but not deleted.

6.11. Queue Consumers

Description

Queue consumers allows scripts to be run against objects in a custom queue.

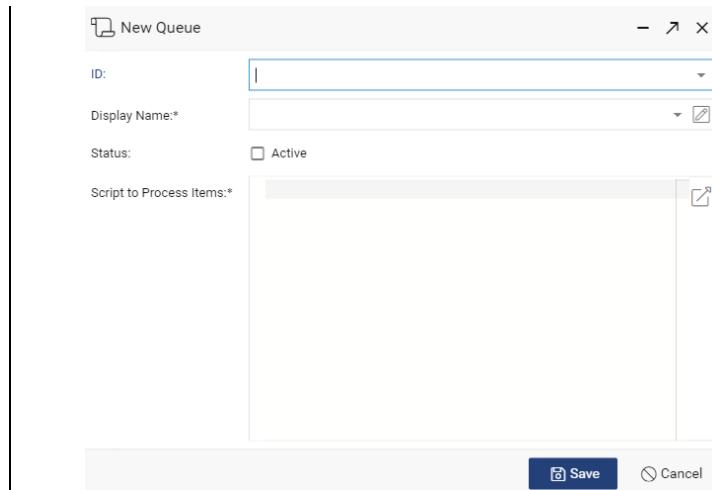


A custom queue is created first and then type-specific queuing configuration is created which determines which events result in an object being placed in the queue, such as object creation or versioning. See the separate chapters of this manual for details of creating custom queues and type-specific queuing configuration.

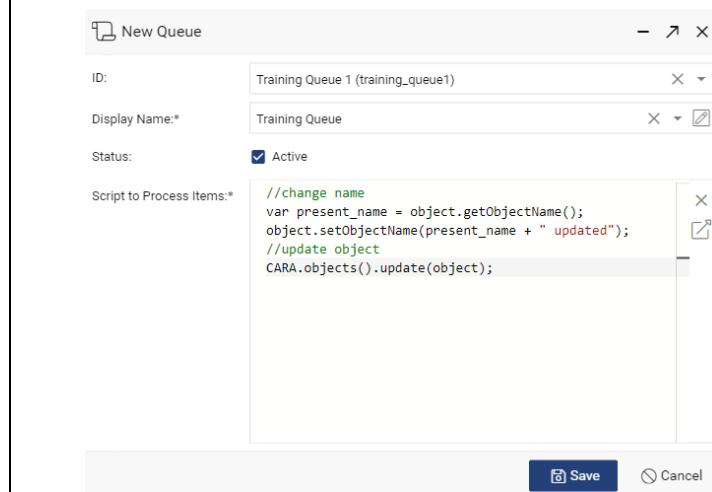
The queue consumer specifies the script that is to be run against objects in the queue. The name of the queue consumer must match the name of the custom queue. Objects within a queue are processed in series in date and time order.

Creating a Queue Consumer

1. Within the CARA Control Panel select **Queue Consumers** from the **General** section.
2. In the Queue Consumers panel select **New**, the New Queue window opens:

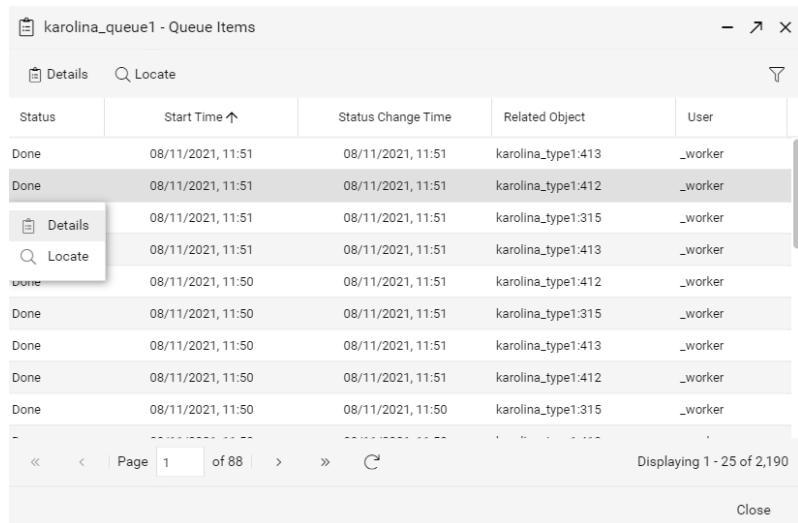


3. Enter or select the **ID**, which must match the name of the queue the script will be associated with.
4. Enter a **Display Name**.
5. Select **Status** as Active. If left as inactive, the script will not be run against items in the queue.
6. Enter the **Script to Process Items**. See the CARA Scripts guide for details and examples of scripting within CARA.



7. **Save** the new queue consumer.

8. The new queue consumer is listed in the Queue Consumers panel.
9. The items that have been sent to the queue and the status of those items can be viewed by highlighting a queue in the Queues panel and selecting Show Queue Items:



Status	Start Time ↑	Status Change Time	Related Object	User
Done	08/11/2021, 11:51	08/11/2021, 11:51	karolina_type1:413	_worker
Done	08/11/2021, 11:51	08/11/2021, 11:51	karolina_type1:412	_worker
Details	08/11/2021, 11:51	08/11/2021, 11:51	karolina_type1:315	_worker
Locate	08/11/2021, 11:51	08/11/2021, 11:51	karolina_type1:413	_worker
Done	08/11/2021, 11:50	08/11/2021, 11:51	karolina_type1:412	_worker
Done	08/11/2021, 11:50	08/11/2021, 11:51	karolina_type1:315	_worker
Done	08/11/2021, 11:50	08/11/2021, 11:51	karolina_type1:413	_worker
Done	08/11/2021, 11:50	08/11/2021, 11:51	karolina_type1:412	_worker
Done	08/11/2021, 11:50	08/11/2021, 11:50	karolina_type1:315	_worker

Displaying 1 - 25 of 2,190

Close

6.12. Relation Types

Description

Users have the option to add relationships between documents. Relationships provide a quick and convenient method of connecting and locating documents:

Parent:			
	Name	Created	Version label
	tr2101g	1/21/22 10:42 AM	0.1, LATEST, Draft

Related:			
	Name	Created	Version label
	Training Document 2010a	10/20/21 1:42 PM	1.0, LATEST, Approved

... Select Remove

Relation Type:*

Comment:

Bind To:*

Selected Version Latest Version Version with Label...

Create Cancel

The **Relations** widget details related documents for the currently highlighted document:

Relations			
	Type	Comment	Name
	Work Instruction	SOP related work instruction	tr2101g

There is also a separate **Relationship Details** window:

Relationship Details

Parent: tr2101g

Relation Type: SOP

Comment: SOP related work instruction

Related: Training Document 2010a

Bind To: Selected Version Latest Version Version with Label...

Save **Cancel**

Administrators define the relationships which users can place documents into, including their description, the version of documents to be related documents and any restrictions.

SOP - Relation Types

General Security & Restrictions Validation Scripts

Integrity Kind: Allow Deletion of Related Objects (AllowDelete)

Parent Type: TMF Studentx (tmf_studentx)

Parent Condition:

Child Type: TMF Studentx (tmf_studentx)

Child Condition:

Restricted Accessors:

Max Relations Count:

Save **Cancel**

Creating Relations

- Within the CARA Control Panel select **Relations** from the **General** section, the **Relation Types** panel opens:

Search				
Add	Edit	Import	Export	Delete
Channel ↑	Display Name	Version	Updated	
	SOP	SOP	7	1/31/22 4:31 PM

2. Click **+ Add**, the **New Relation Type** window opens, which is divided into 3 tabs:

General Tab:

3. **Status.** Only **Active** relation types are selectable users.
4. **ID.** ID is a required field and can only contain letters, numbers, hyphens and underscores. It must start with a letter.
5. **Relation Name.** The relation name describes the relationship from parent to child document. Either enter a value directly into the field or select one from the `_display_labels` dictionary.
6. **Converse Name.** The converse name describes the relationship from child to parent document. Relation Name and Converse Name can be the same, for example “Of Interest”. Either enter a value directly into the field or select one from the `_display_labels` dictionary.
7. **Icon and Icon Color.** Select an icon and an icon colour.

Security and Restrictions Tab:

The screenshot shows a configuration dialog titled "New Relation Type". The "Security & Restrictions" tab is active. The form contains the following fields:

- Integrity Kind:*
- Parent Type:*
- Parent Condition:
- Child Type:*
- Child Condition:
- Restricted Accessors:
- Max Relations Count:

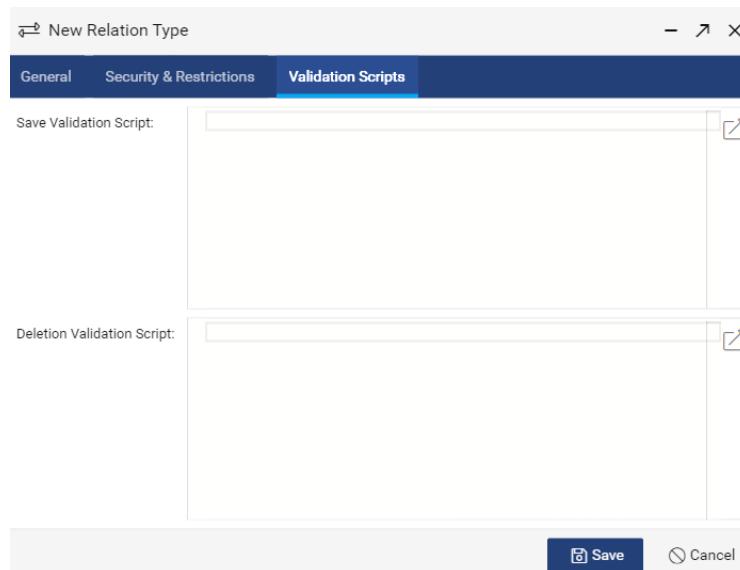
At the bottom right are "Save" and "Cancel" buttons.

8. **Integrity Kind.** Select an integrity kind, either **Allow** or **Restrict** the **Deletion of Related Objects**. Typically a relationship does not change the permissions on the related documents, they remain independent of each other. However the relationship can be used to prevent deletion of the documents within if the **Restrict Deletion** option is selected.
9. **Parent Type.** Select the type of document which can be the parent in the relationship. Please note that where the documents are considered to be in a peer to peer relationship, a Child type and Parent type must still be selected.
10. **Parent Condition.** Optionally add a DQL condition which must be met for the relationship to be available for the selected type.
11. **Child Type.** Select the type of document which can be the child in the relationship.
12. **Parent Condition.** Optionally add a DQL condition which must be met for the relationship to be available for the selected type.

13. **Restricted Accessors.** Select users and/or groups allowed to add the relationship to documents. If left blank, all users are allowed.

14. **Max Relations Count.** A single document can by default have multiple relationships. Optionally select a maximum number.

Validation Scripts Tab:



15. **Save Validation Script:** Optionally enter a validation script which must return as true before the relationship can be saved.

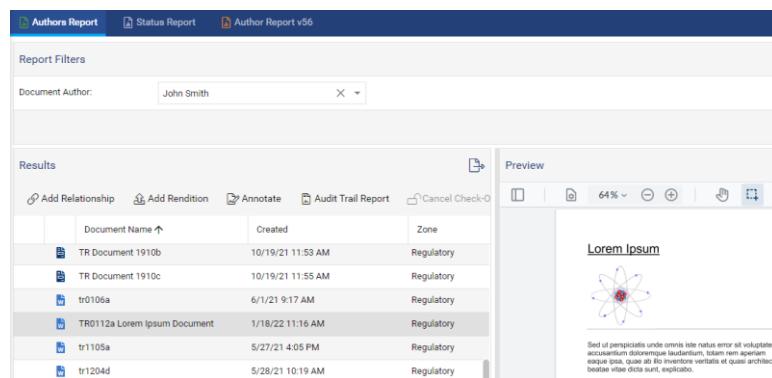
16. **Deletion Validation Script.** Optionally enter a validation script which must return as true before the relationship can be deleted.

17. **Save** the new relationship type.

6.13. Reports

Description Reports return detailed information about documents and other objects. Reports are based on a query or script and can include search filters on a custom form. Reports are not limited to only returning a simple list of results, they can also include full document management functions with a dedicated toolbar, right-click menu and document preview.

Example report with a custom toolbar, search filter and preview panel:



Reports are grouped together into a dashboard and dashboards are accessed by users from a dedicated main menu or as a custom toolbar button, see the relevant chapters on dashboards and custom actions for details.

Creating Reports

1. Within the CARA Control Panel select **Reports** from the **General** panel, the **Reports** panel opens:

The screenshot shows a list of reports in a grid format. The columns are labeled: ID ↑, Display Name, Version, Updated, and Updated By. The data rows are:

ID ↑	Display Name	Version	Updated	Updated By
author_report1	Authors Report	3	1/19/22 4:13 PM	Barry Prince
author_report_v56	Author Report v56	6	1/19/22 3:30 PM	Barry Prince
status_report1	Status Report	3	10/21/21 8:19 PM	Barry Prince

Below the grid, a message says "Displaying 1 - 3 of 3". A "Close" button is at the bottom right.

2. Click **+ Add**, the **New Report** window opens:

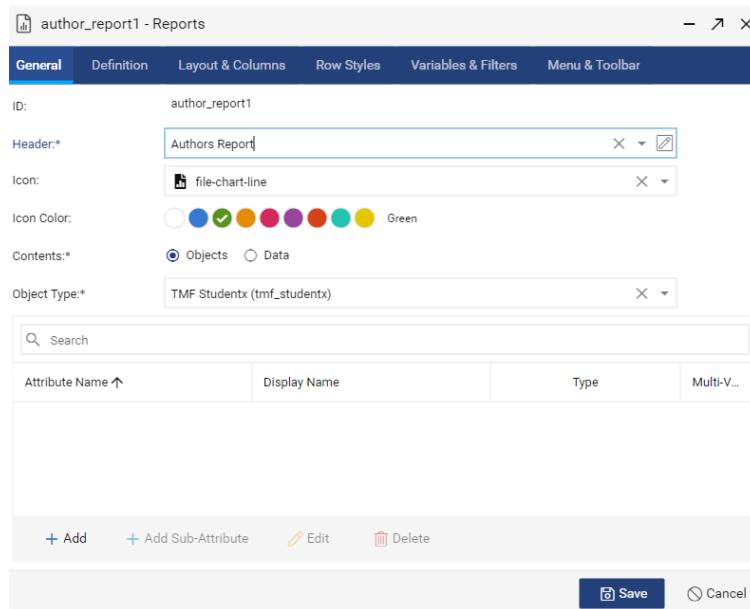
The screenshot shows the "New Report" configuration window. The tabs at the top are General, Definition, Layout & Columns, Row Styles, Variables & Filters, and Menu & Toolbar. The General tab is selected. The fields include:

- ID:*
- Header:*
- Icon: file-chart-line
- Icon Color: (color swatches)
- Contents:*
- Object Type:*

Below these are sections for "Attribute Name ↑" and "Display Name" with a "Type" column. At the bottom are buttons for "+ Add", "+ Add Sub-Attribute", "Edit", "Delete", "Save", and "Cancel".

Report configuration is divided into the following 6 tabs.

General Tab



1. Enter a unique **ID** for the report.
2. Enter a user friendly report **Header** or select one from the **display_labels_** dictionary via the drop-down list.
3. Select an **Icon** and icon colour.
4. Select the **Contents** as either **Objects** or **Data**. The selection here determines the type of information that can be returned and how it is displayed:

If **Objects** is selected, choose an **Object Type** from the subsequent drop-down list. Documents and other objects returned by the report will display format and lock icons and can have a customised right-click menu. Note that it is not necessary to define attributes in the panel beneath when **Content** has been set to **Objects**, attributes to be displayed only need to be defined in the **Layout & Columns** tab.

If **Data** is selected, objects returned by the report will not have format or lock icons and an object type is not selected. Any attributes to be defined as columns within the **Layout & Columns** tab must first be defined within the attributes panel of

the **General** tab. For each attribute click the **+ Add** button to add a **New Attribute**, choosing its **Name**, **Display Name** and **Data Type**:

New Attribute

Name:*

Display Name:*

Data Type:*

Multi-Value

Save **Cancel**

Below is an example of attributes defined for a report where **Content** is set to **Data**, the report returned information about groups and their membership:

New Report

General Definition Layout & Columns Row Styles Variables & Filters Menu & Toolbar

ID:* tr_group_report1

Header*: TR Group Report

Icon: file-chart-line

Icon Color: Default

Contents*: Data

Attribute Name ↑	Display Name	Type	Multi-V...
name	Group Name	String	
user_members	User Members	String	
modified	Modified Date	DateTime	

+ Add **+ Add Sub-Attribute** **Edit** **Delete**

Save **Cancel**

Definition Tab

author_report1 - Reports

General **Definition** Layout & Columns Row Styles Variables & Filters Menu & Toolbar

Source: Query Script

Query:
`select * from tmf_studentx where creator=${var1}`

Report query. May include defined variables as \${var} and conditional statements in [] brackets.

Load Before Filter Selection. Used only when filters form defined.

Save **Cancel**

1. For **Source**, select whether the report is to be based on a CQL **Query** or **Script**.
2. Enter the CQL **Query** or a **Script** in the main panel. Please refer to the CARA Query Language chapter or the Scripts Guide for details of the syntax to be used.
3. If **Load Before Filter Selection** is ticked, the query or script will run immediately when the report is opened, without the user first needing to enter a filter value and click a button to run the report.

Layout & Columns Tab

author_report1 - Reports

General Definition **Layout & Columns** Row Styles Variables & Filters Menu & Toolbar

Display Results As: Grid Tree Grid Enable Preview Panel

Attribute	Label	Alignment	Width	Visible?	Sortable?	Cell Styles	Cell Actions
Name (object_n...	Document Name	Left	200	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Creator (creator)	Author	Left	100	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Created (create...	Created	Left	200	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Zone (doc.zone)	Zone	Left	100	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

+ Add **+ Add multiple** **Edit** **Remove**

Sort By: Ascending Descending

Enable Paging

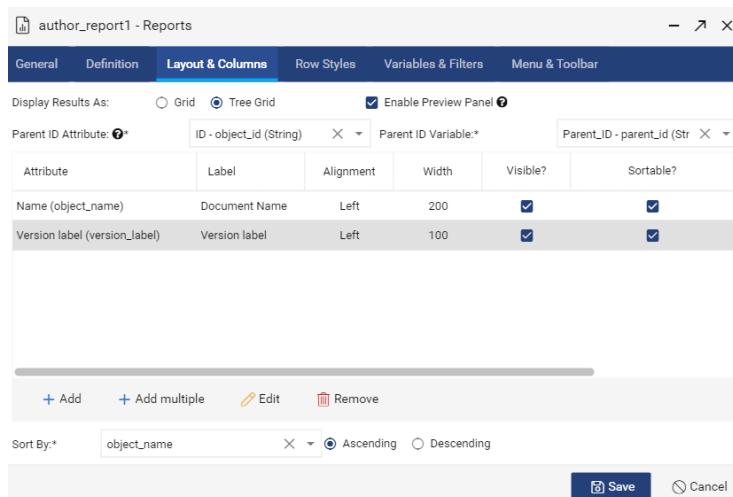
Page Size:

Save **Cancel**

1. Select whether to **Display Results As a Grid** or **Tree Grid**. Reports are typically laid out in a simple grid, where each row displays attribute values for a single document or object.

However reports can be created which display related objects, such as for structures or documents in a relationship, with parent documents on one line and child documents nested on subsequent lines underneath, this is referred to as a tree Grid.

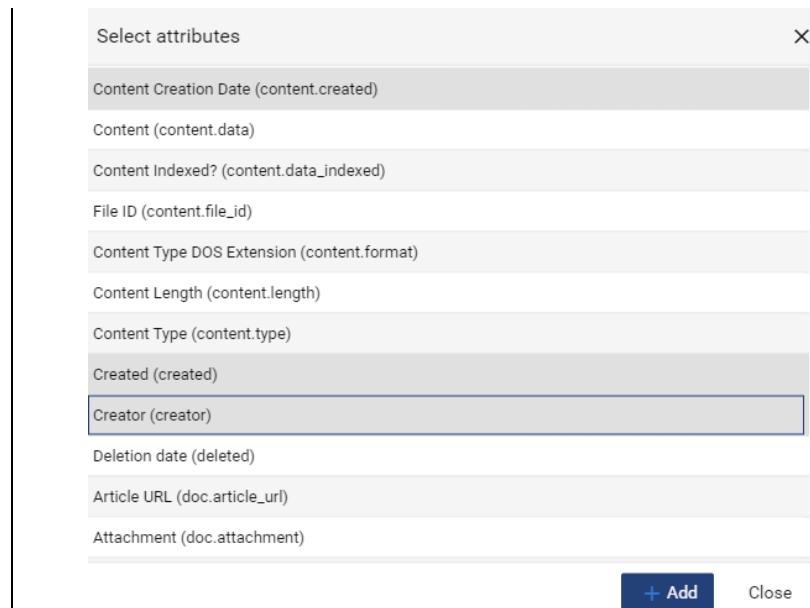
Tree grid reports use a **Parent ID Attribute** and **Parent ID Variable** to identify the parent and child objects in a relationship. The query or script which defines the report must include the parent id variable to identify parent and child objects.



2. **Enable Preview Panel** can be selected, which displays a preview when an object is selected within the report.
3. Click **+ Add** to open the **Add Column** window:

Attribute:*	<input type="text"/>
Label:*	<input type="text"/>
<input checked="" type="checkbox"/> Visible? <input type="checkbox"/> Sortable?	
Width:	<input type="text" value="100"/> <input type="button" value="X"/>
Alignment:	<input checked="" type="radio"/> Left <input type="radio"/> Centre <input type="radio"/> Right
Format: <small>?</small>	<input type="text"/>
Template: <small>?</small>	<input type="text"/>

4. Select an **Attribute** from the drop down list.
5. Choose if the attribute should be **Visible** by default or **Sortable**.
6. Enter a user friendly **Label** or select one from the `display_labels_` dictionary via the drop-down list.
7. Select a **Width** in pixels.
8. Choose an **Alignment** option.
9. Optionally a display **Format** or **Template** can be entered, options for which are detailed in the View > Grid chapter.
10. Custom **Cell Styles** and on-click **Actions** can be configured, as detailed in the View > Grid chapter.
11. **Save** the column configuration, returning to the **Columns** tab.
12. Add further attribute columns as needed.
13. In addition to adding columns individually, multiple attributes can be selected as columns in a single action and then configured individually by using the **Add Multiple** option:



14. Select an attribute to **Sort by**.

Row Styles Tab

author_report1 - Reports

General Definition Layout & Columns Row Styles Variables & Filters Menu & Toolbar

Row Styles:

Condition	Color	Italic	Bold
'lifecycle_state=Draft'	Blue		
'lifecycle_state=Approved'	Green		
'lifecycle_state=Obsolete'	Red		

+ Add Edit Remove

Row Icons:

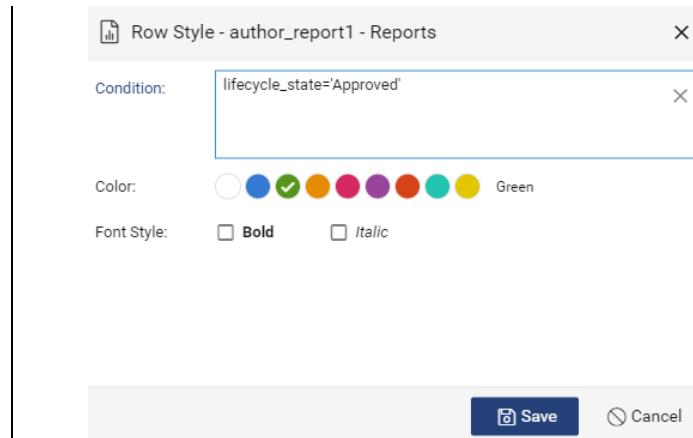
Condition	Icon	Bullet

+ Add Edit Remove

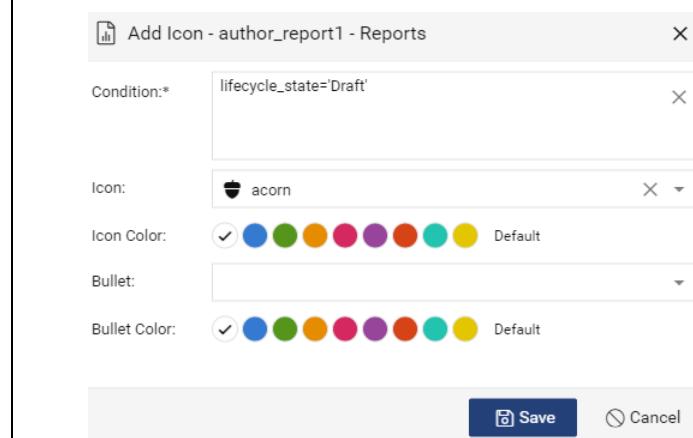
Save Cancel

The report results rows can be colour coded and have custom icons, based on a CQL condition. Note that style of individual cells within the report results can also be customised, via the **Cell Styles** options available when adding report columns in the **Layout and Columns** tab.

1. In the **Row Styles** panel click **+ Add**, enter a CQL format **Condition** in the subsequent window and select a row **Color** and **Font Style**:



2. Click **Save** and then add further row style conditions as needed.
3. In the **Row Icons** panel, click **+ Add** to open the **Add Icon** window. Enter a CQL format **Condition**. Select an **Icon** and icon colour. Optionally select a **Bullet** and bullet colour:



4. Save the icon configuration.

Variables & Filters Tab

ID ↑	Display Name	Type	Multi-V...
var1	var1	String	

Add **Add Sub-Attribute** **Edit** **Delete** **Edit Variables Form**

Custom Apply Filters Button Label:
Run Report

Custom Apply Filters Button Label if Different than "Apply Filters"

Variables Initialization Script:
1

Save **Cancel**

Reports typically allow users to enter report filters, for example document creator, status, product, region etc..

The properties which users can select as report criteria must be defined as variables, included within the report's query definition and presented to the user on a custom variables form.

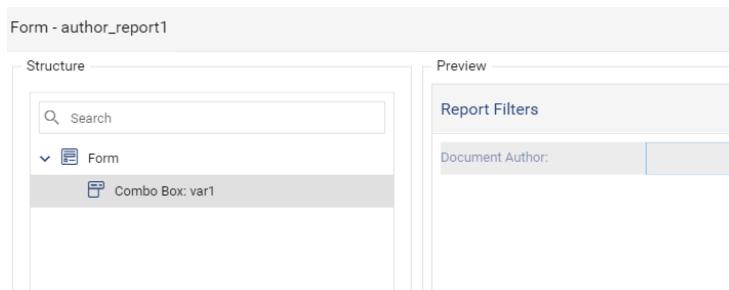
1. Click **+ Add** to open the **New Variable** window:

New Variable

Name:*	author
Display Name:*	Author
Data Type:*	String
<input type="checkbox"/> Multi-Value	

2. Enter a **Name** and **Display Name**. The **Name** must match that used in the **Query** on the **Definition** tab.
3. Select the appropriate **Data Type**.
4. Tick **Multi Value** if multiple values for the attribute can be selected by users.

5. **Save** the new variable, returning to the **Variables & Filters** tab.
Add further variables as required by the report's query.
6. Click **Edit Variables Form** to launch the **Form** builder window.
Place the defined variables on the form, which will be presented to users in the top section of the report window:



The form as displayed within the report:

Document Name ↑	Created
TR Document 1910c	10/19/21 11:55 AM
tr0106a	6/1/21 9:17 AM

The tools available for designing the form are the same as for the main document properties form, please see the **Forms** section of this manual for details.

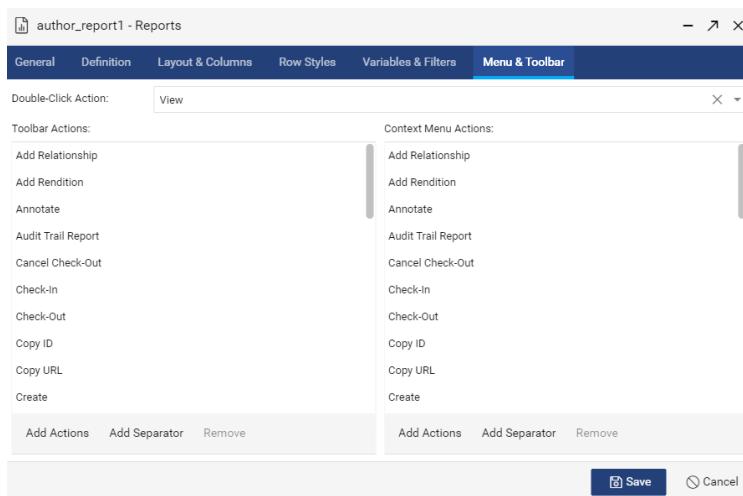
Only the variables defined for the report can be added as attributes to the form.

Save the form once complete, returning to the **Variables & Filters** tab.

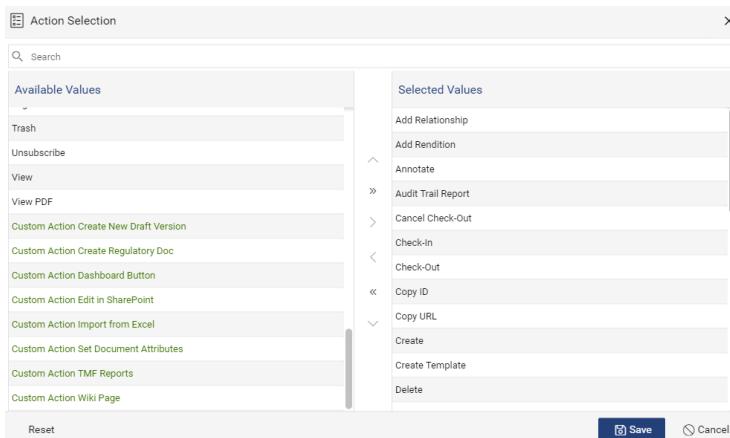
7. Optionally enter a **Custom Apply Filters Button Label**.
8. A **Variables Initialisation Script** can be entered, setting property values for the highlighted document when the report is run.

Menu & Toolbar

Tab



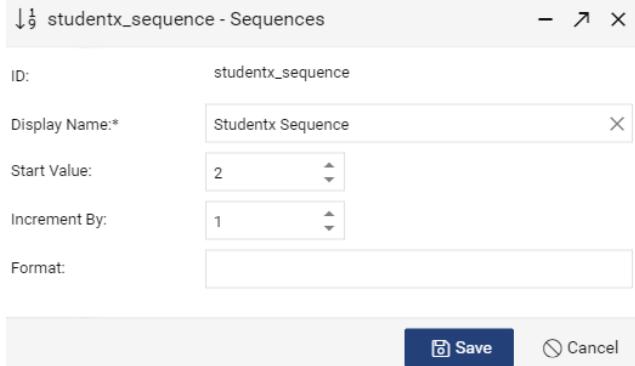
1. Select a **Double-click Action** from the drop-down list.
2. Select **Toolbar Actions** and right-click **Context Menu Actions** from the **Action Selection** window. Actions can be re-ordered by click and drag. Separators can be added to group related actions together. Custom actions can be included and are grouped together and listed in green at the bottom of the **Available Values** panel:



3. Click **Save** once completed.

4. In order for reports to be available to users, they are added to a dashboard. Please see the **Dashboards** chapters for details of creating dashboards.

6.14. Sequences

Description	<p>A sequence is an automatically incremented numeric value that is used within auto-value rules to uniquely number documents.</p> <p>Multiple sequences can be created, so that different types of document or object can have their own sequence, for example there could be separate sequences for clinical and quality documents. This can help to track and compare the number of documents being created for specific types or departments.</p> <p>When a sequence is used within an auto-value rule the sequence value must be stored in an integer attribute, this can be a custom type integer attribute (prefixed doc.) or the core attribute <code>sequence_no</code>:</p> 
Creating Sequences	<ol style="list-style-type: none">1. Within the CARA Control Panel select Sequences from the General section, the Sequences panel opens:

ID ↑	Display Name	Version	Updated
↓‡ sequence2	Clinical Document Sequence	2	2/1/22 9:11 AM
↓‡ studentnb_sequence	Studentnb Sequence	1	12/9/21 2:31 PM
↓‡ studentx_sequence	Studentx Sequence	5	2/1/22 9:10 AM

2. Click **+ Add**, the **New Sequence** window opens:

The dialog box has the title "↓‡ New Sequence". It contains the following fields:
 - ID*: A text input field.
 - Display Name*: A text input field.
 - Start Value: A numeric input field with up and down arrows.
 - Increment By: A numeric input field with up and down arrows.
 - Format: A text input field.
 - Buttons at the bottom: "Save" (blue button) and "Cancel".

3. Enter a unique **ID** for the sequence.
4. Enter a **Display Name**.
5. Enter the **Start Value**. If an existing sequence is to be used, where values have already been incremented, the start value can be set accordingly.
6. Enter the **Increment by** value. Sequences typically increment by 1, but can increment by any chosen numeric value.
7. Enter a **Format**, that is how many digits the increment value will be displayed as, for example if the start value of 1 is to be displayed as 0001, enter 0000.
8. **Save** the New Sequence, returning to the main **Sequence** window.

Sequences which are in use by an auto-value rule cannot be deleted. A warning message is displayed if trying to delete a sequence which is in use. The warning message includes the name of the auto-value rule:

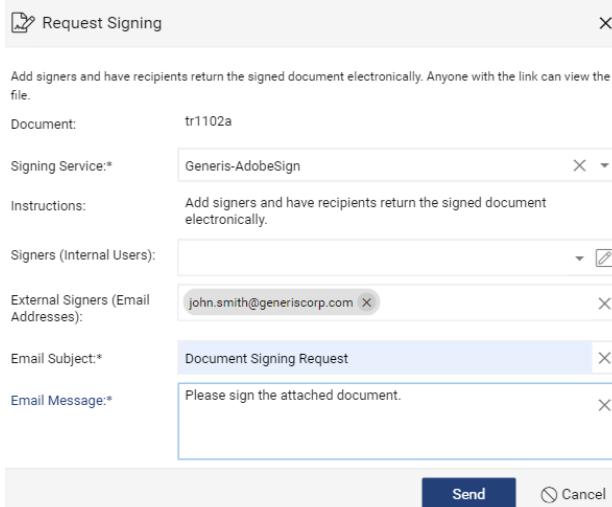
Warning

Sequence can not be removed as it used in: [tmf_studentx]

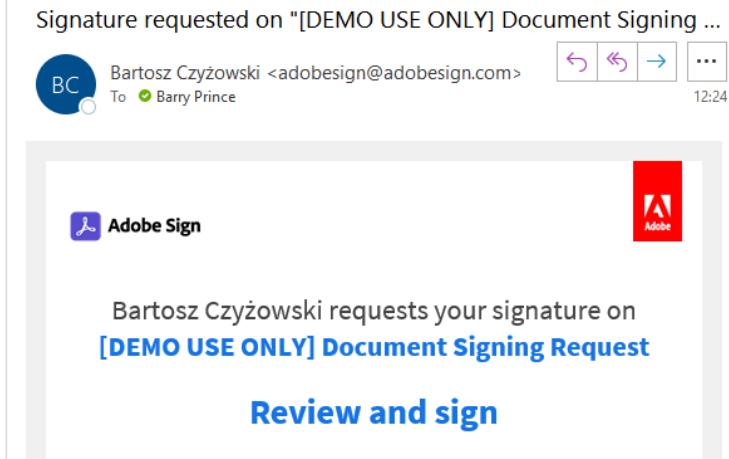
OK

6.15. Signing Services

Description	CARA supports Adobe Sign and DocuSign signing services. Users can request that a document within CARA is electronically signed by a user via either service. Users who are sent signing requests can be existing CARA users or external users.
--------------------	--



Users receive signing requests as external emails with links to the respective signing service.



The signed documents become additional renditions within CARA:

Content/Renditions			
	Format	Size	Created
	Word Document	27.1 KB	11/02/2022, 12:21
	Acrobat PDF	207.6 KB	11/02/2022, 12:21
	Acrobat PDF (AdobeSign)	263.4 KB	11/02/2022, 12:51

Details of how CARA will connect to the services is configured in the Signing Services section of the control panel.

The Request Signing type capability allows users to make signing requests.

A Signing Requests report is available from the Tools menu and is enabled for users via the Signing Requests system capability.

A new job is introduced to CARA for the processing of signing service requests, the cara_signing_request_sync job. The job polls for completed signing requests and generates the rendition creation events.

The setup of signing services is described in this chapter.

Configuring Signing Services

Details of how CARA will connect to the signing service is configured in the Signing Services area of the control panel. Specific account details are required for either service and must be obtained in advance.

1. Within the CARA Control Panel select Signing Services from the General section.
2. In the Signing Services panel click Add, the New Signing Service window opens:

New Signing Service

ID*: [Input Field]

Display Name*: [Input Field]

Instructions to be displayed to the user*: [Input Field]

Signing Service: AdobeSign DocuSign

Save **Cancel**

3. Enter an ID and Display Name for the signing service.
4. Add Instructions to be Displayed to the User when making a signing request.
5. Select the Signing Service to be used. Subsequent fields are service specific:
6. For Adobesign enter the Client ID, Secret and Refresh Token.

New Signing Service

ID*: [Input Field]

Display Name*: [Input Field]

Instructions to be displayed to the user*: [Input Field]

Signing Service: AdobeSign DocuSign

Client ID*: [Input Field]

Secret*: [Input Field]

Refresh Token*: [Input Field]

Save **Cancel**

Please note that Adobesign refresh tokens are by default set to become inactive after 60 days without use. Each time the cara_signing_request_sync job runs the refresh token is used.

7. For DocuSign enter the Private Key, Base URL, Client ID and User ID.

The screenshot shows a configuration dialog titled 'New Signing Service'. It contains several input fields and settings:

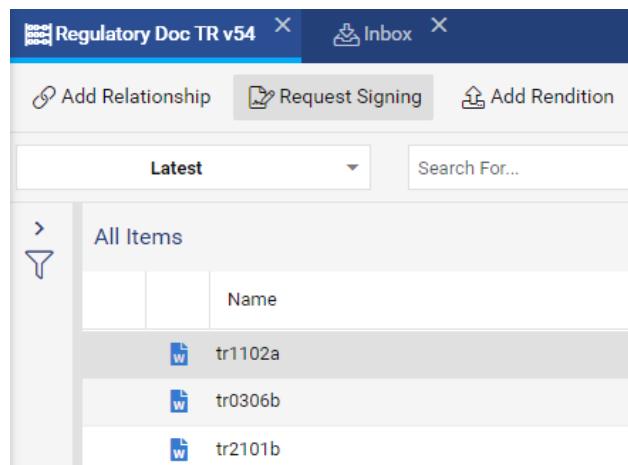
- ID*: A text input field.
- Display Name*: A dropdown menu with a search icon.
- Instructions to be displayed to the user*: A dropdown menu with a search icon.
- Signing Service: A radio button group with 'AdobeSign' and 'DocuSign' options; 'DocuSign' is selected.
- Private Key*: A text input field with a clear button ('X').
- Base URL*: A text input field with a clear button ('X').
- Client ID*: A text input field with a clear button ('X').
- User ID*: A text input field with a clear button ('X').

At the bottom right are 'Save' and 'Cancel' buttons.

8. Save the signing service details.

Request Signing Action and Type Capability

The Request Signing action should be added to the toolbar or right-click menu for views where users will make signing requests:



The Request Signing type capability should be enabled for user groups who will make signing requests.

Signing Requests System Capability

Capability	Default	training_admin
Request Signing	<input type="checkbox"/>	<input checked="" type="checkbox"/>

The Signing Requests system capability provides users with access to the Signing Requests report, available from the Tools menu.

Capability	Default	_admin	_config_manager	training_admin
Copy Cell Content to Clipboard	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Copy Row Content to Clipboard	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Signings Requests	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Tasks - Inbox & Completed Tasks	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Tasks - Inbox - Multiple Tasks Processing	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Tasks - Related Workflows	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Tasks - Send Workflow	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Tasks - Workflow Reporting	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

The Signing Requests report displays all signing requests, pending and completed, for the currently highlighted document.

Document Name	Service	Requested On	Status
tr1102a	AdobeSign	11/02/2022, 12:24	Completed

Page 1 of 1 Close

Signing Request Sync Job

The cara_signing_request_sync job processes signing service requests and generates the rendition creation events. The job should be active and scheduled to run regularly in CARA environments that use signing services.

By default the job will sync with all configured signing services, however it can be set to sync only with selected services by using the services parameter:

The screenshot shows a configuration dialog for a job named 'cara_signing_request_sync'. The dialog has the following fields:

- ID:** cara_signing_request_sync
- Display Name:** Sync Signing Requests
- Method:** Sync Signing Requests (cara_signing_request_sync)
- Status:** Active (unchecked)
- Mode:** Scheduled (selected over Manual)
- Schedule**:
 - Mode: Minutes
 - Every: 30 Minutes
- Managers:** (empty field)
- Notifications:** On Successful Completion (unchecked), On Failure (unchecked)
- Parameters:** A table with one row:

Name	Value
services	

Buttons at the bottom: + Add, - Delete, Save (disabled), Cancel.

6.16. Workflow Templates

Workflow Introduction

Workflow describes a series of tasks which are carried out as part of a process. A workflow could consist of a single or multiple tasks.

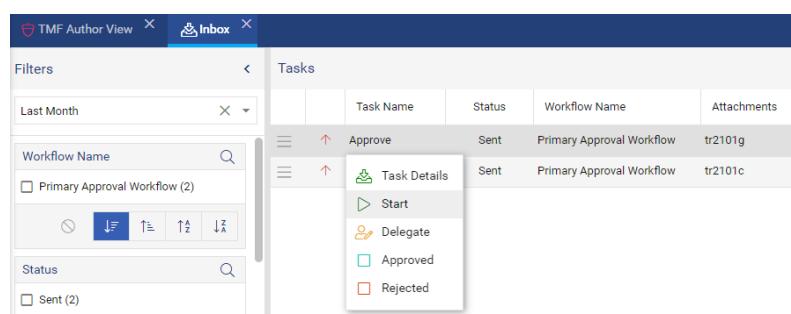
Typically users such as managers or coordinators initiate/start a workflow, selecting recipients and configuring options such as when the workflow is considered overdue.

CARA can also include automatic workflows, for example when users are sent To Be Read notifications following documents becoming effective.

Workflow templates can be specific to a document type and status, or can be general purpose. Workflow templates can also be applied to data object types.

Workflows templates can combine with lifecycles. When users complete tasks the workflow can automatically move the document between lifecycle states.

Task recipients have a dedicated inbox within CARA. Workflow related email notifications can include a link which takes the user directly to the document within CARA.



The screenshot shows the TMF Author View interface with the 'Inbox' tab selected. On the left, there is a 'Filters' panel with dropdowns for 'Last Month', 'Workflow Name' (set to 'Primary Approval Workflow'), and 'Status' (set to 'Sent'). Below these are search and sort buttons. On the right, a 'Tasks' table lists two items:

	Task Name	Status	Workflow Name	Attachments
1	Approve	Sent	Primary Approval Workflow	tr2101g
2	Task Details	Sent	Primary Approval Workflow	tr2101c

A context menu is open over the second task, listing options: Task Details, Start, Delegate, Approved, and Rejected. The 'Approved' option is highlighted with a blue background.

Notifications are automatically sent to users involved in workflows. Task recipients receive emails telling them they have a new task or

an existing task is overdue. Managers who initiated a workflow will be notified when tasks have been completed or delegated.

New Task: Approval for tr0307a in CARA



The following task has arrived in your CARA Inbox.

Task Name: Approval
Task Performer: Barry Prince
Workflow Name: Approval
Message Text: Please approve or reject the attached document.
Sender Name:
Date Sent: 07-03-2020
Workflow Supervisor:

CARA link to the workflow task:

To open the task in CARA, please click on the link below:

<https://training.generis.cloud/cara/task/111>

Managers who initiate a workflow can monitor progress and may be allowed to change recipients or cancel the workflow.

All users with appropriate access to a document can see both current and completed task details. The audit trail includes workflow tasks and electronic signatures where provided.

Creating Workflow Templates

1. Within the CARA Control Panel select **Workflow Templates** from the **General** section.
2. The **Workflow Templates** panel opens, showing the currently configured workflow templates:

ID	Display Name	Version	Updated
office365_collaboration	Office365 Collaboration	2	3/19/21 2:03 PM
studentx_workflow	Primary Approval Workflow	6	2/1/22 9:21 AM
test1105a	1 Step Approval	9	9/17/21 12:13 PM

3. Click **+ Add**.

General

Properties

4. The **New Workflow Template** window opens and is divided into the following 7 tabbed sections:

The screenshot shows the 'New Workflow Template' dialog box with the 'General' tab selected. The form contains the following fields:

- ID:** A text input field.
- Label:** A text input field.
- Options:** A group of checkboxes:
 - Active
 - Available for Manual Creation
 - Allow Only One Active Instance
- Target Duration:** A text input field containing '1 Day(s)'.
- Instructions:** A large text area.
- Users and groups allowed to initiate it:** A dropdown menu.
- Workflow Managers:** A dropdown menu.

At the bottom right are 'Save' and 'Cancel' buttons.

1. Enter a unique **ID** for the workflow template. The ID must start with a letter and can only contain letters (both uppercase and lowercase), numbers, underscores and hyphens. The ID is not typically displayed to end users.
2. Enter a descriptive **Label**. The label will be seen by end users throughout the system.
3. **Active**. Only active workflow templates can be applied to documents, this applies to both templates which will be manually applied to documents by users, and also system templates which are automatically applied to documents based on a condition.
4. **Available for Manual Creation** must be ticked in order for users to be able to manually select the workflow template.
5. **Allow Only One Active Instance**. Select if a document should not be in the same workflow multiple times simultaneously. For

example if a document cannot be subject to 2 or more approvals at the same time.

6. **Target Duration.** The default number of days before the workflow is considered overdue.
7. **Instructions for Workflow Initiator.** Add advice or instructions for the workflow initiator when selecting recipients for each task step. For example, indicate if a step requires a minimum or maximum number of recipients.
8. **Users and Groups Allowed to Initiate.** Select users and groups who will be allowed to apply the workflow template to documents. Specific groups can be created for the purpose within CARA. If left blank, all users will be allowed to initiate the workflow.
9. **Workflow Managers.** Select users and groups who will be allowed to modify instances of the workflow. These users will be able to modify the recipients and cancel individual instances.

Attachments

The screenshot shows the 'Attachments' tab of the 'New Workflow Template' dialog. The tab bar includes General, Attachments (selected), Auditing, Variables, Initialization, On Issue Actions, and Steps. Under 'Required Attachments:', 'Single Document' is selected. The 'Object Type:' dropdown contains 'TMF Studentx (tmf_studentx)'. Under 'Bind To:', 'Selected Version' is selected. The 'Lifecycle:' dropdown contains 'Studentx Lifecycle (studentx_lifecycle)'. The 'Lifecycle State:' dropdown contains 'Draft'. The 'Attachments Menu Actions:' section is empty. At the bottom, there are buttons for 'Save' and 'Cancel'.

1. **Required Attachments.** Select if instances of the workflow must have an attached document.

If **Single** or **Multiple Documents** are selected the user must highlight a corresponding number of documents in the Main View panel before selecting **Tasks > Send Workflow** in order for the template to be visible.

If **None** is selected, documents can still be attached to the workflow, but are not required.

Traditionally workflow tasks involve an activity on a document, however in CARA workflow tasks can also be created which are not related to a document, in which case select **None**.

Please note the following options are only visible if **Required Attachments** is set to **Single** or **Multiple Documents**:

2. **Object Type.** Select the object type required as an attachment. Both document and data object types can be selected..
3. **Bind To.** Select which version of attached documents will be linked to the workflow. Commonly users will be expected to work with, update and view the **Latest** version of the document throughout a workflow.

Alternatively it may be that the specific version of the document **Selected** when the workflow was applied should be associated, even if other versions are subsequently created.

A specific **Version Label** can also be specified, for example Draft or Reviewed.

4. **Lifecycle.** Optionally select a lifecycle which documents must already have applied in order for the workflow template to be selectable.
5. **Lifecycle State.** If a lifecycle has been selected above, additionally select which state of the lifecycle documents must be in before the workflow template is selectable.

6. **Additional Condition.** Optionally add additional conditions for when the workflow template can be applied. Conditions are based on **CQL** (CARA Query Language).
7. **Attachments Menu Actions.** Select the menu actions which will be available when right-clicking the attached document in the Task Details window. Custom actions can be included. Workflow templates which include collaboration should add the respective **Edit in SharePoint** or **Edit in Google Docs** custom menu action.

Auditing

The screenshot shows the 'New Workflow Template' dialog box with the 'Auditing' tab selected. The 'General' tab is also visible at the top. The 'Auditing' tab has a blue background. The form includes the following fields:

- Enable Auditing Of:** A group of checkboxes for 'Sending', 'Cancelling', 'Recipient Change', 'Delegation', and 'Reassign'. 'Not Required' is selected for 'Sending'.
- Signature:*** A group of radio buttons for 'Not Required' (selected) and 'Required'.
- Reason:*** A group of radio buttons for 'Not Required' (selected), 'Required - Free Text' (selected), 'Required - Dictionary', and 'Required - Dictionary or Free Text'.
- Reason Label:** A dropdown menu with a magnifying glass icon.
- Default Reason:** A text input field.

At the bottom right are 'Save' and 'Cancel' buttons.

1. **Enabling Auditing Of.** Select which task related events will be recorded in the audit trail.
2. **Signature.** Select if task recipients are required to enter their electronic signature when completing the task.
3. **Reason.** Select if task recipients are required to provide a reason when completing the task. If a reason is required, select whether recipients should enter free text, choose from a dictionary or have both options available.

4. **Reason Label.** Enter a user friendly label for the Reason dialogue. Note that the label can be entered manually or chosen from the _display_labels dictionary.
5. **Reason Dictionary.** Only visible, and required, if a Dictionary option is selected in the Reason field above. Select a dictionary from which users can select a Reason.
6. **Default Reason.** Optionally enter a default reason.

Variables

ID ↑	Display Name	Type	Multi-V...
department	Department	String	

Buttons: + Add, + Add Sub-Attribute, Edit, Delete, Edit Variables Form, Save, Cancel.

Workflow initiators can be prompted to enter additional information that can subsequently be used within the workflow. For example a workflow initiator could enter a department or other information which then determines which group of users can be selected as task recipients.

A variable is defined for each value the workflow initiator should enter, and those variables are placed on a dedicated properties form.

The form is displayed to the workflow initiator immediately after the template selection window.

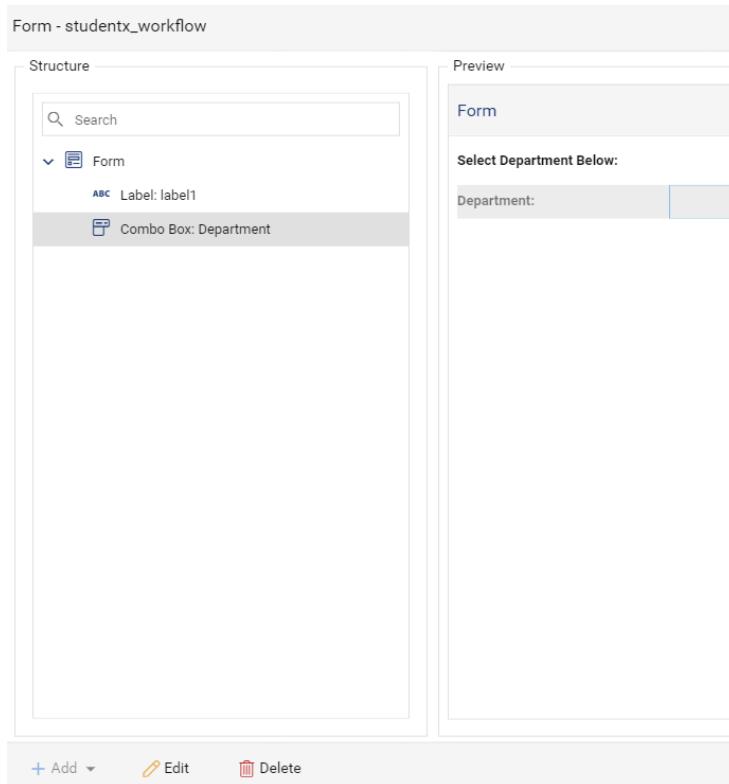
To define variables and create a form:

1. Click **+ Add** to open the **New Variable** window:

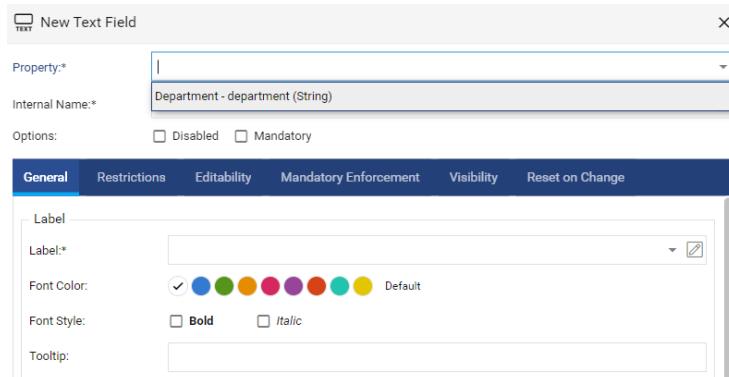
The screenshot shows the 'New Variable' dialog box. It has fields for 'Name:' (department), 'Display Name:' (Department), and 'Data Type:' (String). There is also a checkbox for 'Multi-Value' and a note about 'Source Object Property to Copy as Default' which is set to 'Department'. At the bottom are 'Save' and 'Cancel' buttons.

2. Enter a **Name** for the variable. The name can only contain lowercase letters, numbers and underscores.
3. Enter a **Description**.
4. Select a **Data Type**.
5. If more than 1 value can be entered for the variable, check **Multi-Value**.
6. Optionally enter a **Source Object Property to Copy as Default**.
The source object property must be the same format as the variable data type.
7. **Save** the variable, returning to the workflow templates variables tab.
8. **Add** further variables as required.

9. Once all variables have been defined, select **Edit Variables Form**.
10. The form builder window opens, with the same features available as when defining a document properties form. Please see the properties form section for details.



11. When adding a field control, such as a text field, only variables defined in the previous step can be selected as properties:



12. **Label** fields can be added to provide guidance to workflow initiators when completing the form:

The screenshot shows the 'Edit Label' dialog box with the following settings:

- Display:** Static Text (radio button selected)
- Internal Name:** label1
- General Tab:**
 - Label:** Select Department Below
 - Font Color:** Default (selected)
 - Font Style:** Bold (selected)
 - Tooltip:** (empty text area)

Initialization

The screenshot shows the 'Initialization' tab in the 'Workflow Templates' editor. The tab bar includes General, Attachments, Auditing, Variables, Initialization (selected), On Issue Actions, and Steps. The main area contains a text input field for 'Initialization/Validation Script:' which is currently empty.

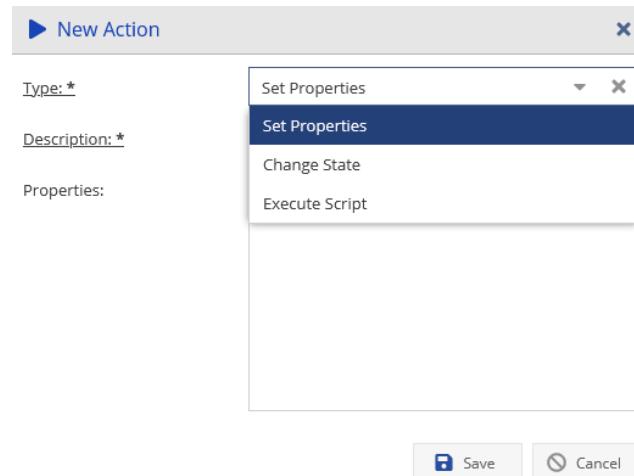
Optionally add an **Initialization/Validation Script**.

On Issue Actions

The screenshot shows the 'On Issue Actions' tab in the 'Workflow Templates' editor. The tab bar includes General, Attachments, Auditing, Variables, Initialization, On Issue Actions (selected), and Steps. The main area contains a table for 'Execute Actions:' with columns for 'Type' and 'Display Name'. At the bottom are buttons for '+ Add', 'Edit', and 'Remove'.

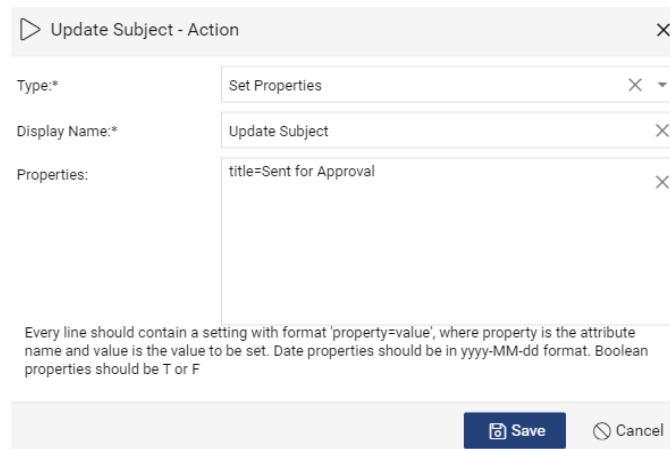
Actions can be performed on attached documents when the workflow is issued. This is in addition to actions based on task outcomes. To add an On Issue Action:

1. Click **+ Add** to open the **New Action** window:



2. Select a **Type** of action:

- a. **Set Properties.** Add a **Display Name** and in the **Properties** field enter a valid **CQL** statement. Note that as illustrated in the example below, string values do not need to be enclosed in quotes:



- b. **Change State.** Add a **Display Name** and select the **lifecycle State** the document is to be moved to:

The screenshot shows the 'New Action' dialog box. It has three input fields: 'Type:' set to 'Change State', 'Display Name:' set to 'Promote to In-Approval', and 'State:' set to 'Approved'. At the bottom are 'Save' and 'Cancel' buttons.

Type:*	Change State
Display Name:*	Promote to In-Approval
State:*	Approved

Save **Cancel**

- c. **Execute Script.** Add a **Display Name** and enter a valid **Script** to be executed:

The screenshot shows the 'New Action' dialog box. It has three input fields: 'Type:' set to 'Execute Script', 'Display Name:' empty, and a large 'Script:' text area which is currently empty. At the bottom are 'Save' and 'Cancel' buttons.

Type:*	Execute Script
Display Name:*	
Script:	

Save **Cancel**

3. **Save** the new action, returning to the **On Issue Actions** tab. Add further actions as required.

Steps

The screenshot shows the 'Steps' tab in the workflow editor. It displays a table with one row for the step 'approve'. The 'Outcomes' column contains 'approved,rejected' and the 'Transitions' column contains 'End,End'. At the bottom are 'Add', 'Edit', and 'Remove' buttons, and 'Save' and 'Cancel' buttons at the very bottom.

Step Name	Outcomes	Transitions
approve	approved,rejected	End,End

Add **Edit** **Remove**

Save **Cancel**

Steps are the individual steps or tasks within the workflow. A workflow can consist of one or more steps. Many workflows are just a single step, for example to approve a document.

Each step requires recipients to be selected either manually or automatically. Recipients are provided with instructions that describe the actions/tasks to be completed.

Each step needs all valid outcomes defined, for example an approval step could result in the document being approved, rejected or marked as not applicable. A To Be Read (TBR) step might have only one possible outcome such as document acknowledged, or multiple outcomes.

Each step also needs all of the possible transitions defined, that is what happens next based on the outcome. The outcome of a Review step could be to start an Approval state, but only if the Review was successful. If unsuccessful the Review state could start a draft state. All workflows must lead to an end state, even if a workflow consists of just a single step/task, that step/task must have an end outcome.

Actions can be carried out automatically based on the outcome of each step, for example the outcome of a successful approval step could be to change the state of the document to Approved. If unsuccessful the state could be changed to Draft.

To add a **Step** click **+ Add**, the **New Step** window opens. The **New Step** window is divided into 4 tabs:

The screenshot shows a configuration dialog for a new step named 'studentx_workflow'. The 'General' tab is active. The 'ID:' field is empty. The 'Label:' field is empty. The 'Priority' section shows 'Medium' is selected. The 'Target Duration:' field contains '5 Day(s)'. The 'Instructions:' field is empty. The 'Roles' section is empty. At the bottom, there are buttons for '+ Add', 'Edit', 'Remove', 'Save' (highlighted in blue), and 'Cancel'.

1. General.

- a. Enter an **ID**. The ID can only contain letters (uppercase as well as lowercase), numbers, underscores and hyphens.
- b. Enter **Label**. Labels can be entered manually or selected from the _display_labels dictionary.
- c. Select a **Priority**. The priority is indicated to recipients by a colour coded icon in the task inbox. Low priority tasks have a green down arrow icon. High priority tasks have a red up arrow. Medium priority tasks do not have an additional icon.
- d. Select a **Target Duration**. Notifications, most commonly by external email, can be triggered when the target duration is exceeded.
- e. Provide **Instructions** for recipients describing the action/task to be completed.

- f. **Roles** are optional and can be defined if auditing requirements will vary based on the recipients role. For example approvers could be required to provide their electronic signature when selecting an outcome or a step, but readers may not be required to do so. If roles are defined, when users initiate a workflow, each task recipient has a role selected. To create a role, click **+ Add** and specify an **ID** and **Display Name** for each role:

Role	ID	Display Name
approver	approver	Approver
reader	reader	Reader

When roles are defined, the role the recipient was allocated is displayed within the Workflow Details window:

Task Name:	Approve	Sent Date:
Workflow Name:	Primary Approval Workflow	Sent By:
Target Date:	Feb 11, 2022	Status:
Role:	Approver	
Instructions:	Please approve or reject the attached document.	

2. Recipients.

The screenshot shows the 'Recipients' tab of the CARA configuration interface. The 'Restrict To' section has 'None' selected. The 'Recipients' section includes fields for Selection Method (Automatically Pre-Populated or To Be Selected by User), Define Default Recipients (None), Minimum and Maximum number of recipients, Restrictions (checkboxes for Prevent Sequence Change, Prevent Recipient Change, Prevent Recipient Delegation), and Sequence (Parallel). Buttons for Save and Cancel are at the bottom.

- Restrict Recipients To:** If the workflow initiators are to have no restrictions on who they can select as task recipients select **None**.

To predefine the users from which the task initiator can choose, select **List** and then select users and groups from the selection window (pencil icon) or directly within the **Allowed Recipients** field (drop-down arrow):

The screenshot shows the 'Recipients' tab of the CARA configuration interface. The 'Restrict To' section has 'List' selected. The 'Allowed Recipients' field contains 'training_reader' and '_group_manager' with a dropdown arrow.

Variable Values, where defined within the workflow template, can be used to predefine the available users. Select a variable from the drop-down list.

- Selection Method.** Recipients can be **Automatically Pre-Populated** or **Selected By the User** (workflow initiator).

If the selection method is set to **Automatically Pre-Populated**, select to **Populate With** either **List** or **Variable Values**:

If selection method is set as **To Be Selected By the User**, optionally a **Default** list of recipients can be defined either by selecting from a list or using a pre-defined variable value:

A **Minimum** and **Maximum** number of recipients can be set, preventing users from issuing a task until an appropriate number of recipients have been selected:

c. **Restrictions.** Optionally select to:

Prevent Sequence Change. If unselected, workflow initiators will be allowed to change the order of task steps.

Prevent Recipient Change. If unselected, workflow managers will be allowed to change recipients for future tasks.

Prevent Recipient Delegation. If unselected, recipients will be allowed to delegate their task to another user. Recipients cannot delegate if they have already marked their task as started.

- d. **Sequence.** Sequence is relevant when there is more than one recipient.

When sequence is set to **Parallel**, all recipients are sent the task at the same time.

When sequence is set to **Serial**, only the first recipient in the list is initially sent the task. Once they have completed the task the next recipient in the list is sent the task, and so on.

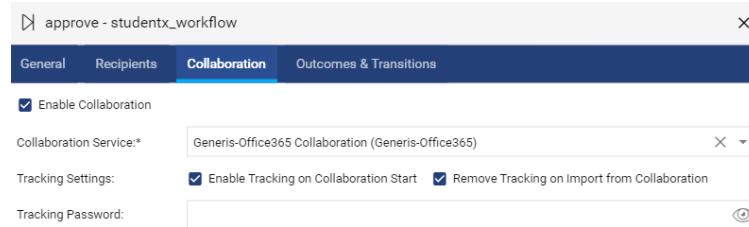
- e. **All Recipients Must Complete.** This option is significant when there is more than one recipient.

If unselected, only 1 recipient must complete the task for the workflow to move to the next step.

If selected, which is the default, all recipients must complete the task before the workflow moves to the next step.

3. Collaboration.

CARA workflows support Office365 and Google Docs collaboration services, allowing multiple users to simultaneously edit a document. If enabled for a workflow template, the collaboration service to be used must be specified. Collaboration services must be configured in advance, please see the separate section of this manual for details.



- a. **Enable Collaboration.** Tick to enable collaboration, which enables the following options:
- b. **Collaboration Service.** Select the collaboration service to be used.
- c. **Tracking Settings.** Document change tracking can be automatically enabled on a document when the task is started. Enabling this option displays the **Tracking Password** field beneath, where the tracking password specified as part of the collaboration services configuration must be entered. Tracking can also be automatically removed from a document when the task is completed.

4. **Outcomes & Transitions:**

approve - studentx_workflow

Outcomes:

Outcome	Label	Signature	Reason
approved	Approved	✓	
rejected	Rejected	✓	

+ Add Edit Remove

Transitions:

Condition	Next Step	Actions
At Least One Outcome = approved	End	Approved
At Least One Outcome = rejected	End	Rejected

+ Add Edit Remove

Save Cancel

- a. **Outcomes.** Click **+ Add** to add an outcome, the **New Outcome** window opens:

New Outcome - approve

ID:

Label:

Color:
 ● ● ● ● ● ● ● Default

Comment:
 Hidden Optional Required

Auditing

Signature:
 Not Required Required

Reason:
 Not Required
 Required – Free Text
 Required – Dictionary
 Required – Dictionary or Free Text

Reason Label:

Roles:

Save Cancel

- I. Enter an **ID**. The ID can only contain letters, numbers, underscores and hyphens.

- II. Enter a **Label**. Labels can be entered manually or selected from the _display_labels dictionary.
 - III. Select a **Color**.
 - IV. **Comment**. Choose if when completing the task, recipients are **Required** to enter a comment, comments are **Optional**, or no comments can be entered (**Hidden**). Comments are visible in the audit trail and the workflow details window.
 - V. **Signature**. Select if users are required to provide their electronic signature (name and password) when completing the task.
 - VI. **Auditing**. Select if users are required to enter a reason when completing a task. Users can be provided with the option to enter free text, select from a dictionary based drop-down list or have both options.
 - VII. Optionally enter a **Reason Label**. Labels can be entered manually or selected from the _display_labels dictionary.
 - VIII. If roles have been defined for the workflow template, for each role specific auditing requirements can be defined. Click the **Roles** field menu icon and in the **New - Role Audit** window select the auditing required for a selected role. Click **Save** to return to the new outcome window.
 - IX. **Save** the new outcome.
- b. **Transitions**. Click **+ Add** to add a transition outcome, the **New Transitions** window opens:

New Transition - approve

Condition:*

- At Least One Outcome...
- All Outcomes Are...
- Workflow Aborted

Outcome:*

Execute Actions:

Type	Display Name

+ Add Edit Remove

And Go To:*

Save **Cancel**

- i. **Condition.** Select whether the transition is conditional on **At Least One Outcome** being true or **All Outcomes** are true. Alternatively the transition can be set to trigger if the **Workflow is Aborted**.
- ii. **Outcome.** Select which outcome the transition is based on.
- iii. **Execute Actions.** Click **+ Add** to add a **New Action**:

New Action

Type:*

Display Name:*

Set Properties
Change State
Execute Script

Save **Cancel**

- iv. Select a **Type** of completion action. Note that if an action fails, the workflow task will be placed in the Halted status and a notification will be sent to the task initiator. Once the issue has been resolved the workflow can be resumed via a script.

1. **Set Properties.** Add a **Display Name** and in the **Properties** field enter a valid **CQL** statement. Note that as illustrated in the example below, string values do not need to be enclosed in quotes:

New Action

Type:*

Set Properties

Display Name:*

Set Subject

Properties:

doc.subject=Reviewed

Every line should contain a setting with format 'property=value', where property is the attribute name and value is the value to be set. Date properties should be in yyyy-MM-dd format. Boolean properties should be T or F

Save Cancel

2. **Change State.** Add a **Display Name** and select which lifecycle state the document will move to:

New Action

Type:*

Change State

Display Name:*

Mark as Approved

State:*

Approved

Save Cancel

3. **Execute Script.** Add a **Display Name** and enter a valid **Script** to be executed:

The screenshot shows the 'New Action' dialog box. The 'Type:' field is set to 'Execute Script'. The 'Display Name:' field is empty. The 'Script:' field contains a script code area with a small edit icon. At the bottom, there are 'Save' and 'Cancel' buttons.

4. Workflow templates with collaboration enabled have 2 additional actions available, **Complete Collaboration** and **Abort Collaboration**:

The screenshot shows the 'New Action' dialog box with the 'Type:' dropdown open. The options listed are Set Properties, Change State, Execute Script, Complete Collaboration (which is highlighted), and Abort Collaboration. At the bottom, there are 'Save' and 'Cancel' buttons.

5. Save the new action.

- v. **And Go To:** Select the next step, for example a successful outcome for an Approval step could lead to an Effective step. Steps do not have to follow a linear order.

All workflows must reach an End step, including workflows with only a single step.

The screenshot shows the 'And Go To' configuration dialog box. The 'And Go To:' field is set to 'End'. The 'Completion Status:' field is set to 'Document Reviewed' with a checked 'Mark as Successful' checkbox. At the bottom, there are 'Save' and 'Cancel' buttons.

- vi. The following options, **Completion Status** and **Mark as Successful** are only available when the **Next Step** is set to **End**.

The **Completion Status** is shown as a column within the **Related Workflows** report:

	Workflow Name	Attachments	Completion Status	Completion Date
✗	Studentx Workflow	tr1903b		
✓	Studentx Workflow	tr2403a	Document Approved	3/25/21 3:07 PM
✓	Student6 Workflow	student6_test_content	Document Approved	3/25/21 6:51 PM
✓	Student6 Workflow	student6_test_conte...	Document Approved	3/25/21 6:57 PM
✗	Student6 Workflow	student6_test_conte...		

Mark as Successful. Select if the transition is considered to be successful. More than one transition can be marked as successful.

Workflows with an end outcome marked as successful are listed in green within workflow reports. Workflows with an end outcome that is not marked as successful are listed in orange within workflow reports:

In addition, workflows which are overdue are listed in red and workflows which were cancelled are listed in grey.

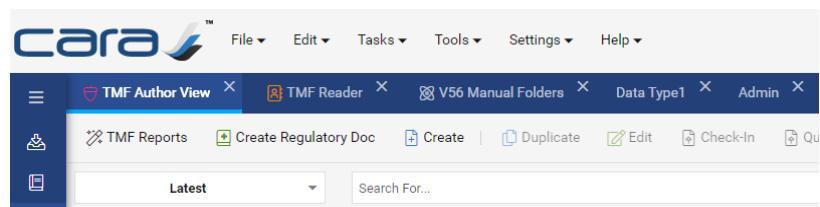
- c. **Save** the workflow template.

6.17. Workspaces

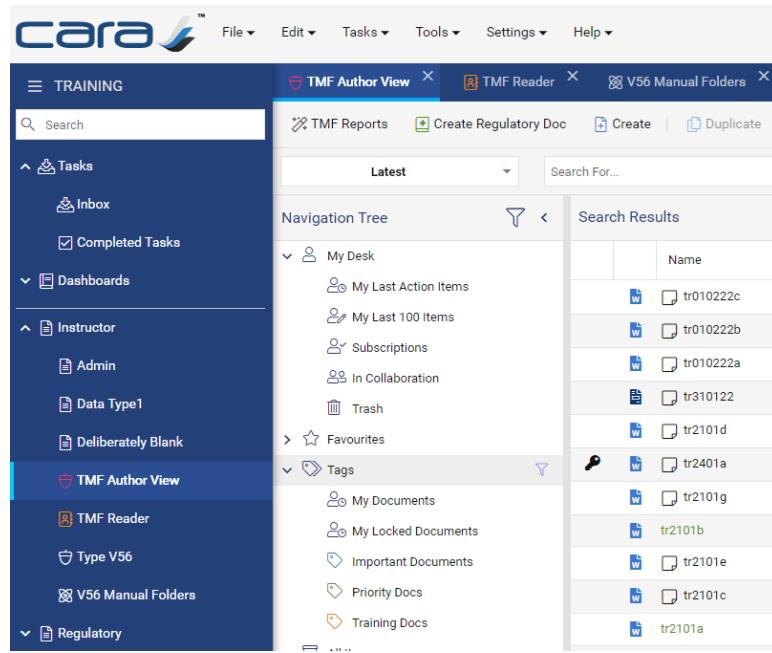
Description

A workspace is a collection of views made available to groups of users. In addition to determining which views users have access to, workspaces are used to limit which configuration areas groups of administrators are allowed to modify.

Views are presented to users as a series of tabs on a toolbar running across the top of the user interface. Views determine the user interface layout and functions available. Each type of document will have at least 1 view, and potentially multiple.



Views are grouped together into workspaces. Workspaces and their views are listed and initially selected from the collapsible left-hand side menu:



Administrators create workspaces and add views to them.

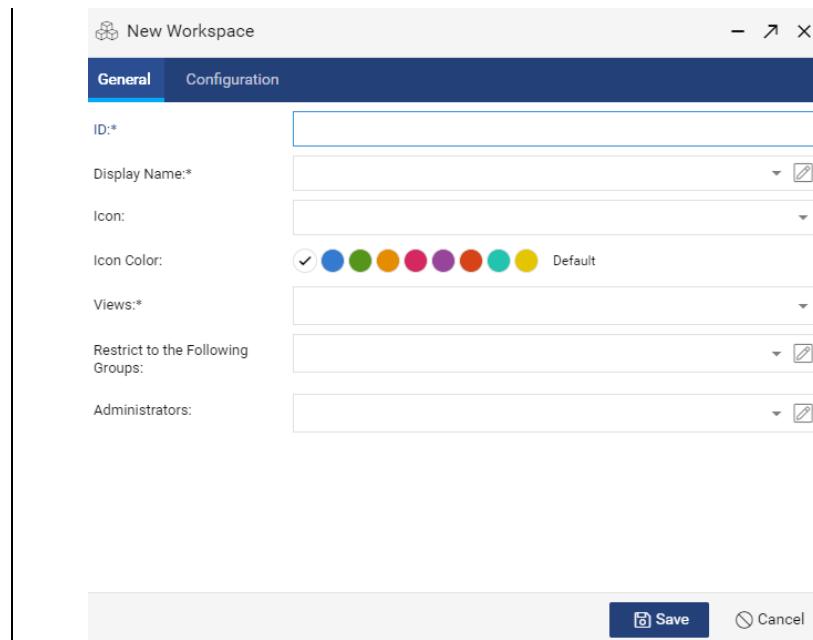
The screenshot shows the 'instructor - Workspaces' configuration window. The 'General' tab is active. The 'ID' field contains 'instructor'. The 'Display Name' field is set to 'Instructor'. The 'Icon' field is empty. The 'Icon Color' section shows a color palette with 'Default' selected. The 'Views' section lists several workspace views: 'TMF Author View (tmf_studentx)', 'Admin (tmf_studentx_admin)', 'TMF Reader (tmf_studentx_tmf_reader)', 'Data Type1 (data_type1)', 'Deliberately Blank (tmf_studentx_deliberately_blank)', 'Type V56 (type_v56)', and 'V56 Manual Folders (v56_manual_folders)'. Below these are fields for 'Restrict to the Following Groups' and 'Administrators', both of which are currently empty. At the bottom right are 'Save' and 'Cancel' buttons.

Workspaces can be available to all users or restricted to selected groups. Workspaces are versioned each time they are updated.

The **Workspaces** panel displays the **ID**, **Description**, **Version**, date **Updated** and who the workspace was **Updated by**.

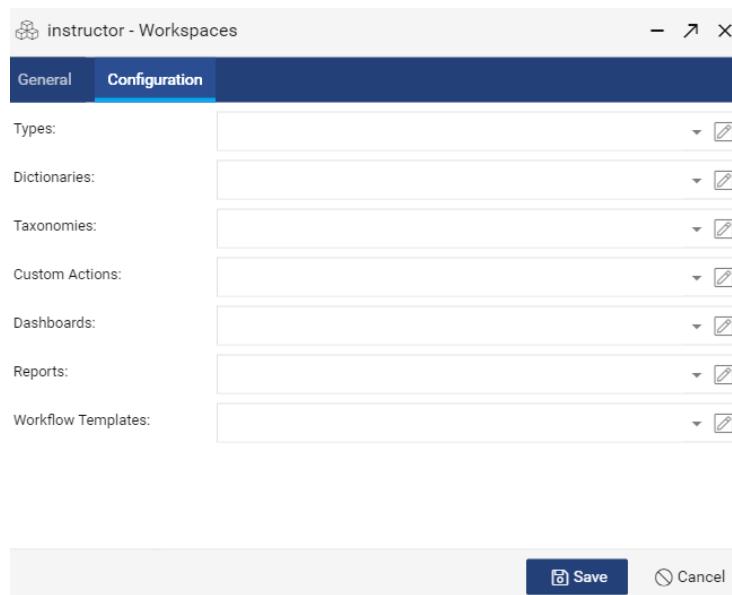
Creating a Workspace

1. Within the CARA Control Panel select **Workspaces** from the **General** panel.
2. Select **Add**, the **New Workspace** window opens:



3. Enter a unique **ID**.
4. Enter a **Display Name**.
5. Select an **Icon** and **Icon Color**.
6. Click on the drop-down arrow in the **Views** field to select the views that will be available within the workspace.
7. Click **Save** and return to the **New Workspace** window.
8. Click on the drop-down arrow in the **Restrict to the Following Groups** to list groups, or click the pencil icon to open the **Groups Selection** window. Select the groups that will have access to the workspace. If no groups are selected, all users will have access to the workspace.
9. Click **Save** and return to the **New Workspace** window when completed.
10. Select the groups that will be **Administrators** within the workspace. Workspace administrators will be permitted to update the configuration elements selected in the **Configuration** tab.

11. In the **Configuration** tab select the individual configuration elements the workspace's administrators will be permitted to modify. For example workspace administrators might be allowed to update specific dictionaries and taxonomies, but not other configuration areas such as type configuration or custom actions:



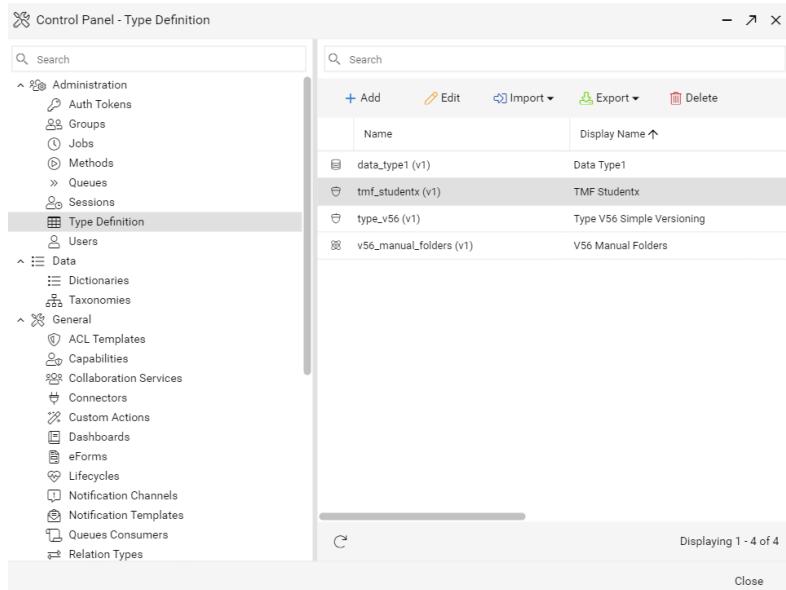
12. **Save** the workspace configuration.

7. Type Configuration

Description

A large part of the configuration of CARA is type related, including GUI layout and user functionality.

Core object configuration, such as attribute definitions and content options, are defined within the **Type Definition** panel, located in the **Administration** section:



The screenshot shows the 'Control Panel - Type Definition' window. On the left is a navigation tree with sections like Administration, Data, and General, with 'Type Definition' currently selected. On the right is a table listing four type definitions:

Name	Display Name
data_type1 (v1)	Data Type1
tmf_studentx (v1)	TMF Studentx
type_v56 (v1)	Type V56 Simple Versioning
v56_manual_folders (v1)	V56 Manual Folders

At the bottom, it says 'Displaying 1 - 4 of 4' and has a 'Close' button.

Further type configuration is then specified in the **Type Configuration** section, as detailed in the following chapters.

The screenshot shows the CARA Control Panel with the title "Control Panel - TMF Studentx (tmf_studentx)". On the left, there's a sidebar with various configuration categories like Collaboration Services, Connectors, Custom Actions, Dashboards, eForms, Lifecycles, Notification Channels, Notification Templates, Queues Consumers, Relation Types, Reports, Script Libraries, Sequences, Signing Services, Workflow Templates, and Workspaces. Below these are Maintenance, Configuration Export, Configuration Import, Configuration Tags, and Type Configuration. Under Type Configuration, there are entries for Data Type1 (data_type1), TMF Studentx (tmf_studentx), Type V56 Simple Versioning (type_v56), and V56 Manual Folders (v56_manual_folders). The main panel displays a table of configuration items with columns for Configuration Type, Version, and Updated. The table is divided into three sections: General (6 items), Initialization (3 items), and Processing (5 items). The "View (TMF Author View) - Default" item is highlighted in the General section. The table shows details like Type Capabilities (Version 8, Updated 10/20/21 5:00 PM), Form (Version 12, Updated 1/24/22 5:21 PM), and View (TMF Author View) - Default (Version 57, Updated 2/1/22 9:08 AM). The bottom right of the table area says "Displaying 1 - 14 of 14".

Configuration Type	Version	Updated
Type Capabilities	8	10/20/21 5:00 PM
Form	12	1/24/22 5:21 PM
View (TMF Author View) - Default	57	2/1/22 9:08 AM
View (Admin)	1	5/26/21 1:34 PM
View (Deliberately Blank)	1	8/13/21 3:59 PM
View (TMF Reader)	9	5/28/21 12:46 PM
Classification Definition	3	9/22/21 12:20 PM
Lifecycle Assignment	3	3/24/21 6:34 PM
Template Assignment	4	9/22/21 12:21 PM
Auditing	4	10/21/21 8:23 PM

Type configuration elements are grouped into 3 areas, **General**, **Initialisation** and **Processing**.

The **Type Configuration** panel displays the **Version**, when the configuration was last **Updated** and who the configuration was last **Updated By** for each configuration element.

7.1. Type Capabilities

Description Type capabilities are type specific functions provided to groups of users, in contrast to the system wide general capabilities.

Type capabilities are functions which determine how users work with documents and other objects and include fundamental content management options such as the ability to view, create and edit documents.

Capability	Default	training_reader
Add Relationship	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Add Rendition	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Annotate	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Audit Trail Report	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Check-in	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Check-out	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Collaborate	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Copy ID	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Copy URL	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Create From eForm	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Create From Template	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Create Object (Placeholder)	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Typically groups are created which define sets of users with a business role, these groups are then provided with the type capabilities which are required to carry out the role.

For example, a clinical authors group is defined and then granted the capability to create and edit clinical documents.

Users who belong to one or more groups with type capabilities specified are granted those type capabilities cumulatively (except those specified as Default). Users who do not belong to any groups with a type capability specified are granted only the Default type capabilities.

The following **Type Capabilities** can be granted to users:

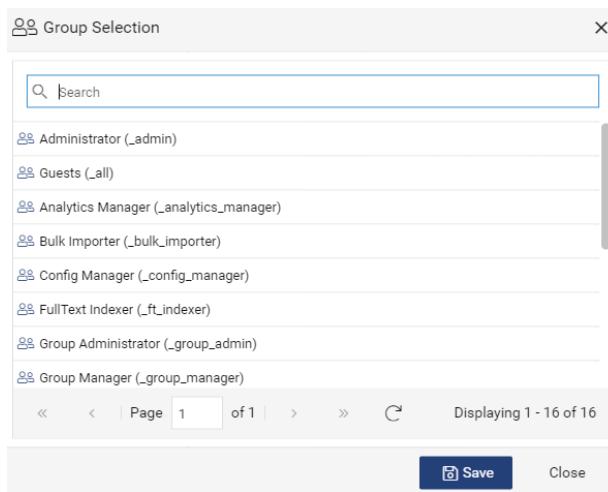
Add Relationship
Add Rendition
Annotate
Audit Trail Report
Check-in
Check-out
Collaborate
Copy ID
Copy URL
Create Folder
Create from eForm
Create from Template
Create Object (Placeholder)
Create Root Folder
Create Template
Delete
Delete Folder
Dump
Duplicate Object
Edit
Export
Export PDF with Annotations
Favorites
Folder Navigation
Folder Restrictions
Import Object
Import Template
Manage Tags
Move Folder
Permissions
Preview
Properties
Quick Check-in
Redact
Remove Relationship

Remove Rendition
Rename Folder
Restore from Trash
Save Private Search
Save Public Search
Signatures
Status Manager
Structure Manager
Subscribe
Tag
Trash
View
View PDF

Capabilities for creating folders, creating root folders, moving folders, renaming folders and deleting folders are only present for object types which use manual linking. For types which use automatic folder linking, the Empty Folders Cleanup job can be used to delete empty folders, see the Jobs and Methods chapter for details.

Configuring Type Capabilities

1. Within the CARA Control Panel select **Type Capabilities** from the **Type Configuration** panel.
2. Click **Add Role**, the **Group Selection** window opens:



3. Select a group and click **Save**, returning to the **Type Capabilities** window.
4. A new column for the group will have been added to the **Type Capabilities** window, locate the column and for each **Capability** which should be granted to the group click once in the corresponding cell to add a tick. Once all appropriate capabilities have been added click **Save**.

7.2. Forms

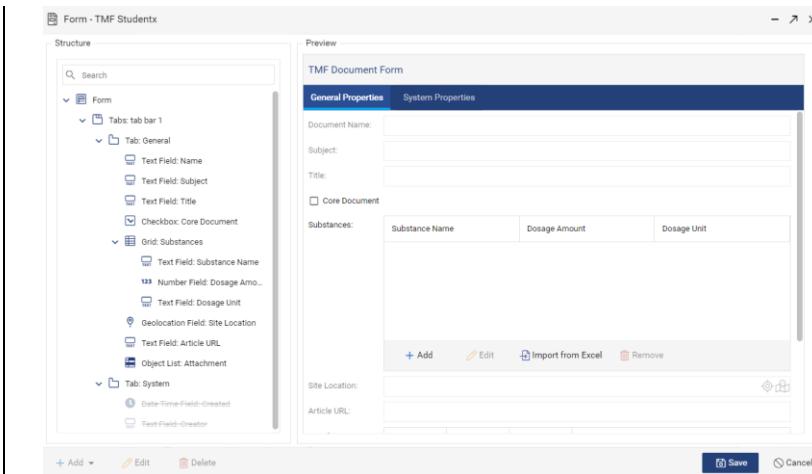
Description

Forms are used to display the attributes of an associated object or document type, they are also presented to users for entering attribute values when a document or object is created or edited.

The screenshot shows a software interface for managing a document. At the top, a title bar reads "TMF Document Form: tr010222c". Below it, a navigation bar has "General Properties" selected, with "System Properties" as an option. The main area contains several input fields and a table:

- Document Name:** tr010222c
- Subject:** Regulatory Document 89
- Title:** (empty field)
- Core Document:** (checkbox is unchecked)
- Substances:** A table with columns "Substance Name", "Dosage Amount", and "Dosage Unit". It currently has one row: "Substance Name" is empty, "Dosage Amount" is empty, and "Dosage Unit" is empty.
- Buttons:** "+ Add", "Edit", "Import from Excel", and "Remove".
- Site Location:** (empty field) with a location pin icon.
- Article URL:** (empty field) with a link icon.
- Attachments:** A table with columns "Name", "Creator", and "Created". It currently has one row: "Name" is empty, "Creator" is empty, and "Created" is empty.
- Buttons:** "Create", "... Select", and "Remove".
- Bottom Buttons:** "Save" (highlighted in blue) and "Cancel".

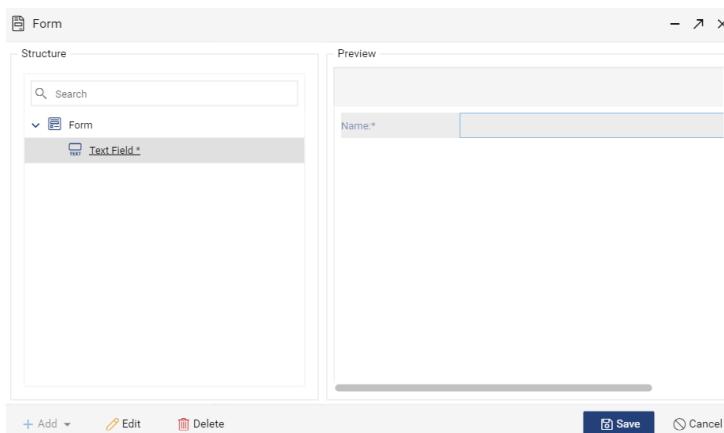
Each object or document type will have a single associated form. Forms are created and edited using the form builder tool described below. Typically, related properties are grouped together into tabbed sections on the form.



Each attribute on a form can have conditions which determine when the attribute is visible, editable or required. Attributes can have a single or multiple conditions based on the user/group or object. Forms can include text labels and graphical icons.

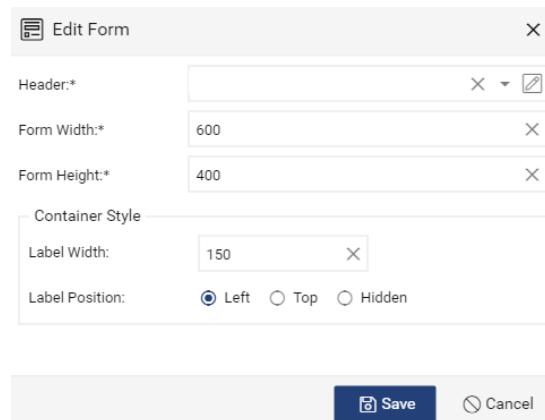
Creating Forms

- When a new type is defined and saved a properties form is created automatically. Initially the properties form contains only a single text field: `object_name`. To update the properties form, within the CARA Control Panel select a type within the **Type Configuration** panel. Within the **Type Configuration** window highlight **Form** and select **Edit**. The form builder window opens:

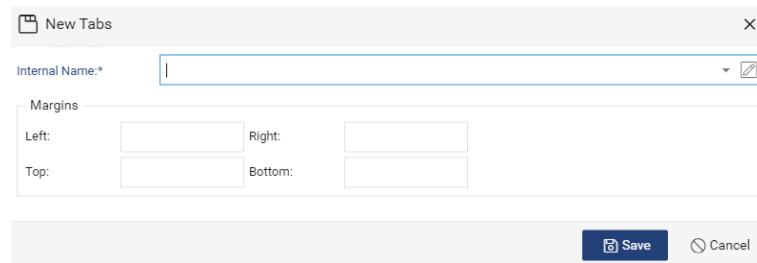


- The form builder window is divided into 2 main panels, the left-hand side **Structure** panel where form elements are added and repositioned, and the right-hand side **Preview** panel.

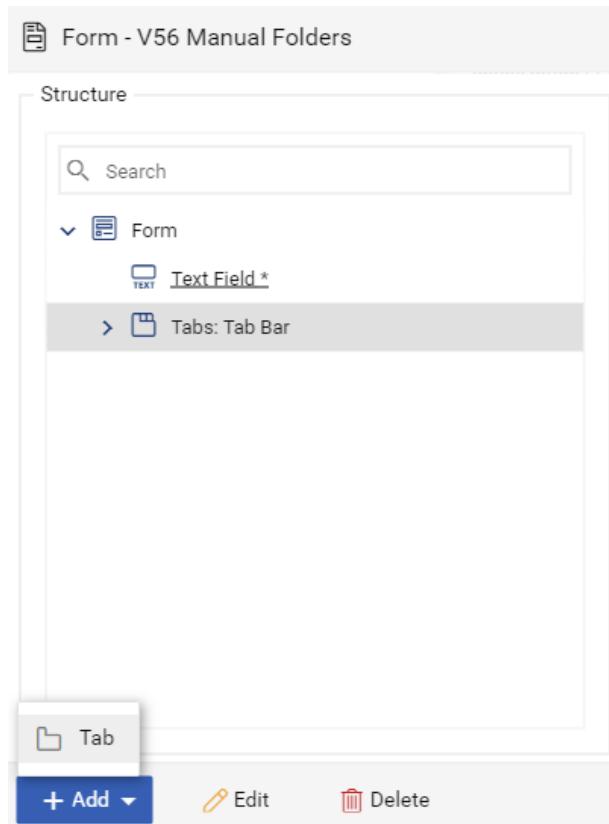
3. A search field is located at the top of the form builder window and can be used to quickly locate fields by their attribute name and label.
4. Highlight the **Form** icon and click **Edit** to open the **Edit Form** window where overall form dimensions are set:



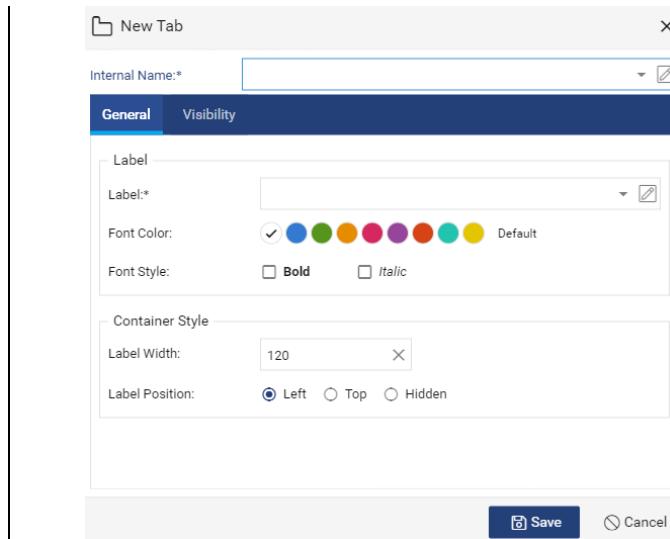
5. Enter a **Header** title, which is displayed at the top of the form window when users view the form.
6. Specify the overall **Form Width** and **Form Height** in pixels.
7. For the **Container Style** specify a **Label Width** and **Label Position**.
8. **Save** and return to the form builder window.
9. Attribute fields can be entered directly onto the form in a simple list one after another. However it is recommended placing fields within tabs. Tabs must be entered onto a tab bar. With the **Form** icon highlighted select **Add > Tabs** to add a tab bar, the **New Tabs** window opens:



10. Enter an **Internal Name**, which will not be visible to users.
11. Set **Margins** for the tab bar or leave the default placement.
12. **Save** and return to the form builder window.
13. Highlight the new tab bar icon and select **Add > Tab:**



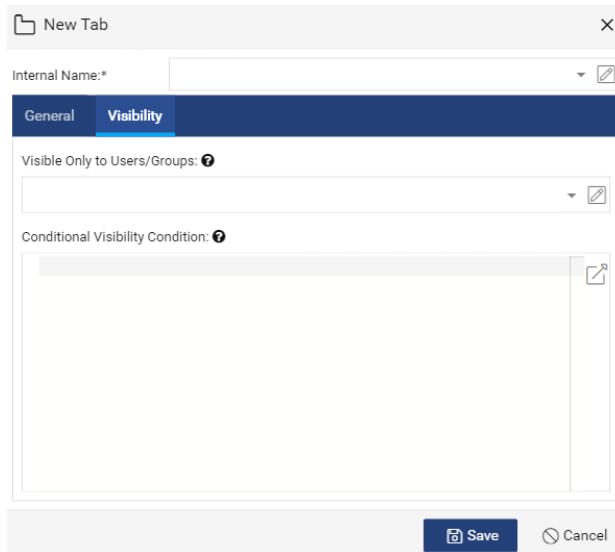
14. In the **New Tab** window enter an **Internal Name** for the tab, which will not be visible to users.



15. In the **General** tab enter a user friendly **Label** or select one from the `_display_labels` dictionary in the drop-down list. Form control labels which have been selected from the `_display_labels` dictionary are shown in bold in the form builder window.
16. Select a **Font Color**.
17. Select a **Font Style** of **Bold** or **Italic**, if appropriate.
18. For the **Container Style** specify a **Label Width** and **Label Position**.
19. Individual tabs can be set to only be visible if a defined condition returns as true. This allows a single form to be created for a document or object type, with the form being displayed differently to different groups of users. Conditions use CARA Query Language syntax.

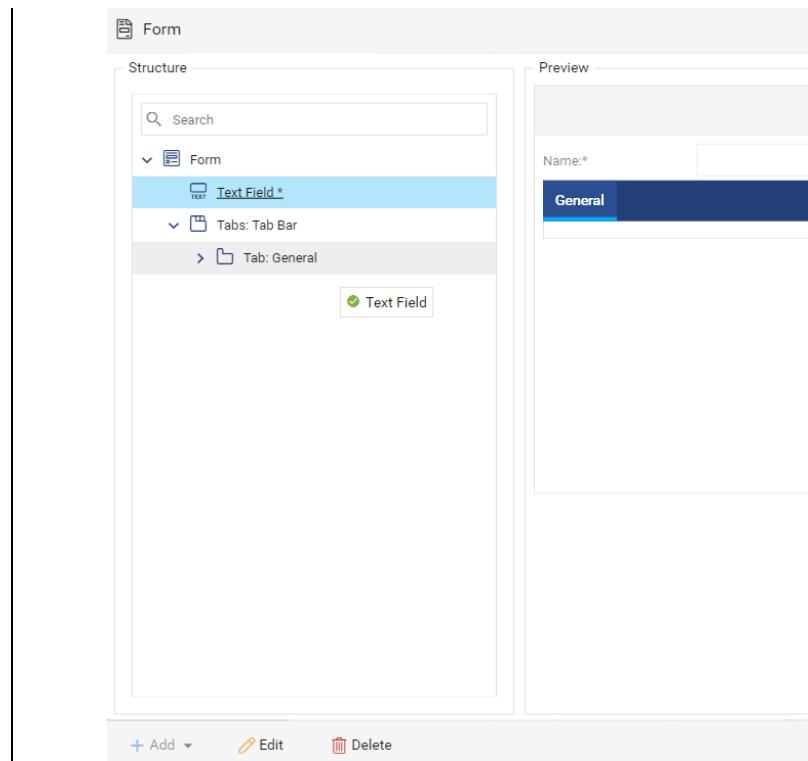
For example, a form might contain a tab which displays properties only when the form is in an approved status, with a condition such as `status="Approved"`. To create a condition click the **Visibility** tab and enter an appropriate condition.

Please note that conditions are not applied while editing the form and so their effects are not apparent in the form preview. To see a condition applied save the form and then view with a document.



20. Once the tab configuration is complete click **Save** and return to the main form builder window:

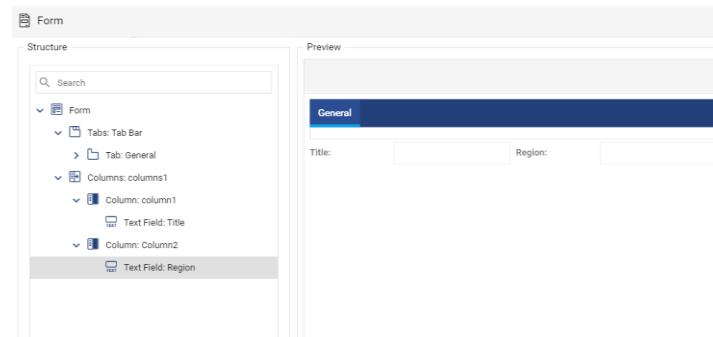
The default object_name field can be dragged and dropped into the new tab, in the Structure panel select the field and drag onto the new tab:



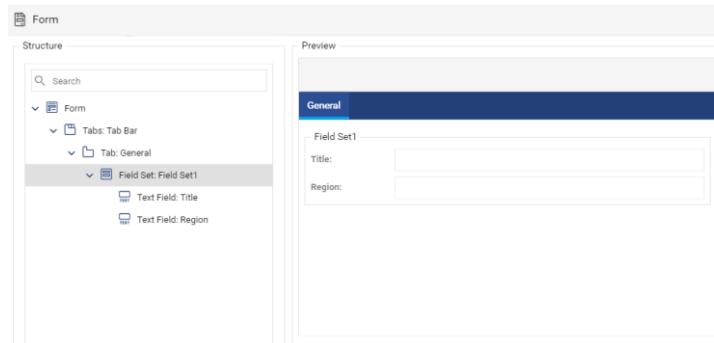
21. Add further tabs as required, being sure to highlight the tab bar before selecting **Add > Tabs**.

In addition to tabs, other layout tools are available and described below:

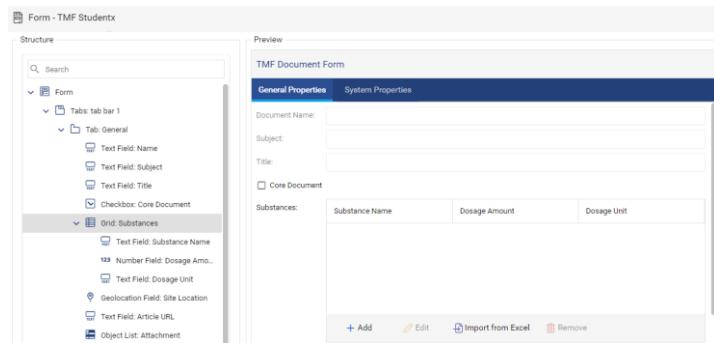
22. **Columns** place fields horizontally across the form. In the same way that tabs must be placed on a tab bar, columns must be placed on a column bar. Column bars and their columns can be placed directly onto the form or placed within other layout tools such as tabs:



23. **Field Set**, visually group fields together on the form by placing a border around them, indicating to users that the fields located within are related. Field sets can be placed directly onto the form or within other layout tools such as tabs:

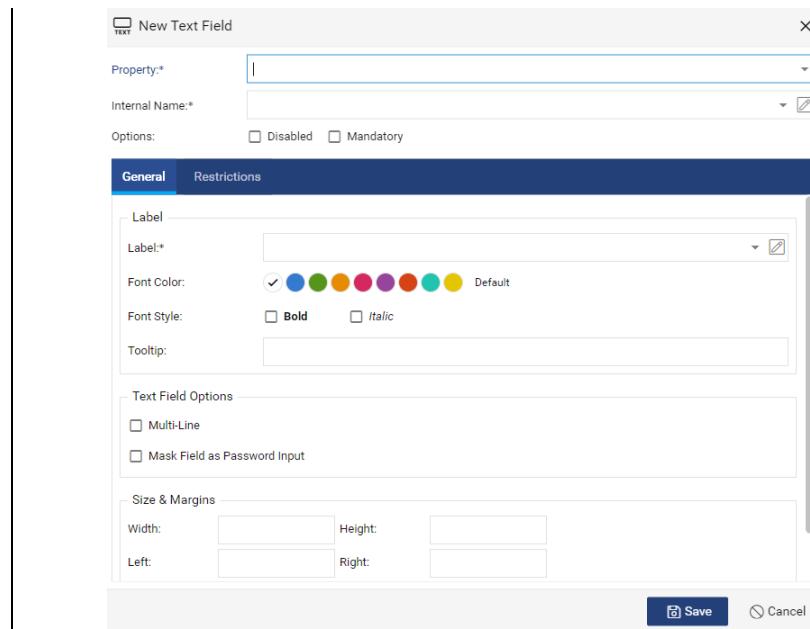


24. **Grid**. Grids are used to display sub-attributes of nested objects. For example, a document might have multiple **Authors** and each author could require their own sub-attributes such as **Name** and **Institution**. In this example Authors would be defined as a nested object attribute and Name and Institution would be defined as sub-attributes of such. A grid would be used to display the sub-attributes of authors:

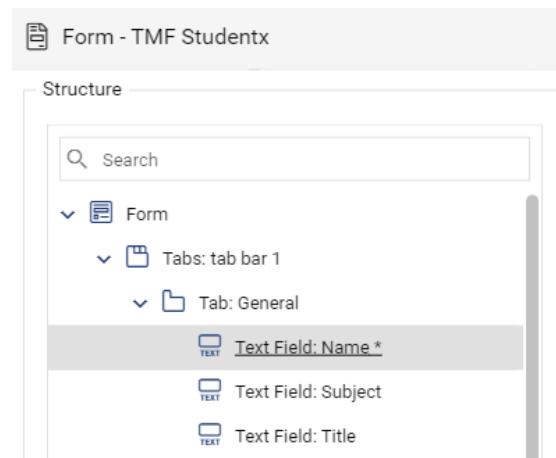


Within a grid users can change the order of rows via click and drag.

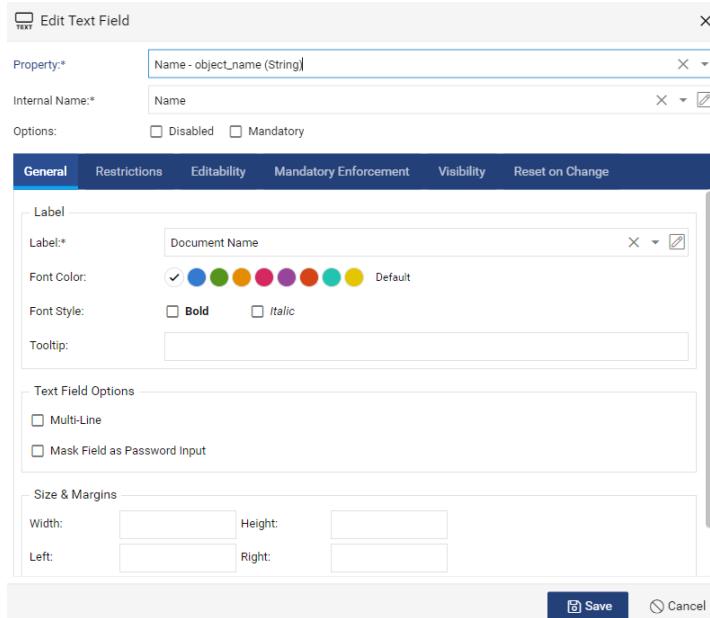
25. To add individual fields, first highlight the element the field should appear below such as a tab, field set or grid and then choose an appropriate attribute field type from the **Add** menu:



26. In the subsequent **New Field** window select the **Property** the field will be linked to and enter an **Internal Name** (which users will not see).
27. Tick **Disabled** if the field should be read-only. When adding system fields which cannot be modified by users the **Disabled** option is automatically selected.
28. Tick **Mandatory** if the field should be required. Fields which are mandatory are displayed on the form designer window with an asterisk and are underlined, such as for the **Name** field in the example below:



29. In the **General** tab:



30. Enter a user friendly **Label**, either manually or select from the `_display_labels` dictionary via the drop-down list.

31. Select a **Font Color** from the drop-down list.

32. For **Style** leave the default or tick **Bold** and/or **Italic**.

33. Enter a descriptive **Tooltip**, to be displayed when a user hovers their mouse over the field.

34. **Size and Margins** settings can be left as default or set as preferred.

35. **Restrictions** tab options are field type dependant, please see the separate section for details.

36. In the **Editability** tab:

The screenshot shows the 'Edit Text Field' dialog box. At the top, there are fields for 'Property:' (set to 'Name - object_name (String)') and 'Internal Name:' (set to 'Name'). Below these are 'Options' checkboxes for 'Disabled' and 'Mandatory'. A navigation bar at the top right includes tabs for 'General', 'Restrictions', 'Editability' (which is highlighted in blue), 'Mandatory Enforcement', 'Visibility', and 'Reset on Change'. Under the 'Editability' tab, there is a section titled 'Editable for Users/Groups' with a dropdown menu. Below it is a larger section titled 'Editability Condition' with a dropdown menu. At the bottom right of the dialog are 'Save' and 'Cancel' buttons.

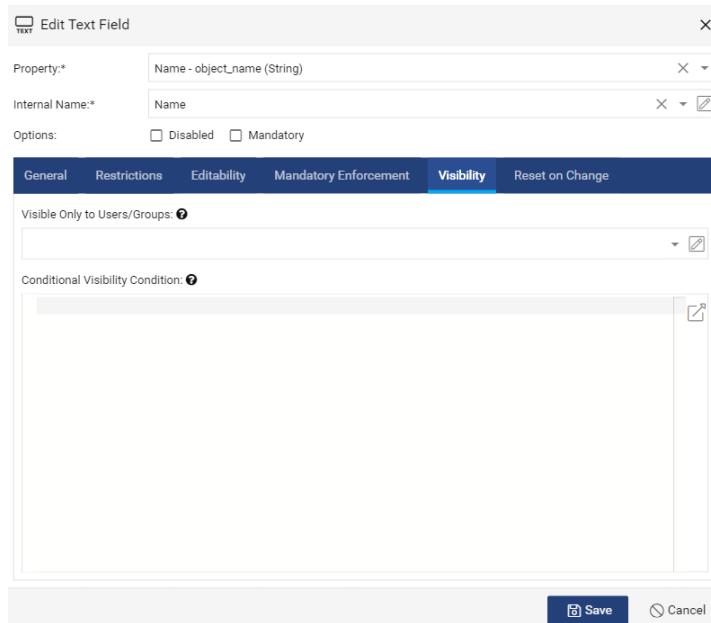
37. Optionally specify which **Users/Groups** and/or what **Condition** must be true for the property to be editable. The **Editability Condition** uses CARA Query Language syntax. If a condition is entered, the **Disabled** and **Mandatory** general options should not be enabled.

38. In the **Mandatory Enforcement** tab:

The screenshot shows the 'Edit Text Field' dialog box with the 'Mandatory Enforcement' tab selected (highlighted in blue). It has the same basic structure as the previous screenshot, with fields for 'Property:' and 'Internal Name:', and an 'Editability' tab. The 'Mandatory Enforcement' tab is active, showing sections for 'Mandatory to Users/Groups' and 'Mandatory Enforcement Condition', each with dropdown menus. The bottom right features 'Save' and 'Cancel' buttons.

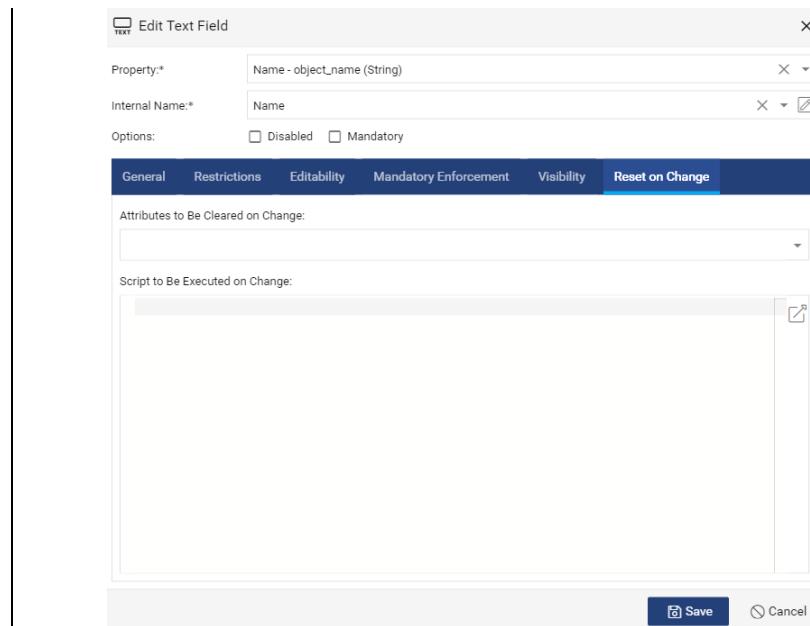
39. Optionally specify which **Users/Groups** are required to enter a value and a **Condition** for when the field must be completed. The **Mandatory Enforcement Condition** uses CARA Query Language syntax. If a condition is entered, the **Disabled** and **Mandatory** general options should not be enabled.

40. In the **Visibility** tab:



41. Optionally specify which **Users/Groups** and/or what **Condition** must be true for the property to be visible. The **Visibility Condition** uses CARA Query Language syntax. If a condition is entered, the **Disabled** and **Mandatory** general options should not be enabled.

42. In the **Reset on Change** tab:



43. Optionally specify attributes which should be cleared if a user changes the field's value. For example if a field named product was changed, it might be appropriate to clear a field named product_code.
44. Optionally enter a **Script to be executed** if a user changes the field's value.
45. **Save** the field configuration, returning to the main form builder window.

Field Control Types

The following field control types are available:

1. **Text Field.** Displays string attributes.
Text fields can be marked as **Multi-Line**.
If Multi-Line is selected, an additional option is provided to **Enable Opening in a Pop-up Dialog**.
Optionally select **Mask Field as Password Input** if characters entered in the field should not be visible when entered:

Property:

Internal Name:

Options: Disabled Mandatory

General [Restrictions](#) [Editability](#) [Mandatory Enforcement](#) [Visibility](#) [Reset on Change](#)

Label

Label:

Font Color: Default Blue Green Orange Red Purple Cyan Yellow

Font Style: Bold Italic

Tooltip:

Text Field Options

Multi-Line
 Mask Field as Password Input

Size & Margins

Width:
Height:
Left:
Right:

Buttons

[Save](#) [Cancel](#)

In the **Restrictions** tab:

A **Max Length** for values can be specified.

A **Mask** can be specified, forcing entered values to match a particular format. The mask uses the following format:

9 = [0-9], a = [A-Za-z], * = [A-Za-z0-9]

For example 99-999 for a postal code.

A **Regular Expression** can be specified, forcing entered values to match a particular format.

For example to check for a valid email address format:

`^[a-zA-Z0-9._-]+@[a-zA-Z0-9._-]+\$`

A **Regex Message** can be entered, to be displayed to users if the value entered is not valid.

Property: Name - object_name (String)

Internal Name: Name

Options: Disabled Mandatory

General Restrictions Editability Mandatory Enforcement Visibility Reset on Change

Max Length:

Mask: 9 = [0-9], a = [A-Za-z], * = [A-Za-z0-9] e.g. 99-999 for postal code

Regex Expression:

Regular Expression e.g. ^[a-zA-Z0-9_-]+@[a-zA-Z0-9_-]+\\$ for email validation

Regex Message:

Message Displayed When Value Is Not Valid

Save **Cancel**

2. **Label.** Used to display text on the form without a user input field. Label fields can display static text or a property name and value. When adding a label field, select whether **Static Text** or a **Property Value** will be displayed.

Label fields which display Static Text include both a text label and a text value. The text value field can accept HTML and property names as well as plain text:

ABC Edit Label

Display: Static Text Property Value

Internal Name: object name

General Alias Visibility

Label

Label: Document Name

Font Color: Default

Font Style: Bold Italic

Tooltip:

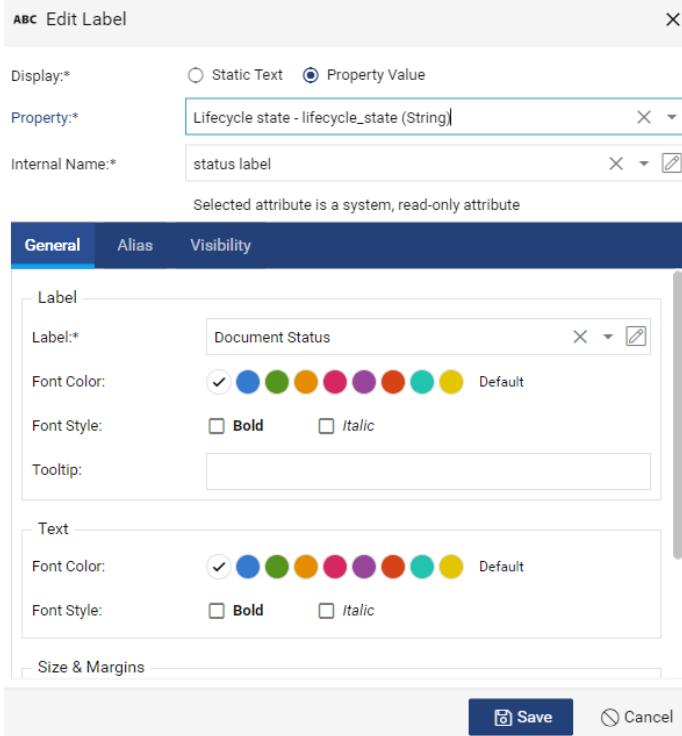
Text

Text: {object_name}

Font Color: Default

Save **Cancel**

When a label field displays a property, the **Property** is selected from an additional drop-down field:



3. **Number Field.** Display numeric attributes. A minimum and maximum value can be set as a restriction on the **Restrictions** tab.

The screenshot shows the configuration interface for a new number field named "123 New Number Field". The "Restrictions" tab is selected. Under "Min Value", there is a text input field containing "1" with up and down arrows for adjustment. Under "Max Value", there is a text input field with up and down arrows. A checkbox labeled "Display an Empty Value If the Value Is Set to Zero" is present. At the bottom right are "Save" and "Cancel" buttons.

4. **Checkbox.** Display Boolean attributes.
5. **Combo Box.** Provide a drop-down list of values to choose from as configured in the **Value Assistance** tab:

The screenshot shows the configuration interface for a new combo box named "New Combo Box". The "Value Assistance" tab is selected. Under "Type", the "Dictionary" option is selected. The "Dictionary" field contains "Country (country)". Under "Options", the "Asynchronous" and "Free Text Allowed" checkboxes are unchecked. Under "Min Autocomplete Characters", the value is set to "3". A note below states: "Minimum number of characters the user must type before the autocomplete is activated (default is 3)". At the bottom right are "Save" and "Cancel" buttons.

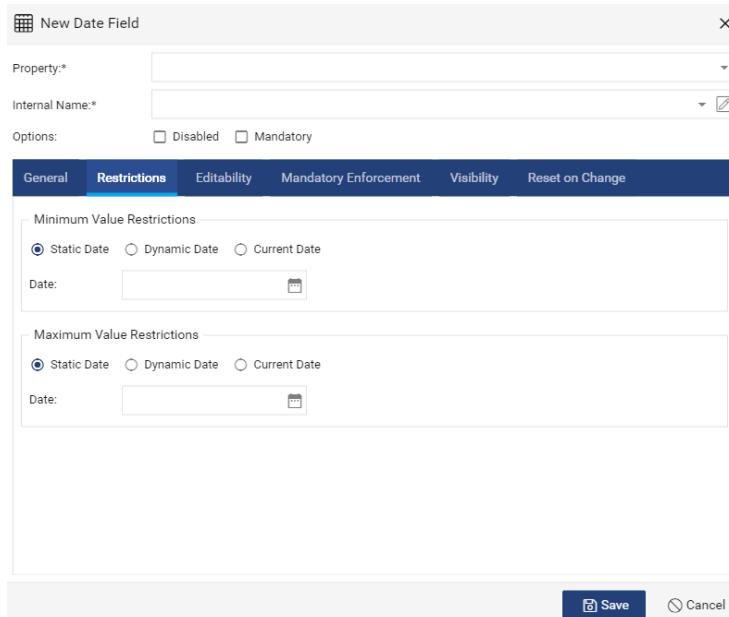
The drop-down list will most commonly be based on a **Dictionary**, however it can also be based on a **Taxonomy**, a manually entered **Value List**, a CQL **Query** or a **Script**.

Users can also be allowed to ignore the drop-down list values presented and instead enter free text.

Autocomplete is available to aid users when selecting from the drop-down list.

6. Date Field. Display date fields.

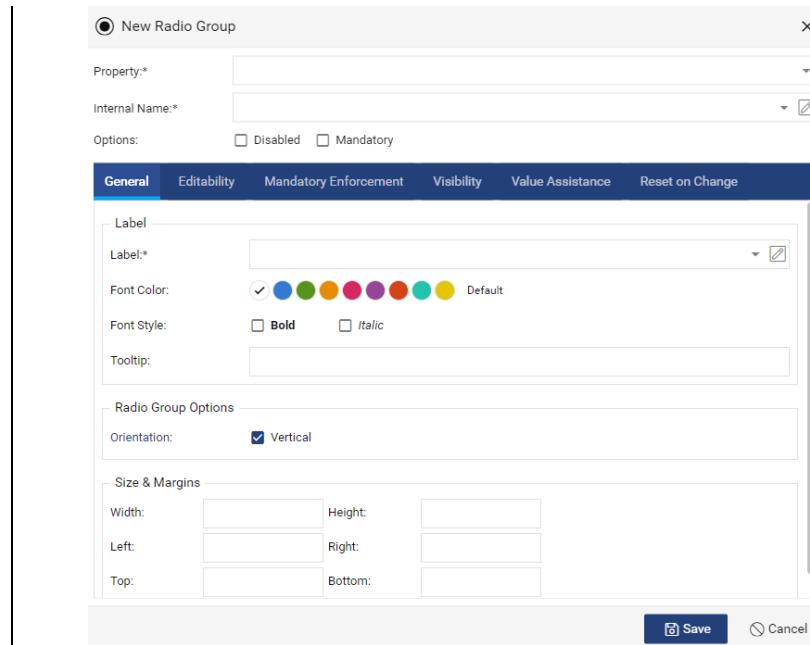
In the Restrictions tab **Minimum** and **Maximum** date values can be set, with an offset from a **Static Date**, **Dynamic Date** (based on another defined field) or the **Current Date**:



7. Date Time Field. Display date and time fields. The same restrictions are available as for Date Fields above.

8. Radio Group. Provide a list of values to choose from as configured in the **Value Assistance** tab. Unlike Combo Boxes there is no option for users to enter a free text value. Like Combo Boxes the values can be based on a **Value List**, **CQL Query**, **Dictionary**, **Taxonomy** or **Script**:

Radio buttons can be orientated horizontally (the default) or **Vertically**:



9. **Check Box Group.** Display multi-value fields. Users can select more than one value. Value assistance can be based on a **Value List, CQL Query, Dictionary, Taxonomy or Script**:

10. **Dual List Field.** Display multi-value fields. Users can select more than one value from the drop-down list presented. Value assistance can be based on a **Value List, CQL Query, Dictionary, Taxonomy or Script**.
Users can also be allowed to ignore the drop-down list values presented and instead enter **Free text**:

New Dual List Field

Property:*

Internal Name:*

Options: Disabled Mandatory

General	Restrictions	Editability	Mandatory Enforcement	Visibility	Value Assistance	Reset on Change
Type: <input type="radio"/> Value List <input type="radio"/> Query <input checked="" type="radio"/> Dictionary <input type="radio"/> Taxonomy <input type="radio"/> Script	Dictionary: Country (country)	Dictionary Alias:	Options: <input type="checkbox"/> Asynchronous <input checked="" type="checkbox"/> Free Text Allowed	Min Autocomplete Characters: 3	Minimum number of characters the user must type before the autocomplete is activated (default is 3)	

11. Multi-text Field. Display multi-value fields. Users can enter more than one value in a free text field.

12. Tag Field. Display multi-value fields. Users can enter more than one value in a free text field. Users can select more than one value from the drop-down list presented.
Value assistance can be based on a **Value List**, **CQL Query**, **Dictionary**, **Taxonomy** or **Script**.
A **Minimum and Maximum Number of Values** can be set:

New Tag Field

Property:*

Internal Name:*

Options: Disabled Mandatory

General	Restrictions	Editability	Mandatory Enforcement	Visibility	Value Assistance	Reset on Change
	Min Number of Values: 1	Max Number of Values: 4				

- 13. Object List.** Display the properties of one or more related objects on the form. Requires a *multi-value* ID (Object Reference) attribute type.

For example, an email and its attachments could be imported into CARA as separate documents but reference each other, listing their name and other attributes.

Users are provided with options to browse for and select the related objects within the form. They can also create and import new related objects directly within the form.

In the **Columns** tab, the attributes of the related objects to be displayed must be selected, the object list cannot be saved until at least one column/attribute has been added:

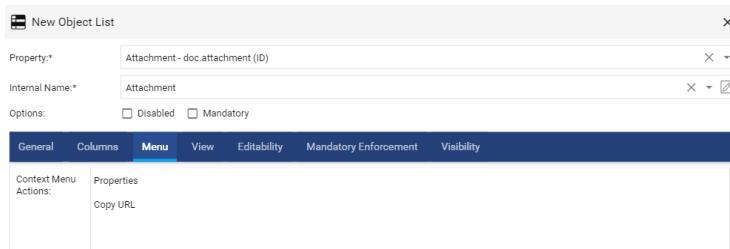
Attribute	Label	Alignment	Width	Visible?	Sortable?	Cell Styles	Cell Actions
Name (object_name)	Name	Left	100	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Created (created)	Created	Left	100	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Creator (creator)	Creator	Left	100	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

In the **View** tab, a view must be selected, which is used to decide the layout of the window within which users can Select related documents. Options for users to Create, Import, Select Existing and Remove related documents can also be enabled:

View Name*	TMF Author View (tmf_studentx)
Allowed Operations:	<input type="checkbox"/> Create <input type="checkbox"/> Import <input checked="" type="checkbox"/> Select Existing <input checked="" type="checkbox"/> Remove
View Filter When Selecting Existing:	<input type="text"/>
Select Button Label:	<input type="text"/>
Select Dialog Header:	<input type="text"/>

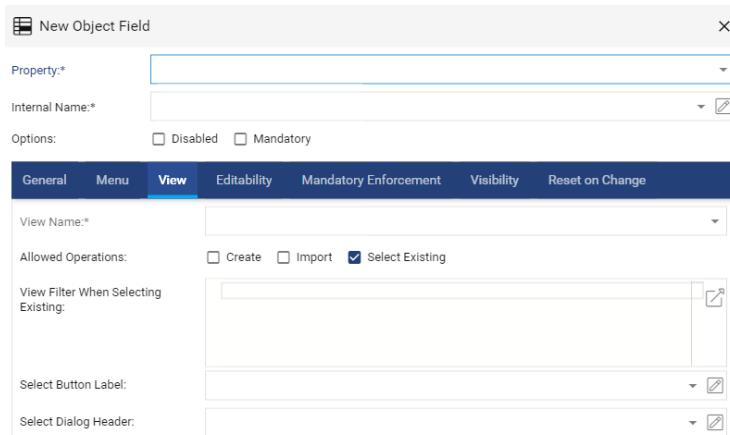
When **Create** or **Import** a new object is enabled, an additional option to create a mapping between the properties of the new object and the current object is provided. Select a **New Object Property** and a corresponding current **Object Property** and select **Update**. The option to rename the Create button is also provided.

In the **Menu** tab, menu options available to users in the right-click context menu can be provided, allowing users to view and edit related documents from within the form.



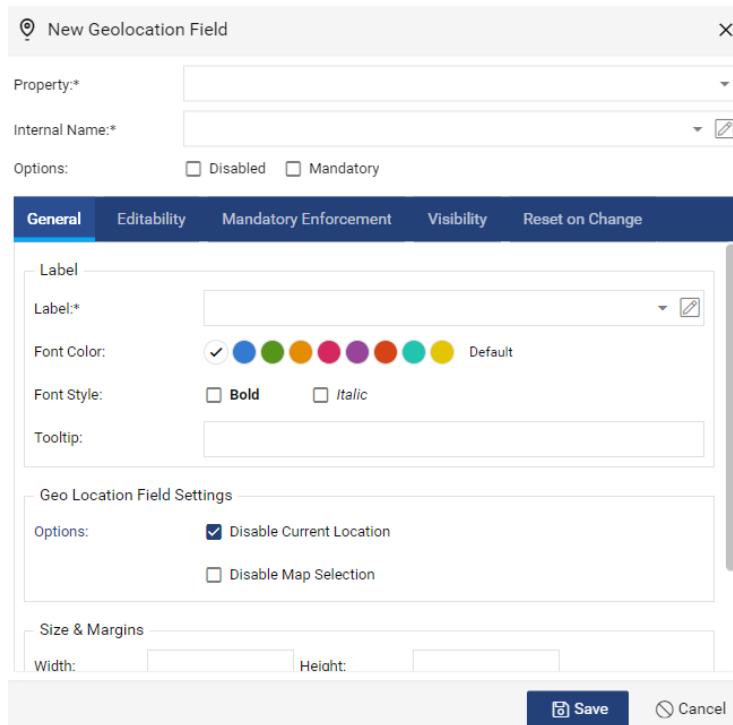
14. Object Field. Display the properties of a single related object.

Requires a *single-value ID* (Object Reference) attribute type. The concept and main configuration options are the same as for the Object List field control above, but users are limited to only selecting a single related object.

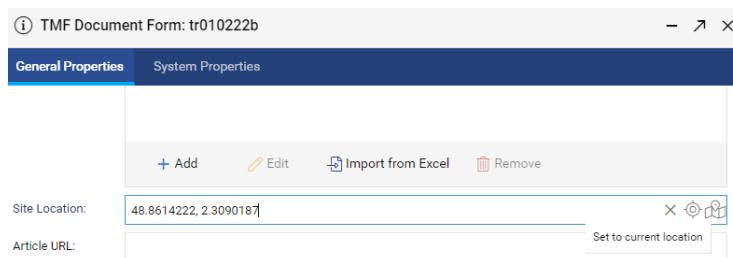


15. Value Picker. Display a report within the form which can map variables from the selected object to the report. The report must have been configured in advance.

16. Geolocation. Specifies a geographical location in latitude and longitude with a comma separator, for example 51.483, -0.604. A geolocation value is used by map filter panels.



When placed on a form, the field provides options to auto-populate with the user's location (based on their IP address) and to open a separate map window where the user can click a location to generate a geopoint value.



These options can be removed by ticking **Disable Current Location** and **Disable Map Selection** respectively.

17. User/Group Field and User/Group List. Allows selecting existing users or groups as a value.

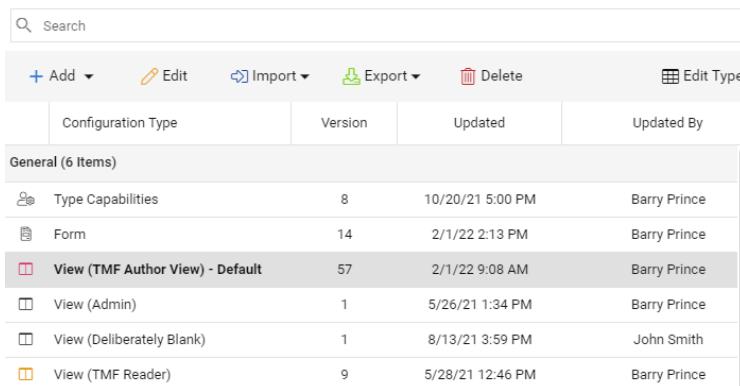
Use **User/Group Field** for single values and **User/Group List** for multiple values. Suitable for use with user/group based default attributes such as creator and modifier (these are referred to as Authority fields), as well as custom attributes of type user/group reference.

The screenshot shows a configuration dialog titled "New User/Group Field". The "Restrictions" tab is active. Under "Allow Selecting", the "Only Users" radio button is selected. Under "Restrict to Groups", a dropdown menu is open, showing a list of groups. At the bottom right are "Save" and "Cancel" buttons.

Editing of the field can be restricted to selected groups.

7.3. Views - General

Description	<p>Views determine how the user interface appears to users and includes fundamental options such as which attributes are displayed, which widgets are available, the filters that users can apply and how searches behave.</p> <p>Views are specific to an object or document type, ensuring the view is tailored to the attributes defined, and other properties of the type. Each object or document type can have multiple views, the first view for a type is labelled as the default.</p> <p>Different groups of users within an organisation (which commonly represent different business roles or departments) can be provided with different views of the same object or document object.</p> <p>Views are made available to groups of users within workspaces and a workspace may include views from multiple object types. The Type Configuration panel displays all of the configured views for a type and their Version, date last Updated and who their configuration was last Updated By:</p>
--------------------	---



The screenshot shows a software interface for managing views. At the top, there's a search bar and several action buttons: '+ Add', 'Edit', 'Import', 'Export', 'Delete', and 'Edit Type'. Below this is a table with four columns: 'Configuration Type', 'Version', 'Updated', and 'Updated By'. The table lists six items under the heading 'General (6 Items)'. The rows show the following data:

Configuration Type	Version	Updated	Updated By
Type Capabilities	8	10/20/21 5:00 PM	Barry Prince
Form	14	2/1/22 2:13 PM	Barry Prince
View (TMF Author View) - Default	57	2/1/22 9:08 AM	Barry Prince
View (Admin)	1	5/26/21 1:34 PM	Barry Prince
View (Deliberately Blank)	1	8/13/21 3:59 PM	John Smith
View (TMF Reader)	9	5/28/21 12:46 PM	Barry Prince

View configuration is divided into the following areas, which the subsequent chapters describe and detail:

General. View display options and which groups will have access to the view.

Actions. Toolbar and context menu action buttons.

Aggregations. User selectable filters available in the left-hand side navigation panel.

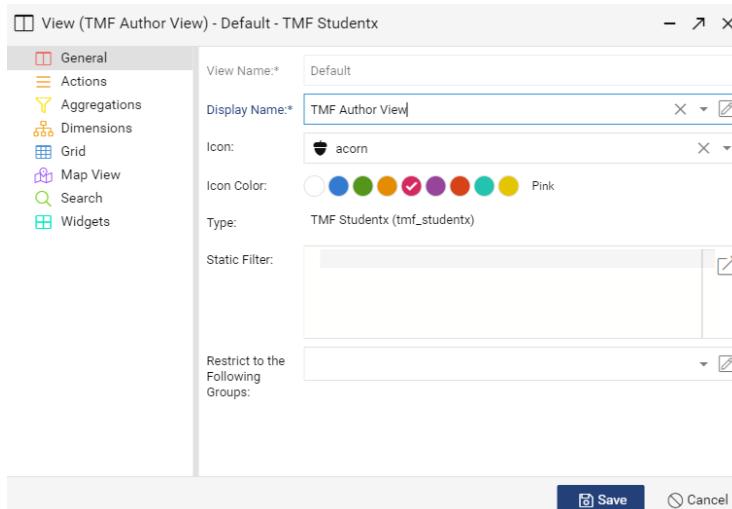
Dimensions. The document and folder hierarchy, browsable by users and grouped by user selectable properties.

Grid. Which attributes are displayed as columns in the main view panel, their sort order and row style etc..

Map View. Configuration of the mapping feature.

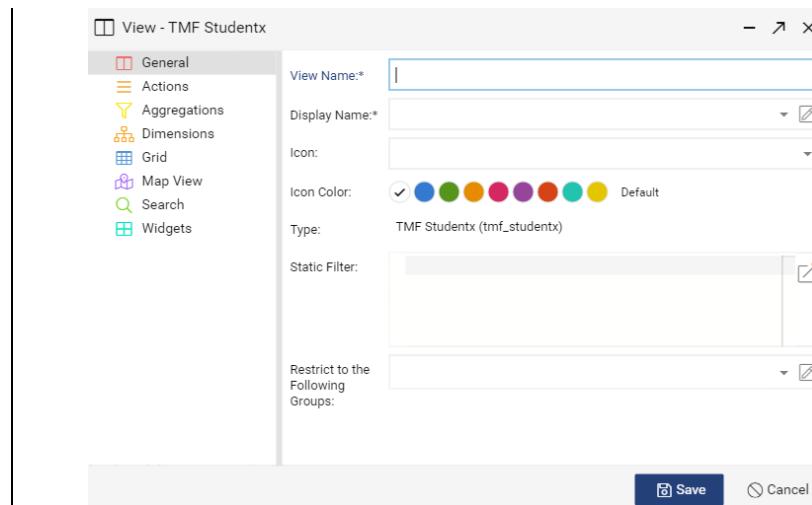
Search. Search behaviour, version filters and tags.

Widgets. Select and configure the widgets displayed in the right-hand panel.



Creating Views

1. Within the CARA Control Panel click on a type within the **Type Configuration** section.
2. Within the **Type Configuration** panel select **Add > General > View**, the **View** window opens:



3. Enter a **View Name**.

4. Enter a **Display Name**.

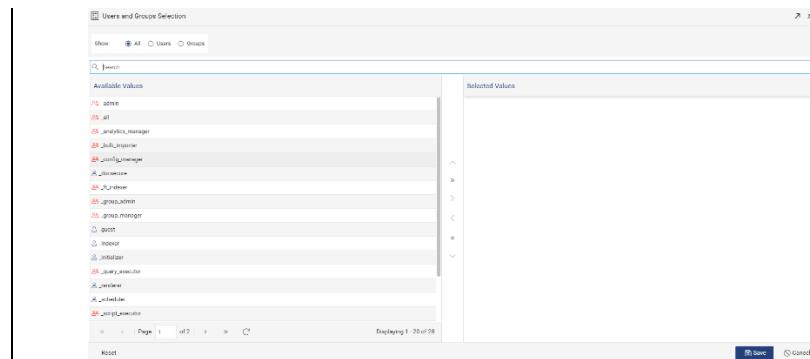
5. Select an **Icon** from the drop-down list.

6. Select an **Icon Color**.

7. Optionally enter a **Static Filter**. Static filters remove documents from the view and are based on CQL syntax. Static filters override any other filter configuration within the view and the ACL permissions of the individual users. Static filters provide a simple method of ensuring a view only includes appropriate documents, regardless of other configuration.

8. **Restrict** the view to specified groups and/or users as appropriate. If no groups or users are specified the view is available to all users, however the view is only visible when included within a workspace which the user has access to.

Select users and groups directly from the drop down list or in the separate selection window that is opened by clicking the pencil icon.

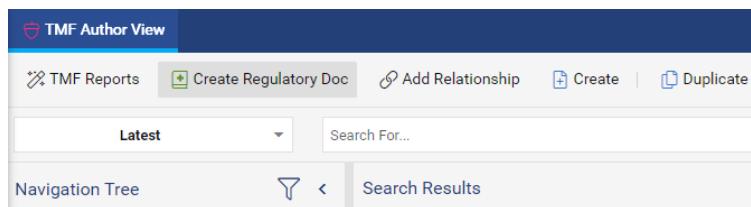


9. **Save** the updated view configuration.

7.4. View - Actions

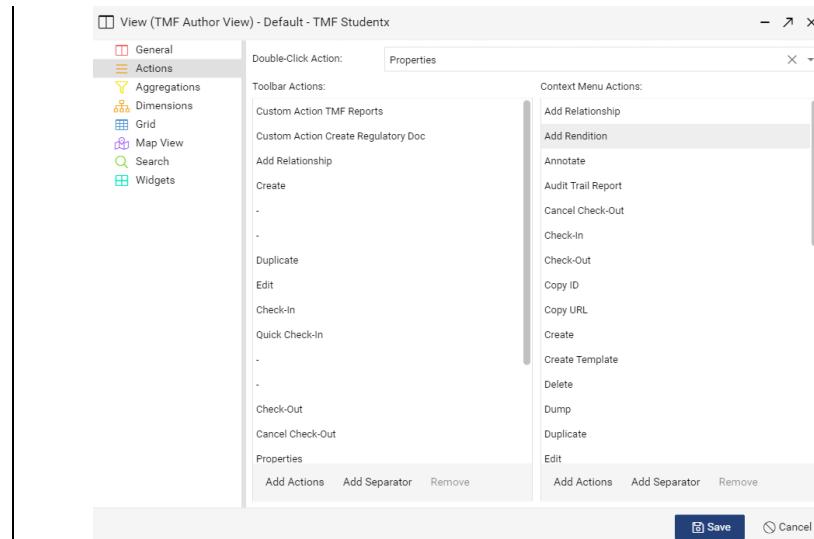
Description	Actions refers to the toolbar and context menu actions available to users. In addition to standard document management actions such as viewing or editing documents, custom actions can also be made available to users within a view. Actions can be placed in a preferred order and separators can be added to visually group actions together within the user interface. The double-click action can also be defined.
--------------------	--

Example main toolbar with action buttons defined:

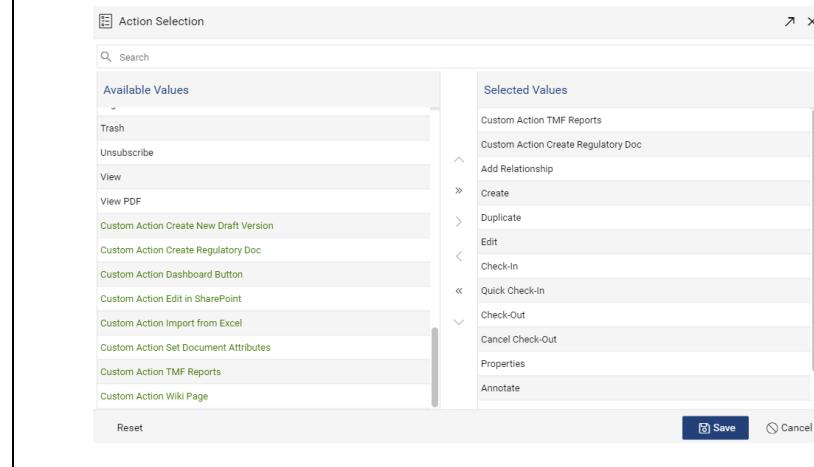


Defining Actions for a View

1. Within the CARA Control Panel click on a type within the **Type Configuration** section. Either create a new View by selecting **Add > General > View**, or highlight and **Edit** an existing view.
2. Within the **View Configuration** window click **Actions**, the Actions panel opens:



3. Select a **Double-Click Action** from the top drop-down list.
Custom actions can be chosen, they are grouped together at the end of the scroll bar. It is recommended selecting an action which does not result in checking-out a document, such as edit, in order to avoid users accidentally locking a document if they unintentionally double-click.
4. Select **Toolbar Actions** in the left-hand panel by clicking **Add Actions** to open the Action Selection window. Custom actions are grouped together at the bottom of the scroll list and colour coded green.

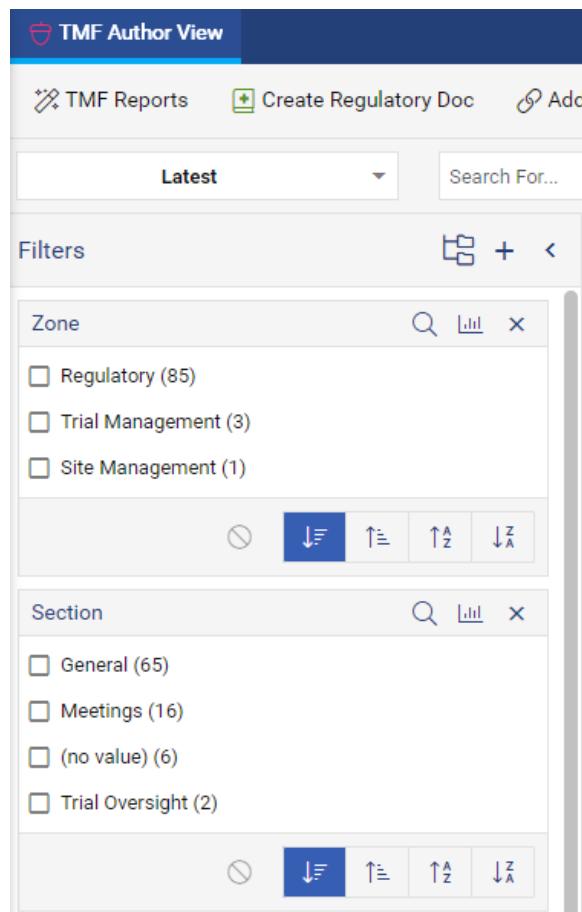


5. Select **Context Menu Actions** in the right-hand panel by clicking **Add Actions**. Custom actions are grouped together at the bottom of the scroll list.
6. Separators can be added to both Toolbar Actions and Context Menu Actions. Highlight the item the separator is to be below and click **Add Separators**.
7. Actions can be reordered by clicking and dragging within the Actions window.
8. Click **Save** once finished.

7.5. View - Aggregations

Description Aggregations are user selectable filters which can be applied to the documents listed in the Main View panel. Aggregations are selectable in the left-hand Navigation panel.

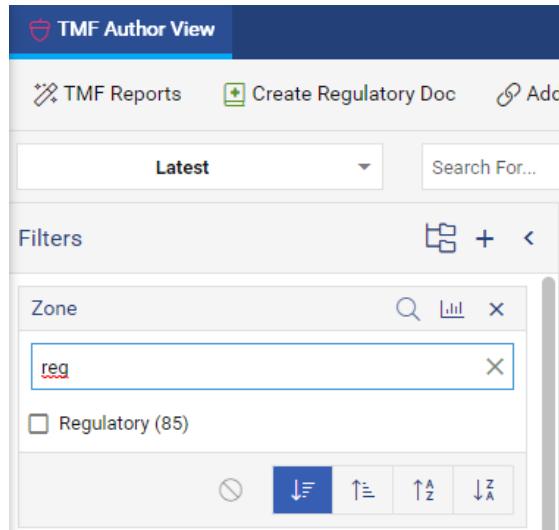
Users can select filter values from auto-populated drop-down lists or free type values. Filter results can be displayed in a graph format.



Users toggle between displaying filters (funnel icon) or the folder hierarchy (network icon, detailed in later chapter).

Users can tick one or more of the automatically generated filter values, or click the magnifying glass icon to open the **Filter Values** field and enter text (complete or partial term).

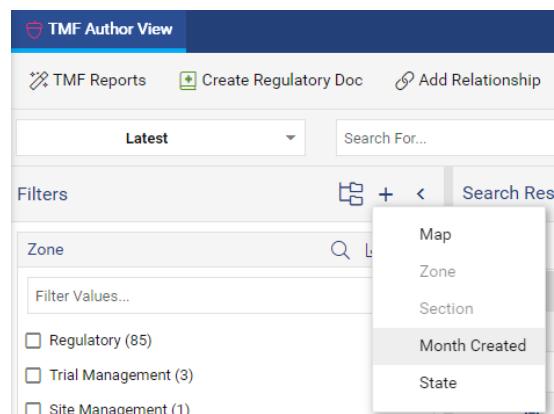
There is no need to press Enter in the **Filter Values** field, the search engine will match results as entries are being typed. Wild card characters do not need to be entered into the **Filter Values** field:



The number in brackets indicates how many documents have the corresponding value.

Users can click to choose the sort order from the icons at the bottom of the **Filters** panel.

Where multiple filters have been made available, users can add them to the Navigation panel by clicking the **+** plus icon:



By clicking the graph icon, the results will be displayed in a new window within the Main View panel as a bar graph.



Creating Aggregation Filters.

1. Within the **Control Panel**, having selected a type in the **Type Configuration** section, select a **View** and then click **Aggregations** to open the **Aggregations** panel.
2. Click **Add**, the **Add Aggregation** window opens:

3. Select the **Attribute** that the filter will be based on from the drop-down list.
4. Select a **Type**, either **Terms** or **Date Histogram**. Date histogram is only available for date fields. When selecting Date Histogram additional options will appear within the window.

5. In the **Header** field manually enter a user friendly label for the filter or select one from the display_labels_ dictionary via the drop-down list.
6. Select an **Aggregation Order**, either **Count** or **Key**.
7. Enter an **Aggregation Size** or leave the default of 100.
8. **Integer** attributes have an additional option, **Metric**. Metric can either be set to **Document count**, which is the standard aggregation behaviour showing the number of documents listed with each value, or alternatively for each value show the **Sum of Attribute Values**.
9. If **Date Histogram** has been selected as the **Type**, then **Interval** is displayed as an additional option. Select either **Year** or **Month**.
10. If **Date Histogram** has been selected as the **Type**, then **Format** is displayed as an additional option. Enter the format required such as mm-yyyy.
11. Select a **Display Size**.
12. Choose a **Display Order**.
13. Click **Save** to return to the **Aggregations** panel.
14. The order the aggregation filters are displayed to users can be changed via click and drag:

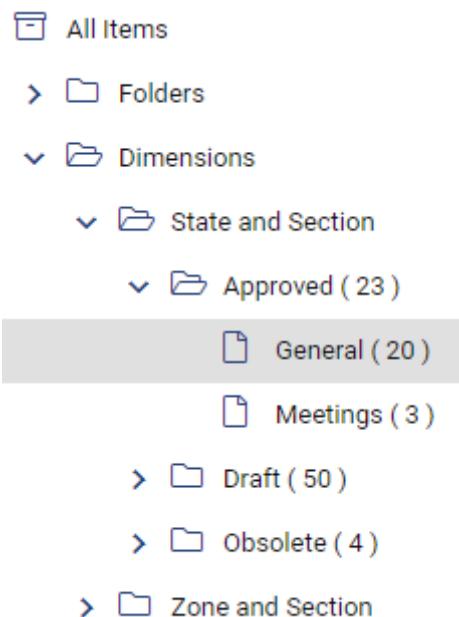
Type	Attribute Name	Header
Terms	Zone (doc.zone)	Zone
Terms	Section (doc.section)	Section
DateHistogram	Drag and Drop to Reorganize	Month Created
Terms	Lifecycle state (lifecycle_state)	State

15. **Save** the aggregations configuration, returning to the view configuration window.

7.6. View - Dimensions

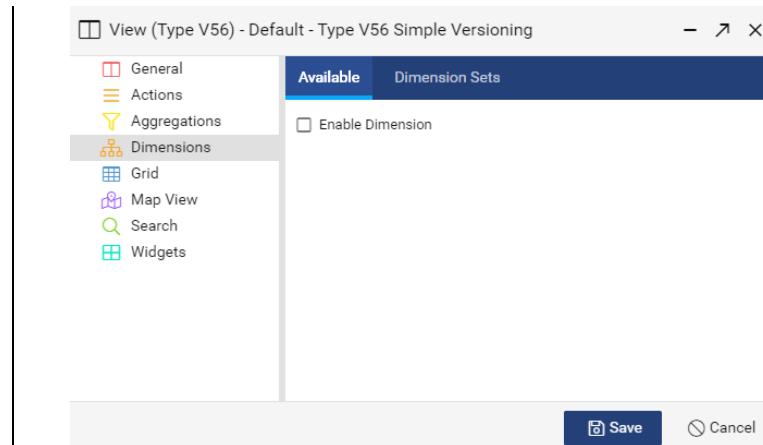
Description Dimensions provide an alternative view of the folder hierarchy which users can browse through. A dimension's folder hierarchy can be grouped by properties according to user preference, for example users might want to browse a folder hierarchy with documents grouped by their region or product and then sub-grouped by status.

Dimensions are displayed in the Navigation panel, beneath the main folder hierarchy. The number in brackets beside each folder is the number of documents within.

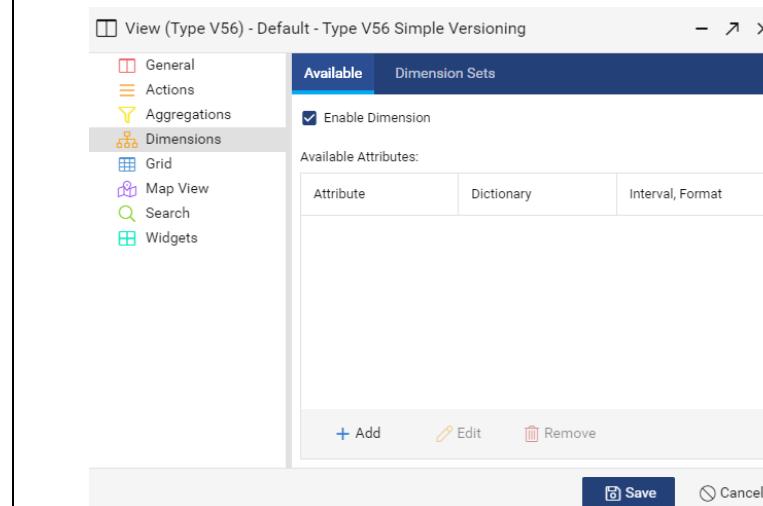


Creating Dimensions

1. Within the **Control Panel**, selected a type in the **Type Configuration** section and then select and **Edit** an associated view. In the view configuration window select **Dimensions**:



2. Tick to **Enable Dimensions**, the **Available Attributes** panel is displayed, where the attributes which can be used to group documents by are listed. Initially no attributes are listed and therefore no dimensions can be defined.



3. Click **Add** and in the subsequent **New Available Dimension** window select an **Attribute**.

When selecting string attributes, options to select a **Dictionary** and **Dictionary Alias** are provided:

New Available Dimension

Attribute:*	Region - doc.region (String)	X ▾
Dictionary:	Creator - creator (Authority)	
Dictionary Alias:	Deletion date - deleted (DateTime)	
	Country - doc.country (String)	
	Region - doc.region (String)	
	Subject - doc.subject (String)	

When selecting date attributes, choose an **Interval** and enter a date display **Format**:

New Available Dimension

Attribute:*	Created - created (DateTime)	X ▾
Interval:*	Month	X ▾
Format:*	mm-yyyy	X

Save Cancel

- Click **Save** to return to the dimensions panel and **Add** further attributes as required.

Available Dimension Sets

Enable Dimension

Available Attributes:

Attribute	Dictionary	Interval, Format
Created (created)		Month, mm-yyyy
Region (doc.region)		
Country (doc.country)		

+ Add Edit Remove

- In the **Dimensions Sets** tab click **Add** to open the **New Dimension Set** window:

New Dimension Set

ID:*	region_country
Display Name:	Documents by Region and Country
Level 1:*	doc.region
Level 2:	doc.country
Level 3:	
Level 4:	
Level 5:	
Level 6:	
Level 7:	
Level 8:	
Level 9:	
Level 10:	

Save Cancel

6. Enter an **ID (name)** for the dimension set.
7. Enter a **Display Name**.
8. Select the **Level 1** attribute, the attribute by which the top level of folders will be grouped by. Only attributes which have been defined in the **Available** tab can be selected.
9. Select an attribute for each additional level, up to 10, as required.
10. Click **Save**, returning to the **Dimensions** window and then click **Save** once more.

Dimension Sets		
ID	Display Name	Levels
region_country	Documents by Region and Country	doc.region, doc.country

Add **Edit** **Remove**

Save **Cancel**

7.7. View - Grid

Description

Grid refers to the column layout for displaying document attributes in the Main View panel, located in the centre of the user interface. The properties to be displayed need to be selected, with user friendly labels added. Size, alignment and format can also be set.

	Name	Created	Version label	Modified ↓
	Training MS Word Template 4	10/21/21 2:20 PM	0.1, LATEST, Obsolete	1/18/22 11:04 AM
	My New Doc	1/14/22 2:31 PM	1.0, LATEST, Approved	1/14/22 4:40 PM
	TR140122a	1/14/22 11:35 AM	1.0, LATEST, Approved	1/14/22 11:40 AM
	TR0112a Lorem Ipsum Document	1/18/22 11:16 AM	0.2, LATEST, Draft	1/18/22 11:16 AM
	Training Document 2110a	10/21/21 2:56 PM	1.0, LATEST, Approved	10/21/21 3:55 PM
	Test Student14	10/20/21 6:46 PM	1.0, LATEST, Approved	10/21/21 3:02 PM
	My New Doc	10/21/21 2:21 PM	0.1, LATEST	10/21/21 2:21 PM

The default column layout for the view is set by administrators, however users are allowed to alter the column layout and change which attributes are displayed.

Editing the Grid

1. Within the **Control Panel**, having selected a type in the **Type Configuration** section, select a **View** to **Edit** and then click the **Grid** icon.

Attribute	Label	Alignment	Width	Visible?	Sortable?
Name (object_n...	Name	Left	250	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Created (creati...	Created	Center	167	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Version label (ve...	Version label	Left	200	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Creator (creato...	Created By	Center	150	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Modified (modifi...	Modified	Center	150	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Modifier (modifi...	Modified By	Center	150	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Content Type DO...	Content Type D...	Left	100	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Buttons at the bottom: + Add, + Add multiple, Edit, Remove, Sort By: modified, Ascending, Descending, Page Size: 50, Save, Cancel.

- The automatically generated default view for a type contains a set of default attributes, whereas additional views are initially blank.
2. The order of columns can be changed via click and drag.
 3. Whether individual columns are **Visible** or **Sortable** can be toggled on and off by clicking within the corresponding cell.
 4. Select an attribute to **Sort By**, in **Ascending** or **Descending** order.
 5. Enter a **Page Size**, the number of rows displayed per page. The total browsing limit for documents is independent of the page size and is 10k.
 6. The **Row Styles** tab allows document attribute rows to be displayed in different styles based upon a condition, for example documents with status='approved' could be in green with documents with status ='obsolete' in red.
To create a row style click **+ Add** and enter a **Condition** using CARA Query Language format, and a **Style Name**.

Columns	Row Styles	Row Icons
Condition	Color	Italic Bold
lifecycle_state='Approved'	Green	
lifecycle_state='Obsolete'	Red	

7. The **Row Icons** tab allows the document icons displayed in the grid to be customised and based on conditions. Different icons can be displayed when a document is unlocked or locked.

Click **Add** and in the **Add Icon** window enter a **Condition**, select an **Icon** and an **Icon Colour**.

The screenshot shows the 'Row Icons' configuration screen. At the top, there are tabs for 'Columns', 'Row Styles', and 'Row Icons'. The 'Row Icons' tab is selected. Below the tabs, there are two sections: 'Custom Format Icons:' and 'Custom Lock Icons:'. In the 'Custom Format Icons:' section, there is a table with one row. The row contains a condition 'document_type='Clinical'' and icons for 'Icon' (a green circle with a white dot) and 'Bullet'. At the bottom of this section are buttons for '+ Add', 'Edit', and 'Remove'. In the 'Custom Lock Icons:' section, there is a similar table with one row, but it is currently empty. There are also '+ Add', 'Edit', and 'Remove' buttons at the bottom.

8. Once the grid properties have been updated click **Save**.

1. To add attributes as columns individually click **+Add**, opening the **Add Column** window:

Add Column - View (TMF Author View) - Default - TMF Studentx

General	Cell Styles & Actions
Attribute:* <input type="text" value=""/> Label:* <input type="text" value=""/> <input checked="" type="checkbox"/> Visible? <input type="checkbox"/> Sortable? Width: <input type="text" value="100"/> <input type="button" value="X"/> Alignment: <input checked="" type="radio"/> Left <input type="radio"/> Centre <input type="radio"/> Right Format: <input type="text" value=""/> Template: <input type="text" value=""/>	

 2. In the **General** tab select an **Attribute** from the drop-down list.
 3. Select a user friendly **Label** from the _display_labels dictionary drop-down list or enter one manually.
 4. Select if the column will be **Visible**. Columns which are ticked as visible are displayed by default. Columns which are added but not ticked as visible are not displayed initially but can be selected by users.

5. Select if the column will be **Sortable**.
6. Enter a **Width** in pixels or leave the default value. Users can resize columns.
7. Select whether the column **Alignment** will be **Left**, **Centre** or **Right**.
8. Optionally specify a **Format**. The format selected must be appropriate to the attribute type. The following format options are available:

Integer format syntax with examples based on 123456.789:

0 (123457) show only digits, no precision.
0.00 (123456.79) show only digits, 2 precision.
0.0000 (123456.7890) show only digits, 4 precision.
0,000 (123,457) show comma and digits, no precision.
0,000.00 (123,456.79) show comma and digits, 2 precision.
0,0.00 (123,456.79) shortcut method, show comma and digits, 2 precision.
0.##### (123,456.789) Allow maximum 4 decimal places, but do not right pad with zeroes.
0.00## (123456.789) Show at least 2 decimal places, maximum 4, but do not right pad with zeroes.

***SIZE** Display in a user friendly data format, where the value is in bytes, for example a value of 2000 becomes 2 KB.

***TIME_S** Display in a user friendly time format, where the value is in seconds, for example a value of 75 becomes 1m 15s.

***TIME_M** Display in a user friendly time format, where the value is in minutes, for example a value of 75 becomes 1h 15m.

Link. String attributes only. Displays as a web link.

mailto. String attributes only. Displays a link that opens the user's email program to send a new message.

Uppercase. String attributes only. Displays in all uppercase.

Percent. Displays as a percentage, such as 99%.

Tel. Displays a link with behaviour that varies with device capabilities:

Cellular devices can autodial a number.

Many operating systems have programs that can make calls, such as Skype.

Websites can make phone calls with the registerProtocolHandler, such as web.skype.com.

Other behaviours include saving the number to contacts or sending the number to another device.

9. Optionally specify a **Template**. Templates are strings that contain HTML tags and placeholders, enclosed by {}.
- Placeholders can contain attributes or expressions such as {attribute:format}.

Example: {object_name:uppercase}

The screenshot shows the configuration interface for a column named "object_name". The "General" tab is selected. The "Attribute:" field is set to "Name - object_name (String)". The "Label:" field is set to "Name". Under "Visible?", both checkboxes for "Visible?" and "Sortable?" are checked. The "Width:" field is set to 250. The "Alignment:" field has "Left" selected. The "Format:" field contains a question mark icon. The "Template:" field contains the expression "{object_name:uppercase}".

The same format options are available as for the formats field. Where both the format and template fields are used, the template field has priority.

10. In the **Cell Styles & Actions** tab conditions can be created for how the cell should be displayed and an icon with an on-click action can be selected. Note that cell styles can also be applied to the whole row as described previously in this section.

Condition	Color	Italic	Bold
lifecycle_state='Approved'	Green		
lifecycle_state='Obsolete'	Purple		

Condition	Icon	On-click Action
lifecycle_state='Draft'	<input type="checkbox"/>	Annotate

11. To add a **Cell Style**, click **Add** and in the **Cell Style** window enter a CQL condition and select the **Font Color** and **Font Style**:

Condition: lifecycle_state='Approved'

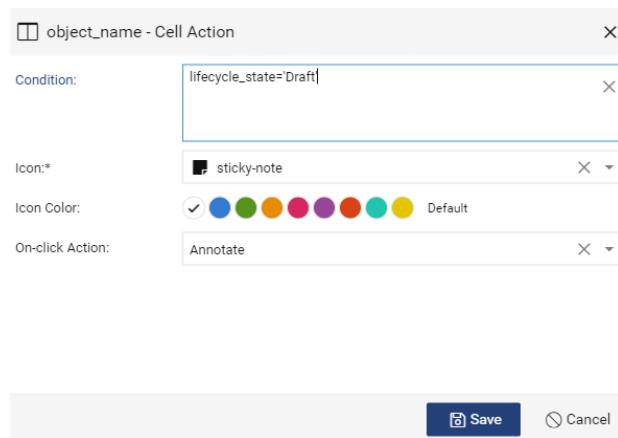
Font Color: Green

Font Style: Bold Italic

Save **Cancel**

12. To add a **Cell Action**, click **Add** and enter a CQL condition.

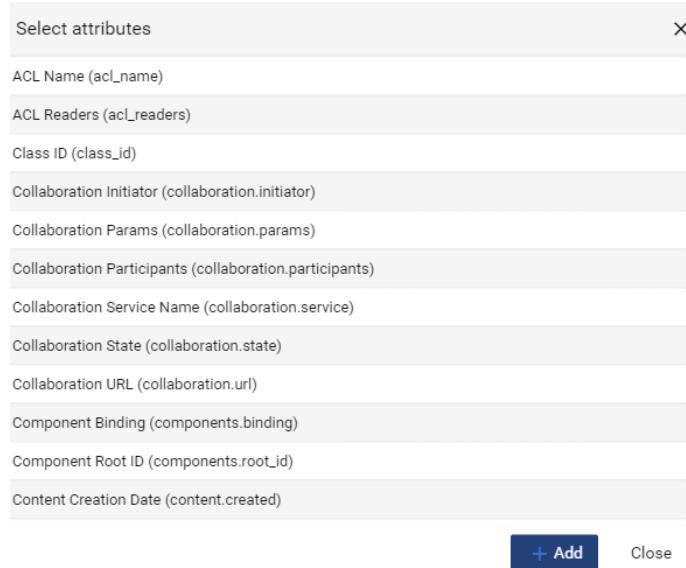
Select an **Icon** and optionally select an action to be executed when the icon is clicked:



13. Click **Save**, returning to the **Columns** tab. Add further columns as required.

Adding Multiple Columns

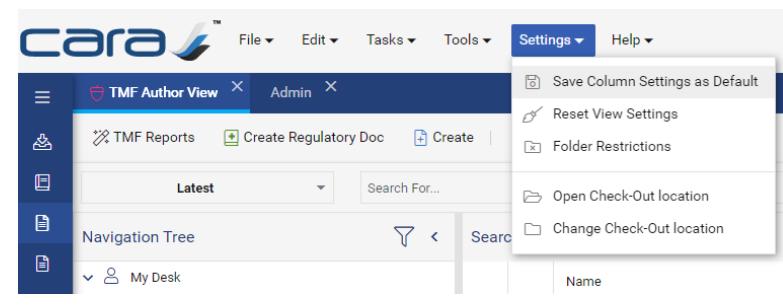
In addition to adding columns individually, multiple columns can be selected and added with a single action using the **Add Multiple** option. In the subsequent **Select Attributes** window hold the Shift or Ctrl keys to muti-select:



Updating Column Settings from the Main UI

Columns added in this way have the default settings applied initially and once added can be updated individually as required.

The order, width and visibility of columns in a view grid can be modified and saved directly within the main user interface by users who belong to the _admin or _config_manager groups. Modify the view's columns as required within the main user interface window and then select **Settings > Save Column Settings as Default:**



7.8. View - Map View

Description

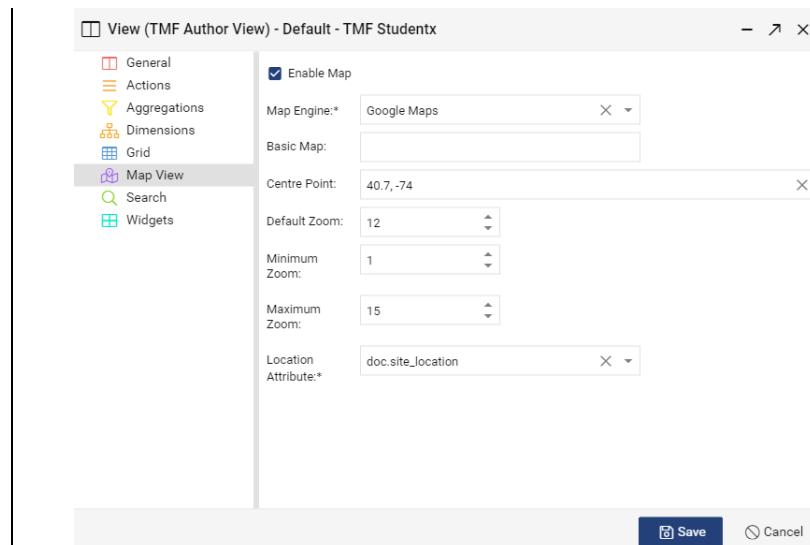
Map search allows users to visually map objects based on geographical attribute data. Initially users select **Show Map** in the **Filters** mode of the Navigation panel:

The map then expands to display a geographical representation of the currently selected objects based on a defined mapping attribute:

Map search involves passing a specified mapping attribute to either Google Maps or ESRI ArcGIS, and is configured as part of a view.

Configuring Map Search

1. To define mapping options click **Map View** within view configuration:

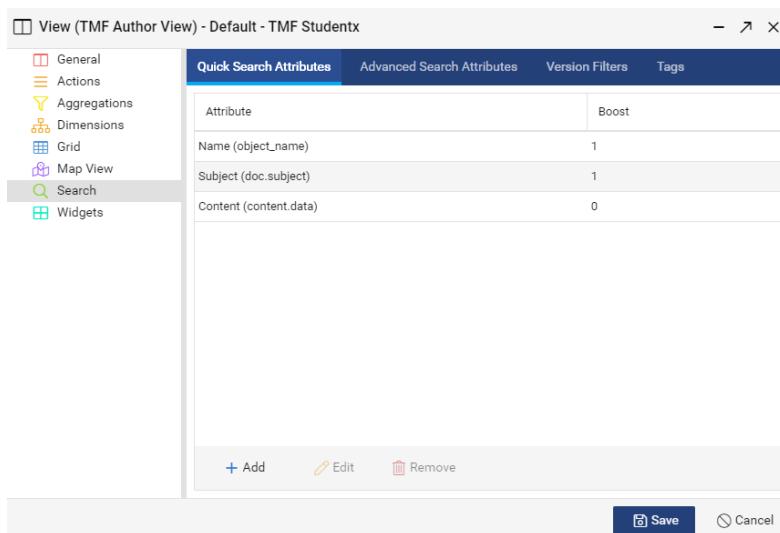


2. Click **Enable Map**.
3. Select a **Map Engine**, either **Google Maps** or **ESRI ArcGIS**.
4. **Basic Map**. Enter the name of the default map type: roadmap, satellite, hybrid.
5. **Centre Point**. A centre point for the map can be entered as longitude and latitude coordinates, for example 51.499190, -0.198008.
6. **Default Zoom**. Enter a default zoom level for the map.
7. **Minimum Zoom**. Enter a minimum zoom level for the map.
8. **Maximum Zoom**. Enter a maximum zoom level for the map. The largest value accepted is 24.
9. **Location Attribute**. Select a geolocation attribute of the type.
10. **Layers**. Layers do not apply to the Google Maps service, they are an optional feature of the ESRI ArcGIS service.
11. Click **Save**, returning to the **View** configuration window.

7.9. View - Search

Description Search configuration determines the behaviour of quick search and advanced search. It is also where version filters and tags are enabled and configured.

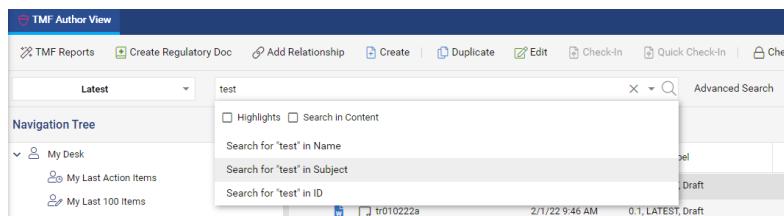
Search Configuration Within the **Control Panel**, having selected a type in the **Type Configuration** section, select a **View** and then click **Search** to open the **Search** panel:



The Search panel is divided into 4 tabs: Quick Search, Advanced Search Attributes, Version Filters and Tags. Each is separately detailed in the following sections.

Quick Search Attributes The **Quick Search** bar is located at the top of the main view panel. Quick search is both case insensitive and a contains operator search, users do not need to enter wildcard characters. Where multiple attributes are included within quick search, users can search within all defined attributes or select a specific attribute from the drop-down list.

The attributes which are included in a quick search must be defined, no attributes are included by default.



To define the attributes included within Quick Search, from the **Quick Search** tab click **Add**. Select an **Attribute** from the drop-down list and a **Boost** amount, then click **Update**. Add other attributes as needed and **Save** changes.

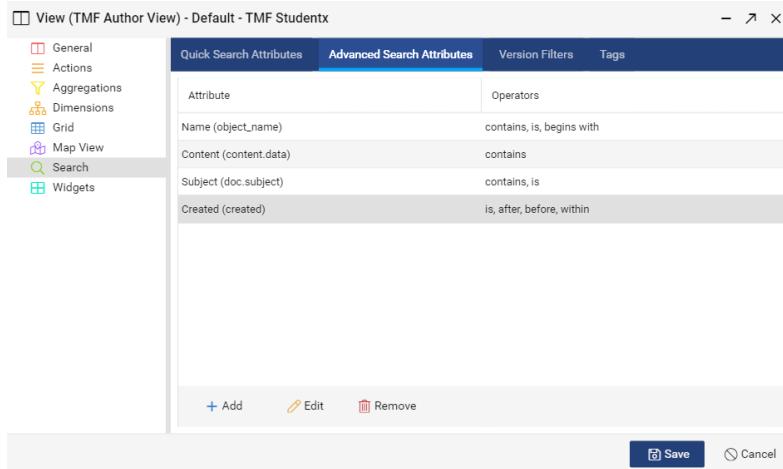
In order for users to be able to search within document content (full-text search) the attribute **content.data** must be added:

Attribute	Boost
Name (object_name)	1
Subject (doc.subject)	1
Content (content.data)	0

Advanced Search Attributes

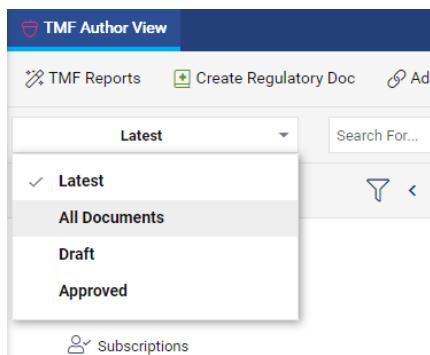
The advanced search window allows users to search multiple attributes and select the operators for each attribute. Administrators configure which attributes and operators are selectable, none are available by default:

To add attributes to the advanced search click **+Add**. Select the **Attribute** and **Operators** from the respective drop-down lists and **Update**. Add further attributes as appropriate and **Save**:

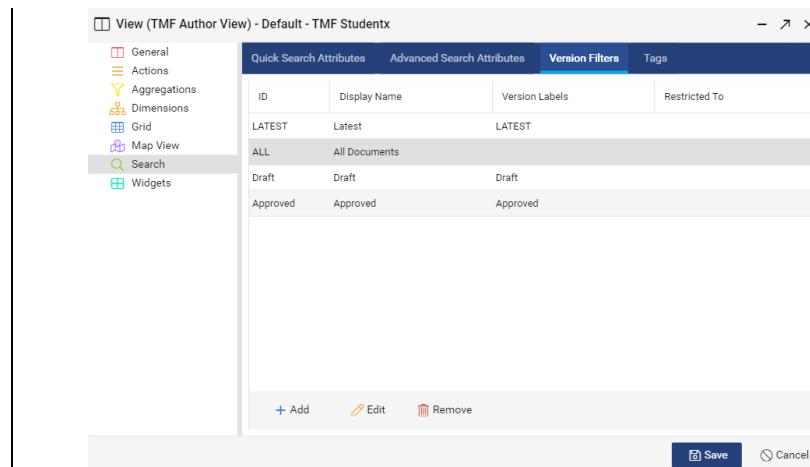


Version Filters

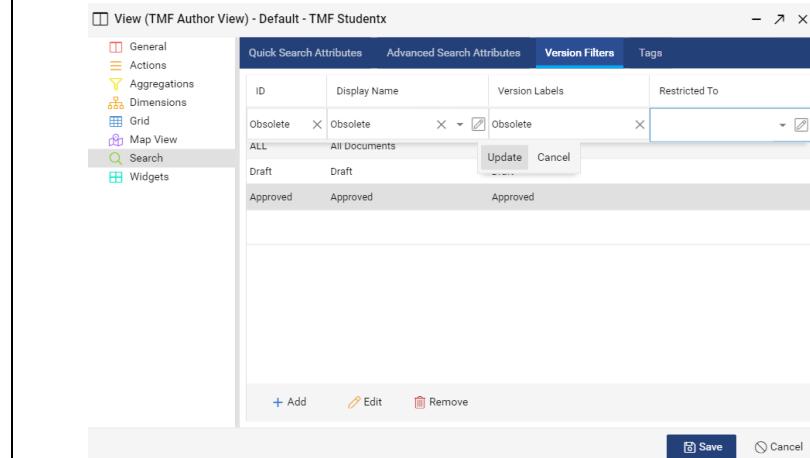
The version filters panel is displayed in the top left corner of the user interface and allows users to narrow the list of documents displayed in the Main View panel by a version label value such as Latest or Current. Users can toggle between the defined version filters and viewing **All Documents**.



1. To define version filter options click the **Version Filters** tab:



2. **Latest** and **All** version filters are present by default, they can be modified or removed as appropriate.
3. To add additional version filters click **Add** and enter an **ID** (internal name), **Display Name** and the **Version Labels** to be displayed. Optionally restrict the version filter to a specified group. Select **Update** to save the filter.



4. The order of version filters within the configuration window is also the order displayed to users. The order can be changed by individually clicking and dragging the version filters.
5. Click **Save**.

Tags

Tags are additional filters which can be based on any attribute. Tags are displayed in the Navigation panel with a dedicated node. They are similar to the global filters feature of previous versions of CARA, where users select from pre-configured filters. Private tags can be created by users and public tags can be pre-configured by administrators:

Name	Type
tr2101d	
tr010222a	
tr010222b	
tr010222c	
tr310122	
tr2401a	
tr2101g	
tr2101b	

Creating Tags

1. To create public tags click the **Tags** tab.
2. Click **Add** and enter an **ID** (internal name), **Description** (user friendly label) and **Condition** using CARA Query Language syntax, then click **Update**:

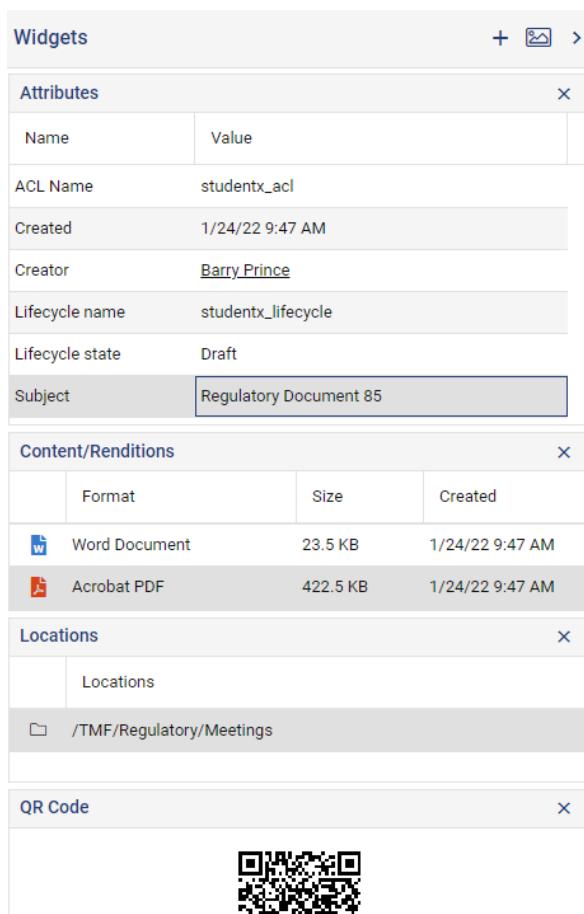
ID	Display Name	Condition
mydocs	My Documents	creator=user
mylocked	My Locked Documents	

3. **Add** additional tags as required and then click **Save**.

7.10. View - Widgets

Description

Widgets provide detailed information about the document currently highlighted in the Main View panel. The Widgets panel is located on the right-hand side of the user interface and can be minimized and resized as per user preference. Users can resize and close individual widgets can change their order through drag and drop. The Widgets panel is also used to display the document preview and there is an icon for switching between the two modes.



A set of standard widgets are available out of the box. Widget configuration involves deciding which widgets will be displayed to users by default and which can be additionally selected by users.

Standard widgets:

Attributes

Configuring Widgets – General Options

- Bar Code
- Content/Renditions
- Details
- QR Code
- Thumbnail
- Versions

Each widget type can have a user friendly label defined. Some widgets, such as versions, have additional configuration options.

1. Within the **Control Panel**, having selected a type in the **Type Configuration** section, select a **View** and then click **Widgets** to open the **Widgets** panel:

ID	Header	Type
attributes	Attributes	Attributes
content	Content/Renditions	Content/Renditions
locations	Locations	Locations
qr	QR Code	QR Code
versions	Versions	Versions
details	Details 2	Details
relations	Relations	Relations

Add **Edit** **Remove**

Save **Cancel**

2. Click **Add** to open the **New Widget** window:

Type:*

ID:*

Header:*

Attributes (attributes)

Content/Renditions (content)

Details (details)

Locations (locations)

QR Code (qr)

Relations (relations)

Thumbnail (thumbnail)

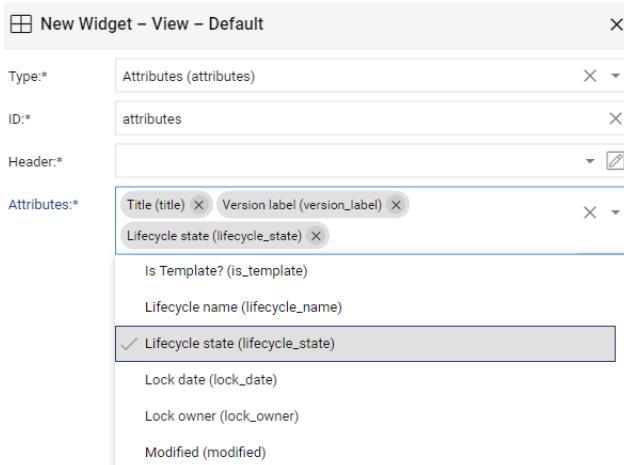
Versions (versions)

Save **Cancel**

3. Select the **Type** of widget to be added.
4. Enter an **ID** (system name).
5. Enter a user friendly label in the **Header** field or select one from the _display_labels dictionary drop-down list.
6. **Save** the widget configuration, returning to the **Widgets** panel.
7. **Add** other widgets as required and **Save**.

Widget Specific Options

Attributes widget. Select which attributes will be displayed from the Attributes drop-down list:



Content/Renditions widget. Optionally enter a **Hidden Format**, that is rendition formats which will not be displayed to users within the widget. This is useful if there are rendition formats being generated automatically via rendition services which are not intended to be viewed by end users.

The screenshot shows a configuration dialog titled "content – Widget – View – Default". It includes fields for Type (Content/Renditions), ID (content), Header (Content/Renditions), and Hidden Format. At the bottom are "Save" and "Cancel" buttons.

Details widget. In the **Template** field enter a HTML – X Template that should be displayed in the widget.

The screenshot shows a configuration dialog titled "New Widget – View – Default". It includes fields for Type (Details (details)), ID (details), Header, and Template. At the bottom are "Save" and "Cancel" buttons.

QR Code widget. Select if the QR code should link to the document, the primary content or the PDF content. Also select whether the QR code points to the specific version selected or always the latest version:

qr - Widget - View - Default

Type: QR Code

ID: qr

Header:*

Link To:

- Document
- Content - Primary
- Content - PDF

Link to Versions:

- Selected
- Latest

Save **Cancel**

Relations widget. Select which attributes are displayed and whether Child and Parent relations are shown. Each attribute can have a specific cell style and action defined, as described in the View - Grid section of this manual.

relations - Widget - View - Default

Type: Relations

ID: relations

Header:*

Related Type: TMF Studentx (tmf_studentx)

Relation Type: SOP

Include:

- Child Relations
- Parent Relations

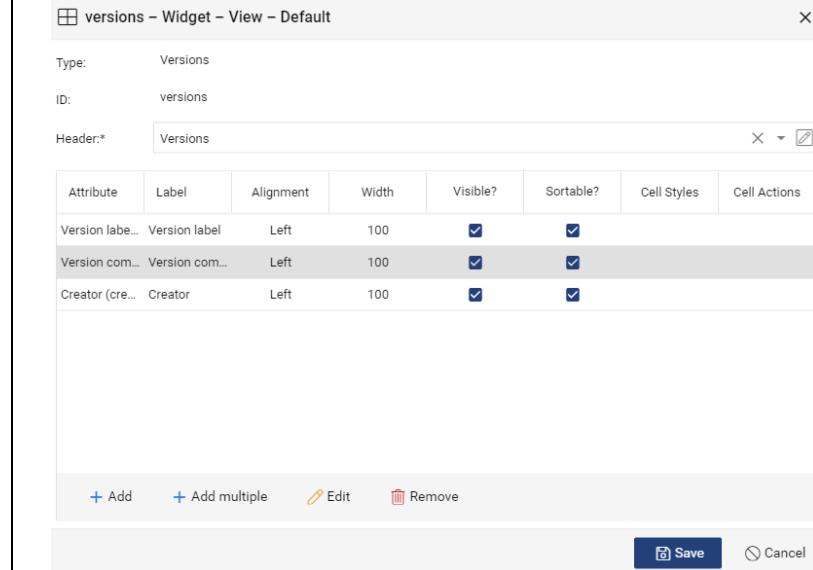
Attribute	Label	Alignment	Width	Visible?	Sortable?	Cell Styles	Cell Actions
Name (obje...	Name	Left	100	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Version labe...	Version label	Left	100	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Modified (m...	Modified	Left	100	<input checked="" type="checkbox"/>	<input type="checkbox"/>		

Add **Add multiple** **Edit** **Remove**

Save **Cancel**

Versions widget. Select the attributes to be displayed within the widget, these are typically version specific attributes such as version label. Click **Add**, select an **Attribute** and enter a user friendly **Label** or select one from the _display_labels dictionary drop-down list.

Optionally enter a column **Width** in pixels or leave the default and choose an **Alignment** option.



Each attribute to be displayed can have a specific cell style and action defined, as described in the View - Grid section of this manual.

7.11. Classification Definition

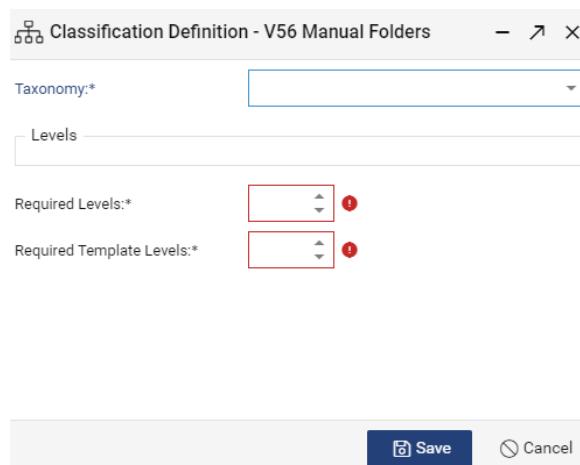
Description	<p>Classification definition involves specifying the taxonomy from which users select a classification when creating new documents or objects. Users can be presented with up to 5 levels of choices, the number of levels is determined by the selected taxonomy.</p> <p>Not all levels within the taxonomy have to require a user to make a selection, some levels may be set as optional, for example users might be required to select a department that the document applies to at the top level, but a secondary level for region could be left as optional if some documents apply across regions.</p> <p>Each classification choice that a user makes has to be stored as a distinct attribute value for the document, for example a user chooses a document's department and region and those choices are stored in the corresponding attributes.</p> <p>In the example below, presented to users when creating a new document or importing a document, users are required to select both a classification Zone and Section. Both choices are required, as indicated by the asterisk beside each field.</p>
--------------------	--

The screenshot shows the 'Create' dialog box with the following interface elements:

- Classification** section:
 - Zone: * (selected value: Regulatory)
 - Section: * (selected value: General)
- Creation Method** section:
 - Create: Placeholder From Template From eForm
- Buttons**: Next > and Cancel

Configuring the Classification Definition

1. Within the **Control Panel**, having selected a type in the **Type Configuration** section, select **Add > Initialisation > Classification Definition**.
2. Each type only has a single classification definition, the menu option to add is removed after the first has been defined.
3. In the **Classification Definition** window select the defining **Taxonomy**.



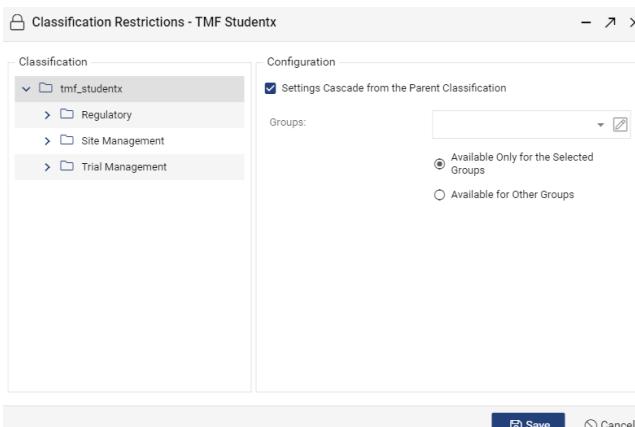
4. The window will automatically populate the **Levels** panels with the number of levels defined in the taxonomy.
5. For each level select an attribute where the choice made for each individual document will be stored. Typically for each level an identically named attribute is selected, though this is a convention and not a requirement. For example users might select a department which is stored in an attribute named Department and a region which is stored in an attribute named Region.
6. Select the number of **Required Levels** for documents, that is how many levels down users are required to make a choice when creating or importing documents. Choices beyond that number will be optional.

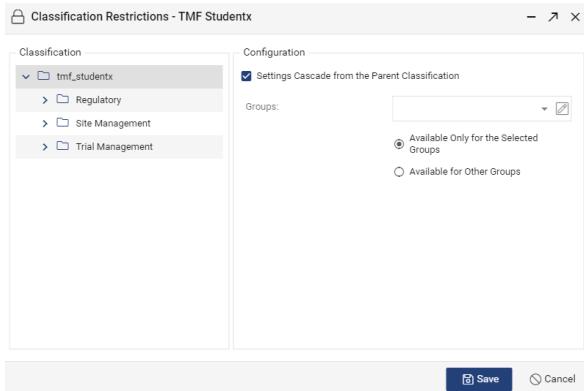
7. Select the number of **Required Template Levels**. This can be different from the number of required levels for documents. For example while documents might always apply to a specific department and region, templates might be department specific but be useable across regions.

The screenshot shows the 'Classification Definition - TMF Studentx' window. At the top, it displays 'Taxonomy:' and 'Studentx Main Classification'. Below this, under 'Levels', there are two sections: 'Level 1 Attribute:' containing 'doc.zone' and 'Level 2 Attribute:' containing 'doc.section'. Under 'Required Levels:', the value is set to 2. Under 'Required Template Levels:', the value is set to 1. At the bottom right, there are 'Save' and 'Cancel' buttons.

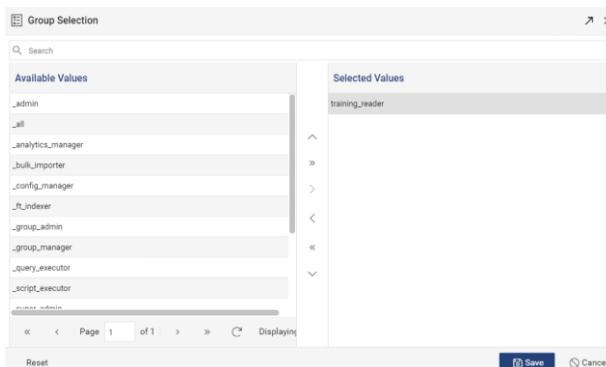
8. Click **Save** and return to the **Type Configuration** window.

7.12. Classification Restrictions

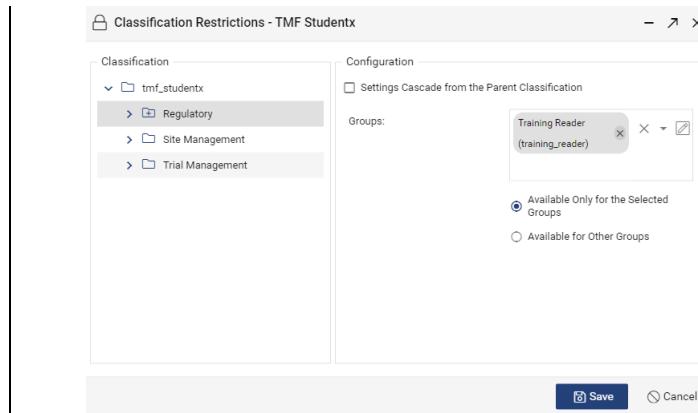
Description	<p>Classification restrictions determine which groups of users can create which document or object types.</p> <p>The Classification Restrictions window automatically populates with the defining taxonomy and updates accordingly if changes are made to the taxonomy.</p> <p>In common with other Initialization configuration elements, a specific classification restriction can be created for each node of the main classification, or nodes can inherit the configuration of their parents. This makes it is easy to create a default configuration that applies to all documents, and yet still create unique configuration for specific document types where required.</p>
Defining Classification Restrictions.	 <ol style="list-style-type: none"> Within the Control Panel, having selected a type in the Type Configuration section, select Add > Initialisation > Classification Restrictions. It should be noted that each type only has a single classification restriction configuration definition, the menu option to add is removed after the first has been saved. The Classification Restrictions window opens, displaying the object type at the root and the main classification taxonomy cascading below. No restrictions are set initially.



3. To set a restriction, highlight a folder/node and untick **Settings Cascade from the Parent Classification**, the right-hand side **Configuration** panel can then be updated.
4. In the **Groups** field on the right-hand side of the window, use the drop-down arrow to select groups directly or click the pencil icon to open the **Group Selection** window and then locate and select groups, adding them to the right-hand panel.



5. Click **Save** to return to the **Classification Restrictions** window:



6. Underneath the **Groups** field, set whether the restriction should be **Available Only for the Selected Groups** or **Available for Other Groups**.
7. Note that the folder icon for the selected node now includes a plus icon, indicating that it has a restriction set.
8. Drill down through the main classification as required, specifying further group access as required for each node.
9. Once complete **Save** the configuration.

7.13. Lifecycle Assignment

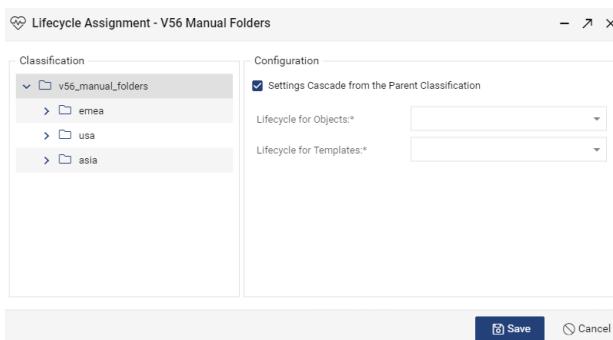
Description

Lifecycles can be applied to a document automatically when created or imported. Each node of the main classification can have a different lifecycle assigned, or inherit the assignment from its parent classification.

Different lifecycles can be assigned to documents and templates. The lifecycles to be assigned must be created in advance in the Lifecycles section of the control panel.

Assigning a Lifecycle

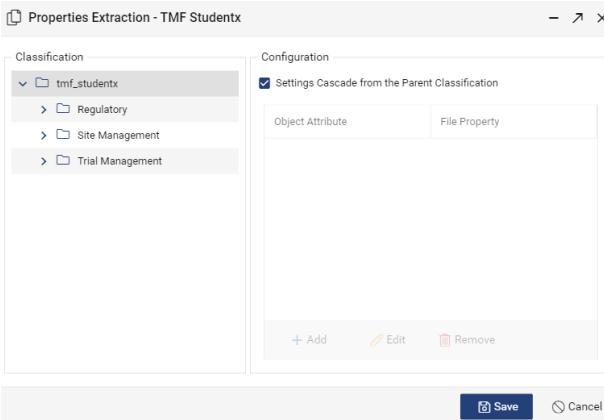
1. Within the **Control Panel**, having selected a type in the **Type Configuration** section, select **Add > Initialisation > Lifecycle Assignment**.
2. The **Lifecycle Assignment** window opens, displaying the object type at the root and the main classification taxonomy cascading below:

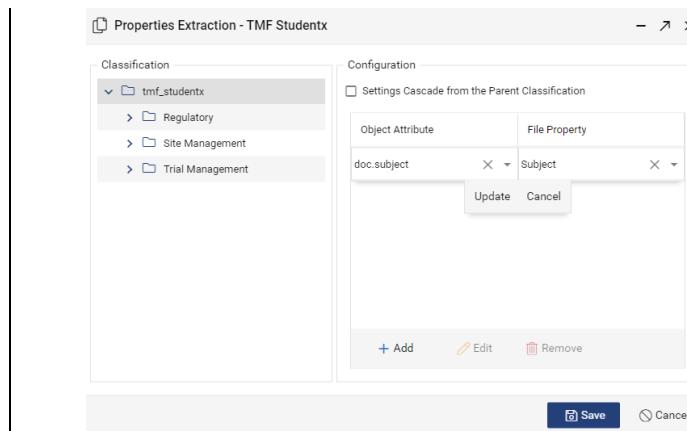


3. To assign a lifecycle, highlight a folder/node and untick **Settings Cascade from the Parent Classification**.
4. In the **Configuration** panel on the right-hand side of the window select a **Lifecycle for Objects** from the drop-down list.
5. Select a **Lifecycle for Templates** from the drop-down list.

6. Note that the folder icon for the selected node now includes a plus icon, indicating that it has a lifecycle assigned.
7. Repeat for other nodes within the main classification as required.
8. Click **Save** to return to the **Type Configuration** panel.

7.14. Properties Extraction

Description	<p>Populate document or object attributes with values taken from its content file. Multiple attributes can be selected to be populated in this way.</p> <p>Each node within the classification hierarchy can have a different specific extraction rules set, or can inherit those of its parents.</p>
Configuring Properties Extraction	<ol style="list-style-type: none"> 1. Within the Control Panel, having selected a type in the Type Configuration section, select Add > Initialisation > Properties Extraction. 2. The Properties Extraction window opens, displaying the object type at the root and the main classification taxonomy cascading below:  <p>The screenshot shows the 'Properties Extraction - TMF Studentx' window. On the left, there's a 'Classification' tree with nodes like 'tmf_studentx' (selected), 'Regulatory', 'Site Management', and 'Trial Management'. On the right, there's a 'Configuration' panel with a checked checkbox for 'Settings Cascade from the Parent Classification'. Below it is a table with columns 'Object Attribute' and 'File Property'. At the bottom, there are buttons for '+ Add', 'Edit', 'Remove', 'Save', and 'Cancel'.</p> <ol style="list-style-type: none"> 3. Highlight a folder/node and untick Settings Cascade from the Parent Classification. 4. In the Configuration panel on the right-hand side of the window click Add and select an Object Attribute which will receive the value and the File Property it will be copied from.



5. Click **Update** to save the change.
6. Click **Add** to add further attributes to make use of properties extraction.
7. Once complete click **Save**, returning to the **Type Configuration** window.

7.15. Properties Inheritance

Description

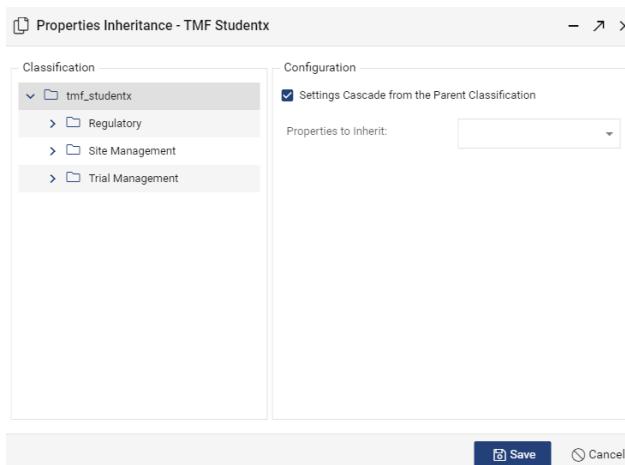
If a user highlights a document prior to selecting Create, the new document can copy the classification and selected attributes of the highlighted document. This can save users significant time when creating new documents which have a multiple attributes to be completed.

Which attributes are to be copied needs to be configured, no attributes are set to be copied by default. Fields which should be unique for each document, such as object name, should not be selected for properties inheritance.

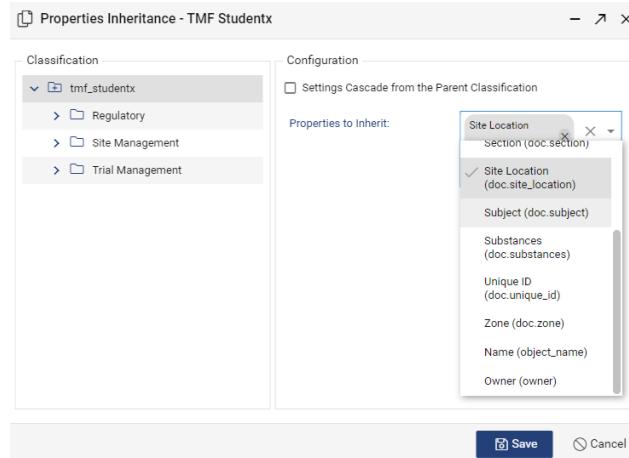
As for most type-based configuration within CARA, each level of the classification taxonomy can have a unique rule configured, or those of the parents can be inherited.

Configuring Properties Inheritance

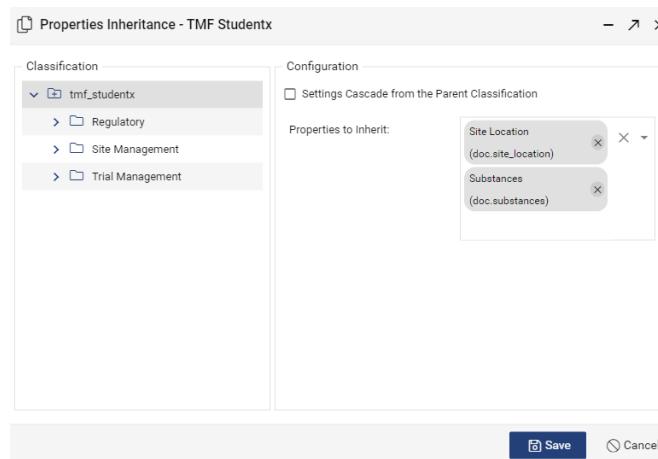
1. Within the **Control Panel**, having selected a type in the **Type Configuration** section, select **Add > Initialisation > Properties Inheritance**.
2. The **Properties Inheritance** window opens, displaying the object type at the root and the main classification taxonomy cascading below:



3. Highlight a folder/node and untick **Settings Cascade from the Parent Classification**.
4. In the **Configuration** panel on the right-hand side of the window, in the **Properties to Inherit** field, click the drop-down arrow and select the attributes to be inherited, then click **Save**.



5. Repeat for other classification nodes and **Save** once completed.



7.16. Properties Initialization

Description

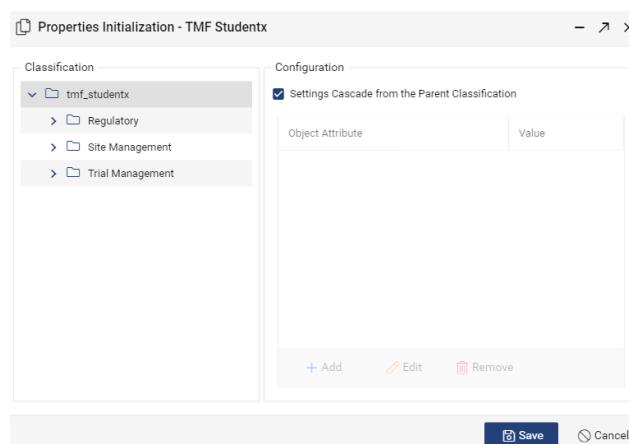
Property values can be set automatically at the time a document is created, with the aim of reducing the number of attribute values users need to manually enter and increasing the accuracy of data by removing human error where possible.

As for most type-based configuration within CARA, each level of the classification taxonomy can have a unique rule configured, or those of the parents can be inherited.

Please note that **Properties Initialization** differs from **Auto Values**. Properties initialization applies only when a document is initially created, whereas Auto Values apply each time a document is checked in.

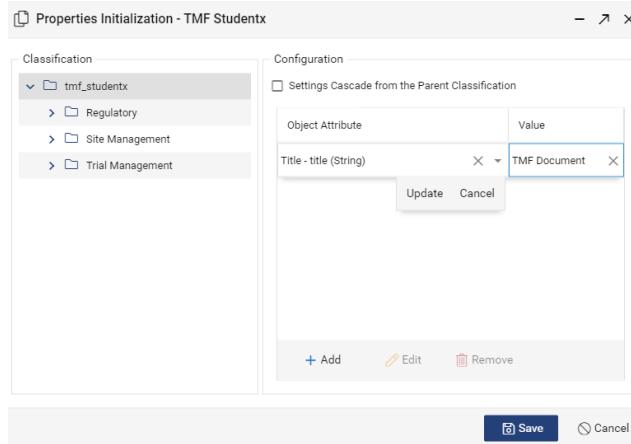
Configuring Properties Initialization

1. Within the **Control Panel**, having selected a type in the **Type Configuration** section, select **Add > Initialisation > Properties Initialization**.
2. The **Properties Initialization** window opens, displaying the object type at the root and the main classification taxonomy cascading below:



3. Highlight a folder/node and untick **Settings Cascade from the Parent Classification**.

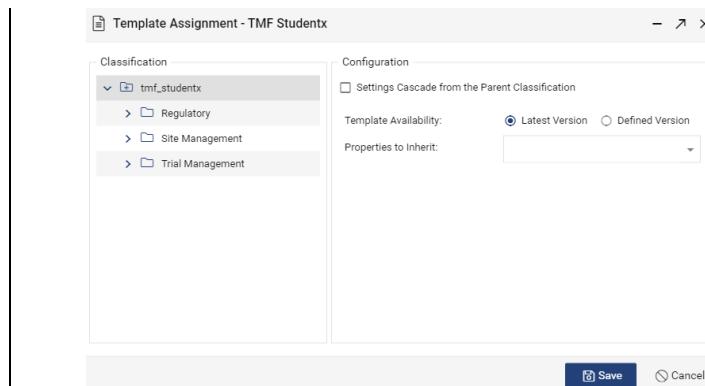
4. In the **Configuration** panel on the right-hand side of the window click **+ Add** and select an **Object Attribute** which will receive the value and the **Value** to be set.



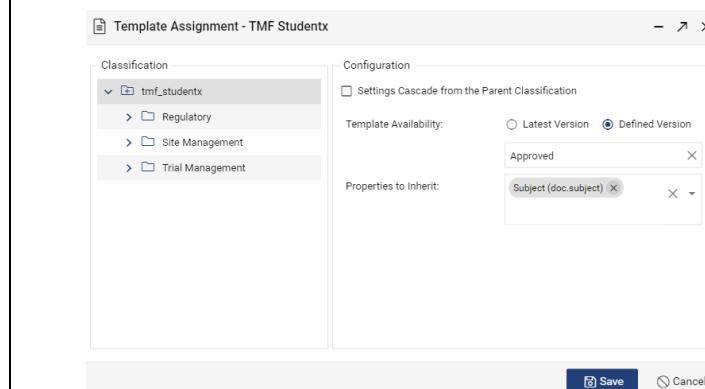
5. Click **Update** to save the change.
6. Click **Add** to add further attributes and set their initial values.
7. Once complete click **Save**, returning to the **Type Configuration** section.

7.17. Template Assignment

Description	<p>Template assignment determines which attribute values, if any, will be inherited from templates. It is also used to specify which template versions are made available to users for each classification, either the LATEST version or those with a specified version label, for example Approved.</p> <p>Although template assignment is used to determine which versions of applicable templates are made available, it does not impact the overall selection of templates, which is based on the classification each template was given at the time it was imported or created.</p> <p>Until the Template Assignment has been configured, no templates are available to users.</p> <p>Note. In CARA v5.6 Title was added as a core attribute. When the template selection dialogue is presented to users upon creating a new document based on a template, the template names are listed with the title as a prefix where populated.</p>
Configuring Template Assignment	<ol style="list-style-type: none">1. Within the Control Panel, having selected a type in the Type Configuration section, select Add > Initialisation > Template Assignment.2. The Template Assignment window opens, displaying the object type at the root and the main classification taxonomy cascading below:



3. Highlight a folder/node and untick **Settings Cascade from the Parent Classification**.
4. In the **Configuration** panel on the right-hand side of the window select either **Latest Version** or **Defined Version**. If **Defined Version** is selected enter a version label, such as Approved.
5. Select the properties to be inherited from the right-hand side drop-down menu.
6. Click **Save**.

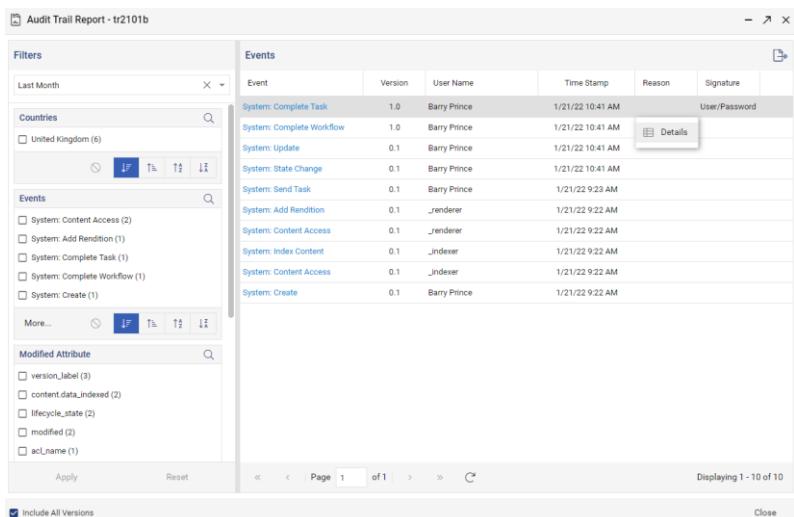


7. Repeat for other nodes in the main classification as required.
8. Once complete click **Save**, returning to the **Type Configuration** window.

7.18. Auditing

Description Auditing creates a history of activities performed on objects and documents, such as creation, editing, viewing and task completion.

The audit history is displayed to users as an **Audit Trail Report**, available from the main user interface toolbar after selecting a document. The audit trail report includes the user name, time stamp and changes to properties.



The screenshot shows the 'Audit Trail Report - tr2101b' window. On the left, there is a 'Filters' sidebar with sections for 'Last Month', 'Countries' (United Kingdom), 'Events' (System: Content Access, System: Add Rendition, etc.), and 'Modified Attribute' (version_label, content_data_indexed, lifecycle_state, modified, acl_name). The main area is titled 'Events' and displays a table with columns: Event, Version, User Name, Time Stamp, Reason, and Signature. A 'Details' button is visible in the 'Reason' column for the first event. The table lists ten events, all performed by 'Barry Prince' on 1/21/22 at various times between 10:41 AM and 9:23 AM. At the bottom, there are buttons for 'Apply', 'Reset', and 'Close', and a status message 'Displaying 1 - 10 of 10'.

Additional details including the previous and updated values for properties, are visible in a separate window, accessed by selecting **Details** from the right-click menu over an event:

The screenshot shows a dialog box titled "Audit Trail – Details". At the top, there is a "General" tab. Below it is a table with two columns: "Attribute" and "Value". The table contains the following data:

Attribute	Value
Location	North Shields, England, United Kingdom
Signature	User/Password
Document (7)	tr2101b
Object Name	tr2101b
Type Name	tmf_studentx
Object ID	144
Root Version ID	144
Version	1.0
Source State	Approved
Target State	Approved
Workflow (5)	studentx_workflow
Workflow	studentx_workflow_15
Task	172

At the bottom right of the dialog box is a "Close" button.

Different document and object types will often have different auditing requirements, for example regulated documents may require an extensive history of activities to be maintained, whereas general purpose documents may require no history at all.

Audit configuration can be toggled between **Active** and **Inactive**. Audit configurations which are inactive will not have new audit events recorded, but previously recorded audit events will be retained and remain accessible. Marking configuration as inactive can be used when setting up configuration ahead of the time it is required.

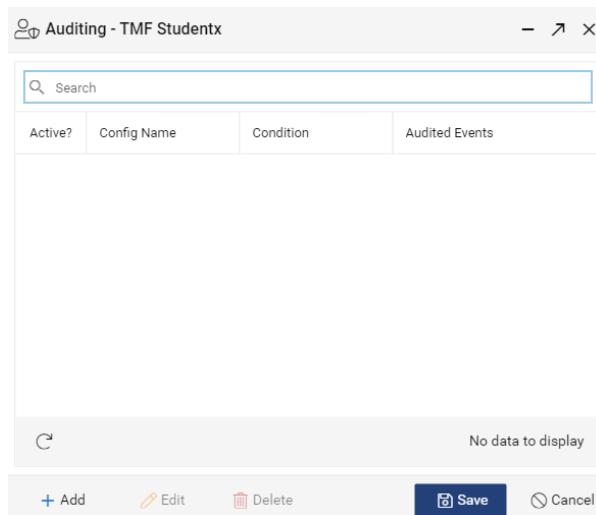
Each document or object type can have multiple different audit configurations, where different events are recorded, based on conditions. These conditions are based on attribute values, for example a document of type X may require a full history to be recorded if that document has status = approved, but may require a less extensive history if status = draft. The conditions are written using CARA Query Language syntax, for example:
lifecycle_state="Approved".

Creating an Auditing Configuration

For each audit event users can be required to enter a signature and provide a reason/meaning/capacity, either by free typing or choosing from a dictionary. The signature window label can be customised.

Where auditing has not been configured for a type, the events marked as active in the audit_events dictionary are recorded.

1. Within the **Control Panel**, having selected a type in the **Type Configuration** section, select **Add > Processing > Auditing**.
2. The **Auditing** window opens.



3. Click **+ Add** to open the **New Auditing** window.

The screenshot shows a configuration window titled "New Auditing - TMF Studentx". It contains the following fields:

- A checkbox labeled "Active?".
- A text input field labeled "Configuration Name:" with an asterisk (*) indicating it is required.
- A dropdown menu labeled "Condition".
- A table labeled "Events" with three columns: "Event Name", "Signature", and "Reason".
- Buttons at the bottom: "+ Add", "Edit", "Remove", "Save" (highlighted in blue), and "Cancel".

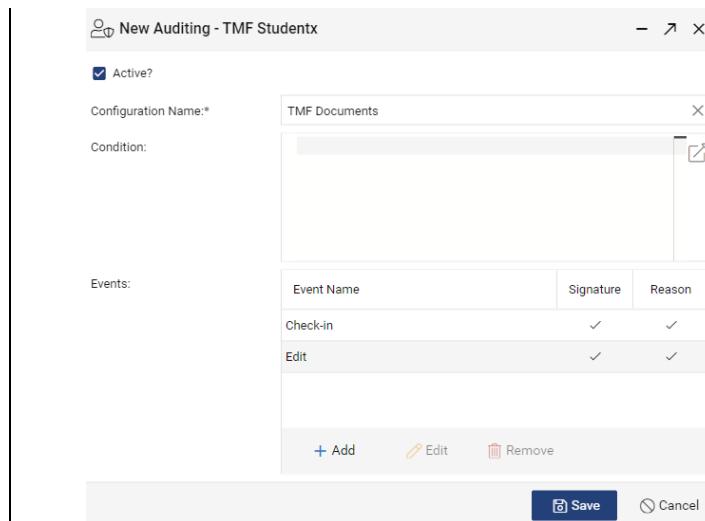
4. Tick **Active** to make the audit configuration active once it has been saved. Audit configurations which are temporarily not required can be made inactive by unselecting Active. An inactive audit will not record new events, but previously recorded events are retained and viewable.
5. Enter a unique **Configuration Name**.
6. Optionally enter a **Condition** using CARA Query language syntax. If left blank the audit configuration will apply to all documents of the connected type.
7. The **Events** panel lists the currently configured events to be recorded, by default this will be blank. Click **+ Add** to open the **New Audit Event** window:

The screenshot shows a modal dialog titled 'New Audit Event'. It contains several input fields and dropdown menus:

- Event Name:** A dropdown menu with a single item selected.
- Signature:** A dropdown menu with two options: 'Not Required' (selected) and 'Required'.
- Reason:** A dropdown menu with four options: 'Not Required' (selected), 'Required – Free Text', 'Required – Dictionary', and 'Required – Dictionary or Free Text'.
- Reason Label:** An input field with a small edit icon.

At the bottom right of the dialog are 'Save' and 'Cancel' buttons.

8. Select an **Event** from the drop-down list.
9. Choose whether or not a user carrying out the event will be **Required** or **Not Required** to enter their electronic **Signature**.
10. If a signature is required, enter a **Signature Label** which will be displayed to the user at the time the event is triggered.
11. Choose if a **Reason** is to be entered by the user when the event triggers. If required, the user can be asked to select a reason from a drop-down list, enter free text or choose between the two.
12. If a reason is required, enter a **Reason Label**.
13. If a reason is required and users are to be provided with a drop-down list to select from, for **Reason Dictionary** select the dictionary to be used.
14. **Save** the audit event and return to the **New Auditing** window.
15. **Add** and configure other audit events as appropriate.



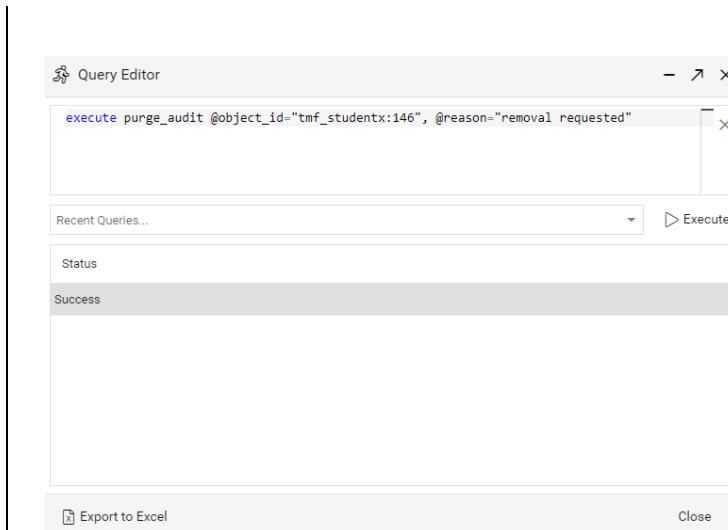
16. When complete **Save** the new auditing configuration, returning to the **Auditing** panel.

Auditable Events

- Add Relation
- Add Rendition
- Cancel Checkout
- Check-In
- Check-Out
- Controlled Print
- Delete
- Edit
- Export
- Print
- Remove Relation
- Remove Rendition
- State Change
- Trash
- Update
- View
- View PDF

Purging Audit Entries

Audit entries for individual documents can be purged with a CQL command run from the Query Editor. The purge command can only be run by administrators.



When the purge command is run, an audit event entry is recorded against the user that ran the command. A reason must be provided when running the purge and is included within the audit event.

The screenshot shows the 'Events' interface with the following details:

- Event:** System: Purge Audit Trails
- User Name:** Execute Query
- Time Stamp:** Not explicitly shown in the screenshot, but implied by the context.
- Reason:** removal requested
- Audit Trail - Details:**
 - General** tab is visible.
 - Properties Change** tab is selected.
 - Added** section shows the following properties:

Attribute	Value
object_id.binding_type	ById
object_id.id	146
object_id.type_name	tmf_studentx
reason	removal requested
- Buttons:** Close

The CQL syntax for the purge audit command is as follows:

```
execute purge_audit @object_id="type:id", @reason="reason"
```

@object_id. The object ID of the single object version whose audit events will be purged. To purge the audit events of all versions of the same document, the purge command would need to run multiple times

@reason. A reason must be provided, which is recorded and displayed in the purge audit trail event details.

Optional Conditions with Example Values:

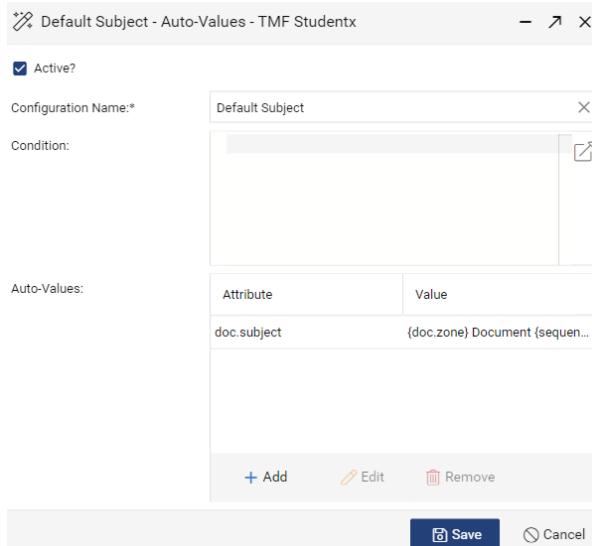
Event name and time range can be included as conditions:

```
@event_name='checkout',  
@range_from='2021-05-08T12:00:00.00Z',  
@range_to='2021-05-09T12:00:00.00Z'
```

7.19. Auto-Values

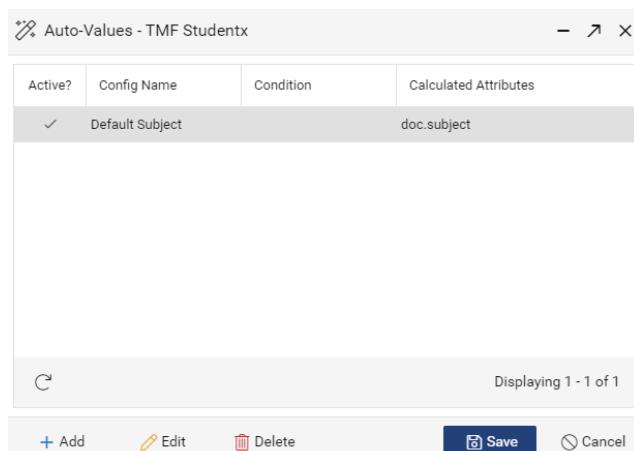
Description

Auto-values configuration allows property values to be set automatically based on a condition, when a document is checked-in. Please note that **Auto-values** differs from **Properties Initialization**, which are set only when the document is initially created.

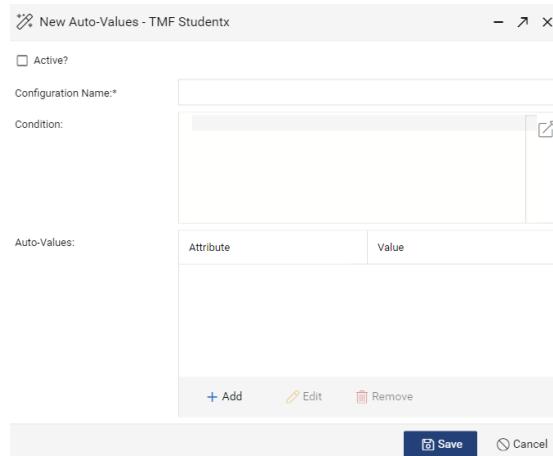


Configuring Auto-values

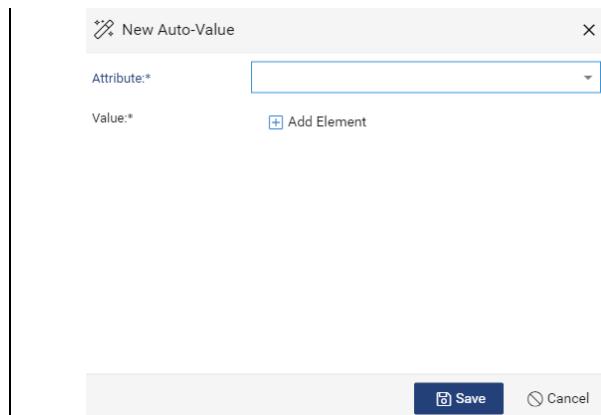
1. Within the **Control Panel**, having selected a type in the **Type Configuration** section, select **Add > Processing > Auto-values**.
2. The **Auto-values** panel opens, displaying any existing auto-values rules:



3. Click **+ Add**, the **New Auto-values** window opens:



4. Tick **Active** if the rule is to be active once saved. Alternatively auto-values rules can be configured in advance of being needed and activated only when appropriate.
5. Enter a **Configuration Name**.
6. Optionally enter a **Condition**, based on CARA Query Language (CQL), for example:
`status="Draft"`
Please note that the condition does not require the Where command to be included.
If no condition is entered, the rule will apply to all documents of the selected type.
7. In the **Auto-values** field, click **+ Add**, the **New Auto-value** window opens:



New Auto-Value

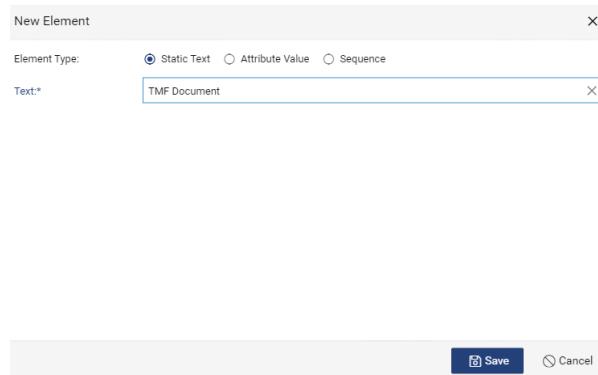
Attribute:*

Value:*

+ Add Element

Save Cancel

8. Select an **Attribute** from the drop-down list.
9. Click **+ Add Element** beside **Value**.
10. Select the **Element Type**:
11. **Element Type: Static Text.** Enter the text into the **Text** field. If other elements will be added that follow the text value it may be desirable to add a space at the end of the text string.:



New Element

Element Type:

Static Text Attribute Value Sequence

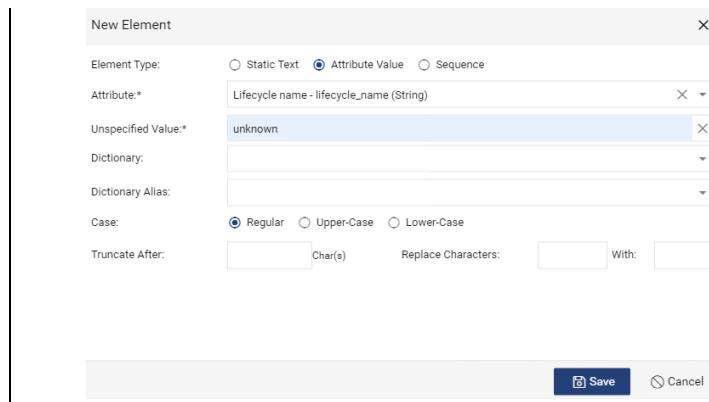
Text:*

TMF Document

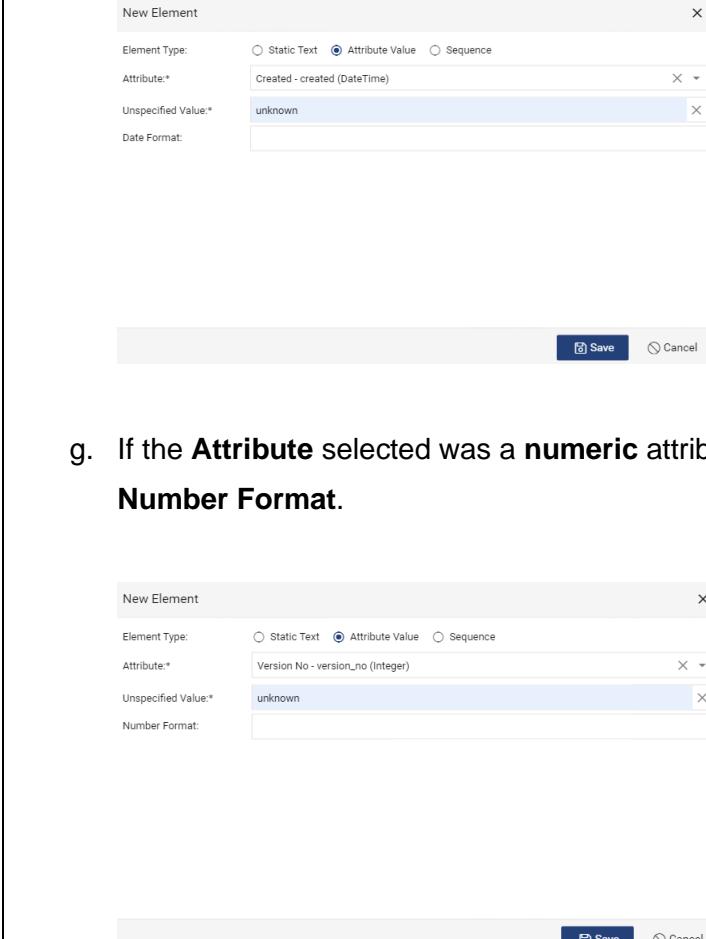
Save Cancel

12. **Element Type: Attribute Value.** Existing attribute values can be used in the new attribute Auto-value rule. Select an **Attribute** from the drop-down list:

Subsequent options are attribute data type dependant.



- a. Enter an **Unspecified Value**, which will be entered in the auto-value if no valid value for the specified attribute is available.
- b. Optionally select a **Dictionary** and whether a **Dictionary Alias** should use a specified fixed alias (such as English, French, German etc..) or an alias based on the user's locale. This is useful where users may use multiple languages but a single language is used in the auto values for consistency.
- c. Select the **Case** for the auto-value element as either **Regular** (the text case will be the same as that entered, including mixed-case), all **Upper-case** or all **Lower-case**.
- d. Optionally select to **Truncate** after x number of characters. This can be useful where individual attributes values are potentially long but the total auto-value should be kept within a specified limit.
- e. Optionally where attributes values are a fixed format, characters in a specified position and be replaced, enter a **Replace Characters** value and the alternative values to replace them **With**.
- f. If the **Attribute** selected was a **date** based attribute enter a **Date Format**.



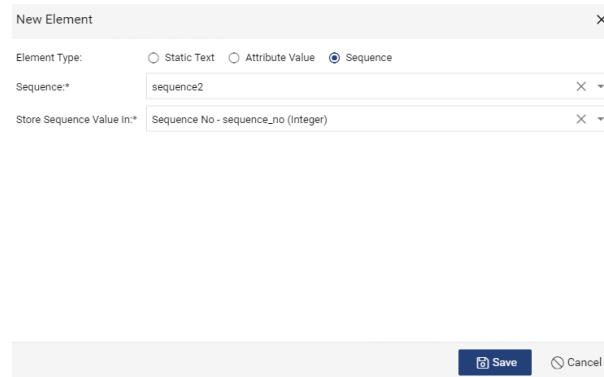
The screenshot shows the 'New Element' dialog box. The 'Element Type:' section has 'Attribute Value' selected. The 'Attribute:' field contains 'Created - created (DateTime)'. The 'Unspecified Value:' field contains 'unknown'. The 'Number Format:' field is empty. At the bottom right are 'Save' and 'Cancel' buttons.

- g. If the **Attribute** selected was a **numeric** attribute enter a **Number Format**.

13. **Element Type: Sequence.** A sequence can be added as an element of the auto-value rule, these are often used to ensure that each document or object is uniquely numbered. In order for a sequence value to be included in an auto-value rule it must be separately stored in an attribute associated with the document or object, any numeric field can be used for this purpose, but system fields should be avoided. Dedicated numeric fields are typically created for the purpose. The field used to store the sequence value does not need to be visible to users. Please see the separate **Sequences** chapter for details of configuring sequences.

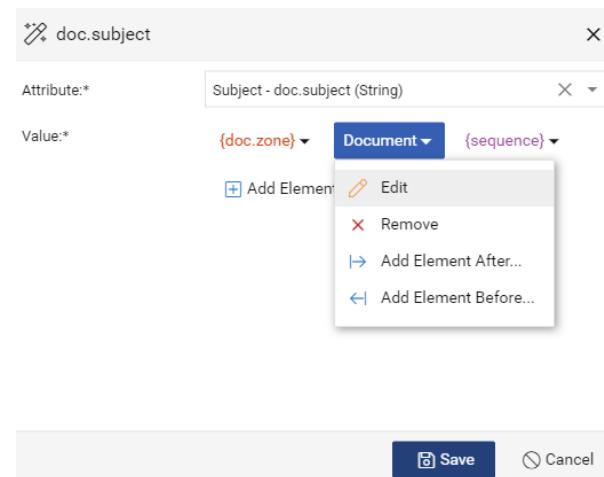
- a. Select a **Sequence** from the drop-down list.
- b. Select an attribute to **Store Sequence Value in**. Only integer fields can be selected. The system field **sequence_no** can be

used, alternatively any integer based type-specific field (those prefixed with **doc.**) can be selected:



14. Save the New Auto-value element.

- a. Add additional elements to the auto-value rule as required.
- b. Auto-value elements can be edited, re-ordered and removed as required from elements' drop-down menu:



- c. Once the auto-value rule is complete click **Save** to return to the main Auto-values window.

15. Save the Auto-values configuration:

7.20. Change Notifications

Description

Users can manually subscribe to documents, in which case they will be sent change notifications for the selected documents. This function is available by default within the user interface. The events which trigger change notifications are all of those which are currently active within the _notification_events dictionary.

In addition administrators can create condition based notifications to be sent to users who are explicitly selected or based on a dynamic attribute such as creator. This avoids the need for users to setup notifications manually and for those users to receive notifications for all active events.

New Change Notifications - TMF Studentsx

Active?

Configuration Name:

Condition that must be valid before change:

Condition that must be valid after change:

Users/Groups to Notify:

Dynamic Recipients: {creator}

**Configuring
Change
Notifications**

1. Within the **Control Panel**, having selected a type in the **Type Configuration** section, select **Add > Processing > Change Notifications**.
2. In the **Change Notifications** window click **Add** to open the **New Change Notifications** window:

The screenshot shows a configuration dialog titled 'New Change Notifications - V56 Manual Folders'. It includes fields for 'Active?' (checkbox), 'Configuration Name:' (text input), 'Condition that must be valid before change:' (list box), 'Condition that must be valid after change:' (list box), 'Users/Groups to Notify:' (list box with a dropdown arrow), and a 'Save' button.

3. Select if the change notification will be **Active**. Notifications are only triggered for active notification rules.
4. Enter a **Configuration Name**.
5. Optionally enter a **Condition that must be valid before change**.

For example if a notification of a document being approved was only to be sent if the preceding lifecycle state was draft, and not any other state:

`lifecycle_state = 'Draft'`

6. Enter a **Condition that must be valid after change**.

This field is required and is the primary condition that the notification is intended to trigger from. For example a notification of documents entering an approved state:

`lifecycle_state = 'Approved'`

7. Select the **Users/Groups to Notify**. Multiple users and groups can be selected. If a group is selected, all members of the group will receive individual notifications.

8. Click **Add Recipient** if users are to be notified based on a dynamic attribute such as creator, the **Dynamic Recipients** and **Add Element** fields are displayed:

Click **Add Element** to open the **New Dynamic Recipient** window.

The screenshot shows the 'New Element' dialog box. At the top, there is a header bar with a close button ('X'). Below it, the title 'New Element' is displayed. Underneath the title, there is a section labeled 'Element Type:' with two radio buttons: one for 'Static Text' (which is selected) and one for 'Attribute Value'. A text input field labeled 'Text:' with an asterisk (*) is present, containing a placeholder text area. At the bottom right of the dialog box, there are two buttons: a blue 'Save' button and a grey 'Cancel' button.

- a. Select an **Element Type**, either **Static Text** or **Attribute Value**.
- b. To enter the recipient value as a simple text string select **Element Type** as **Static Text**.
- c. To generate the recipient value from an attribute select **Element Type** as **Attribute Value** and select an **Attribute** from the drop-down list, such as creator.

The screenshot shows the 'New Element' dialog box. At the top, there is a header bar with a close button ('X'). Below it, the title 'New Element' is displayed. Underneath the title, there is a section labeled 'Element Type:' with two radio buttons: one for 'Static Text' and one for 'Attribute Value' (which is selected). A dropdown menu labeled 'Attribute:' is open, showing the option 'Creator - creator (Authority)'. Below the dropdown, there is a text input field labeled 'Unspecified Value:' with an asterisk (*) containing the word 'unknown'. At the bottom right of the dialog box, there are two buttons: a blue 'Save' button and a grey 'Cancel' button.

Additionally enter an **Unspecified Value** to be substituted should the selected attribute be blank.

Attribute values which are string format can be aliased from a dictionary, have their case modified and be truncated as necessary:

New Element

Element Type: Static Text Attribute Value

Attribute:* Country - doc.country (String)

Unspecified Value:* unknown

Dictionary:

Dictionary Alias:

Case: Regular Upper-Case Lower-Case

Truncate After: Char(s) Replace Characters: With:

Save **Cancel**

- d. **Save** the **New Dynamic Recipient** element, returning to the New Change Notifications window. Add additional elements as required.

9. **Save** the New Change Notification rule, returning to the **Change Notifications** window where the new rule is displayed:

Change Notifications - TMF Students

Active?	Config Name	Condition Before Chan...	Condition After Change	Users/Groups to Notify	Dynamic Recipients
✓	Author Approval Notification	lifecycle_state='Draft'	lifecycle_state='Approved'	{creator}	

Displaying 1 - 1 of 1

+ Add **Edit** **Delete** **Save** **Cancel**

10. **Add** other notification rules as appropriate and **Save**.

7.21. Content Access Configuration

Description	Content access configuration is used to restrict which renditions and/or formats are available to users when viewing documents. Content access configuration does not apply to the formats available when editing documents.
-------------	--

Active?	Config Name	Condition	Protection Mode
	Approved Documents ...	lifecycle_state='Approved'	DocSecure
✓	Approved Documents	lifecycle_state='Approved'	Restrict to Defined Format

No data to display

+ Add Edit Delete Save Cancel

By default users can view all available formats, provided they have appropriate security permissions on the document. For example, a user with read content permission can view the document in either the original MS Word format or the rendition .pdf format. Content access configuration can be used to restrict users to only viewing the .pdf format.

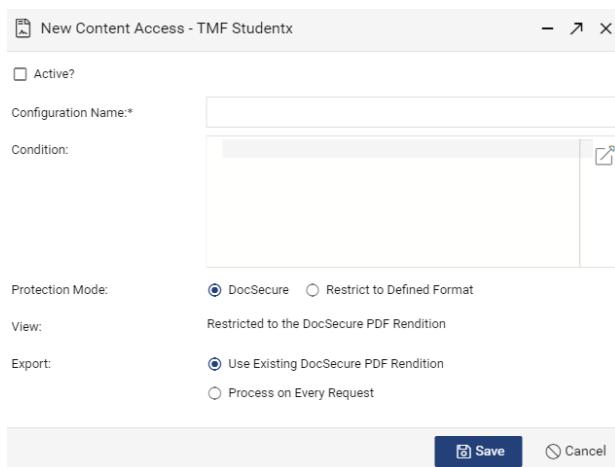
Content access configuration can be based on CQL conditions, for example the viewing of approved documents could be restricted to .pdf while draft documents are not restricted.

Content access configuration is also used to determine the formats and specific renditions available for viewing when DocSecure is enabled for the CARA environment. DocSecure is a separate application that provides additional functionality for watermarking, electronic signature pages and property overlays on PDF files. DocSecure is part of the CARA product family. While it is installed

Configuring Content Access

separately from CARA, with its own configuration and extensive options, it is fully integrated with CARA v5.

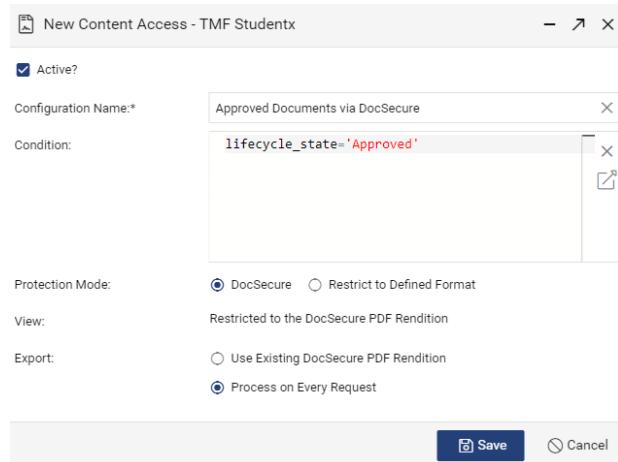
1. Within the **Control Panel**, having selected a type in the **Type Configuration** section, select **Add > Processing > Content Access Configuration**.
2. In the Content Access Configuration panel click **Add** to open the **New Content Access** window:



3. Select if the content access configuration will be **Active**.
4. Enter a **Configuration Name**.
5. Optionally enter a **Condition** using CARA Query Language syntax. If no condition is entered, the configuration will apply to all documents of the selected type.
6. Select the **Protection Mode** as either **DocSecure** or **Restrict to Defined Format**.
 - a. **DocSecure** is a separate application that provides additional functionality when renditions are created, such as adding watermarks, electronic signatures and overlays. If DocSecure is part of the CARA environment select DocSecure.

The View setting will be set to **Restricted to the DocSecure PDF Rendition**.

For **Export** select to either **Use Existing DocSecure PDF Rendition** or **Process on Every Request**. This option is useful where attributes used by watermarks or overlays may have changed since the document was created or updated and a new rendition should be generated each time a request to export is made.



- b. If DocSecure is not being used select **Restrict to Defined Format**.

In **Restrict to Format** enter the viewing formats available to users, typically pdf. This field is required.

Optionally in the **Restrict to Rendition** field enter a rendition identifier. This option is not usually required, however where multiple renditions are created for a document each one may be configured with an identifier such as 'internal' or 'approved' and it may be appropriate to restrict users to only viewing renditions with a specific identifier.

New Content Access - TMF Studentx

Active?

Configuration Name:

Condition:

Protection Mode: DocSecure Restrict to Defined Format

Restrict to Format:

Rendition identifier:

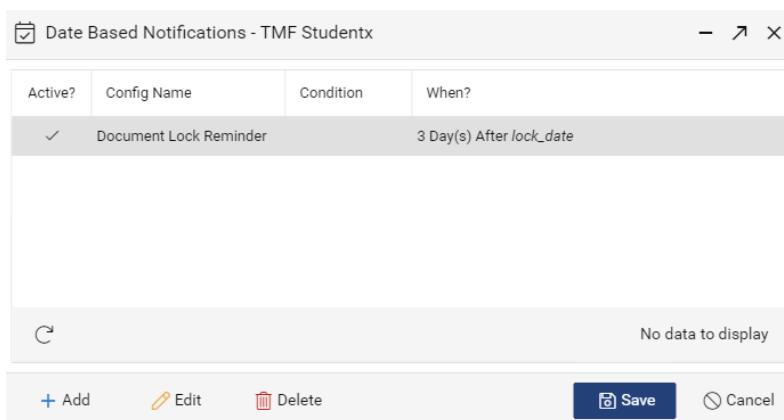
7. **Save**, returning to the **Content Access** window. Add further content access configuration as required.

8. **Save** and close the **Content Access** window.

7.22. Date Based Notifications

Description

Notifications can be automatically sent to users based on document date attributes. Standard system date-based attributes such as created or deleted can be used. Type specific date-based attributes can also be used. Additional conditions can be added based on CQL.



Date based notifications use custom notification templates.

The recipients of date based notifications can be a combination of manually selected users/groups and dynamic recipients based on an attribute value such as creator or lock_owner.

The CARA Send Date Based Notifications job checks for and sends active date based notification configurations each time it is run. The job logs detail which date based notifications were triggered.

Creating Date Based Notifications

1. Within the **Control Panel**, having selected a type in the **Type Configuration** section, select **Add > Processing > Date Based Notifications**.
2. In the **Date Based Notifications** window click **Add** to open the **New Date Based Notifications** window:

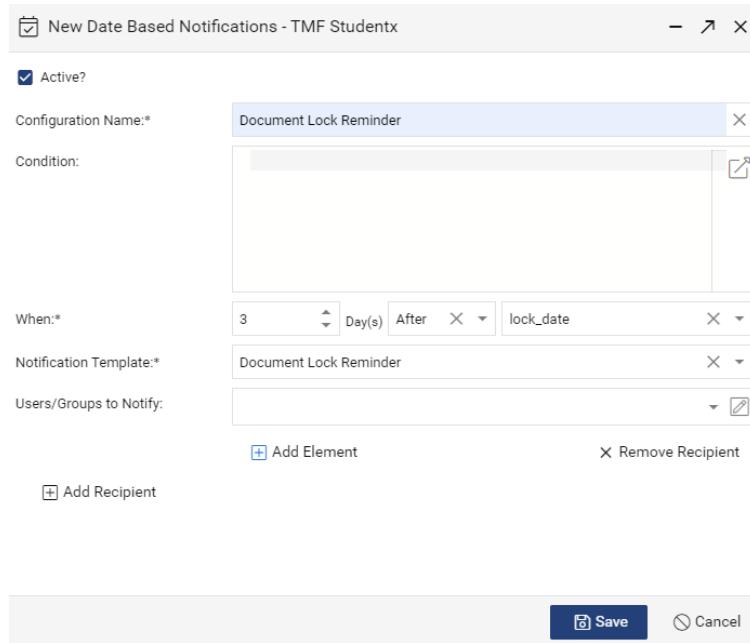
The screenshot shows a configuration dialog box titled "New Date Based Notifications - TMF Studentx". It contains the following fields:

- Active? (checkbox)
- Configuration Name:*
- Condition: (dropdown menu with a search icon)
- When:*(highlighted with a red box)
 - Day(s) After
- Notification Template:*
- Users/Groups to Notify: (dropdown menu with a search icon)
- Add Recipient (button)

At the bottom right are "Save" and "Cancel" buttons.

3. Select if the configuration will be **Active**. Only active configurations are processed when the **CARA Send Date Based Notifications** job is run.
4. Enter a **Configuration Name**.
5. Optionally enter a **Condition** using CARA Query Language syntax. Any condition entered here is in addition to the required date based condition beneath.
6. **When**. Select the date & time field, and the number of days before or after, which will trigger the sending of notifications. A system date field such as created or lock_date, or a type specific date field, can be selected.
7. Select the **Notification Template** to be used. Only custom notification templates can be selected.
8. Select the **Users/Groups to Notify**. Multiple users and groups can be selected. If a group is selected, all members of the group will receive individual notifications.

9. Click **Add Recipient** if users are to be notified based on a dynamic attribute such as creator or lock_owner. The **Add Element** field is then displayed:



Click **Add Element** to open the **New Dynamic Recipients** window.

- Select an **Element Type**, either **Static Text** or **Attribute Value**.
- To enter the recipient value as a simple text string select **Element Type as Static Text**.
- To generate the recipient value from an attribute select **Element Type as Attribute Value** and select an **Attribute** from the drop-down list such as creator or lock_owner:

New Element

Element Type: Static Text Attribute Value

Attribute: * Lock owner - lock_owner (Authority)

Unspecified Value: * unknown

Save **Cancel**

Enter an **Unspecified Value**, which is substituted should the selected attribute be blank.

- d. **Save** the **New Dynamic Recipient** element, returning to the New Date Based Notifications window. Add additional recipients if required.

Active?

Configuration Name: * Document Lock Reminder

Condition:

When: * 3 Day(s) After lock_date

Notification Template: * Document Lock Reminder

Users/Groups to Notify: {creator}

Add Recipient **Save** **Cancel**

10. Click **Save** and return to the **Date Based Notifications** window.

11. **Save** and close the **Date Based Notifications** window.

7.23. Legal Hold Items

Description Legal hold can be used to prevent the modification of documents or objects of a specified type. This can be useful when working with documents which are subject to regulation, where it has been requested that no further updates are made to documents related to a specific product or process.

A legal hold configuration will override the user permissions granted by ACLs applied to individual documents.

Legal hold configurations can include a condition, therefore if a product was under investigation, it should not be necessary to halt work on all related documents if the investigation relates to a specific subset of documents, for example only documents which have been approved or published.

Named groups, such as administrators, can be allowed to continue to update documents on legal hold.

Multiple legal hold configurations can be created, each with a specific condition. Each legal hold configuration can be active or inactive, allowing configuration to be created in advance of being required.

Legal Hold Items - TMF Studentx			
Active?	Config Name	Condition	Groups Allowed to Modify Do...
✓	regulatory_doc_hold	doc.reg_status='Hold'	_admin
No data to display			
+ Add	Edit	Delete	Save Cancel

**Configuring
Legal Hold**

1. Within the **Control Panel**, having selected a type in the **Type Configuration** section, select **Add > Processing > Legal Hold Items**.
2. In the **Legal Hold Items** window click **Add** to open the **New Legal Hold Items** window:

The screenshot shows the 'New Legal Hold Items' configuration window. At the top, it says 'New Legal Hold Items - TMF Studentx'. Below that is a checkbox labeled 'Active?'. A field labeled 'Configuration Name:' contains the text 'TMF Studentx'. Under 'Condition:', there is a large text area with a 'Condition' placeholder and a 'Group Selection' button. At the bottom, a dropdown menu labeled 'Groups Allowed to Modify Documents on Legal Hold:' is shown with a 'Group Selection' button next to it. At the very bottom are 'Save' and 'Cancel' buttons.

3. Select if the legal hold item will be **Active**.
4. Enter a **Configuration Name**.
Optionally enter a **Condition** using CARA Query Language syntax. If no condition is entered, the rule will apply to all documents of the selected type.
5. In the **Groups Allowed to Modify Documents** field, click the drop-down arrow to open the **Group Selection** window and choose which groups, if any, will be allowed to modify documents on legal hold.

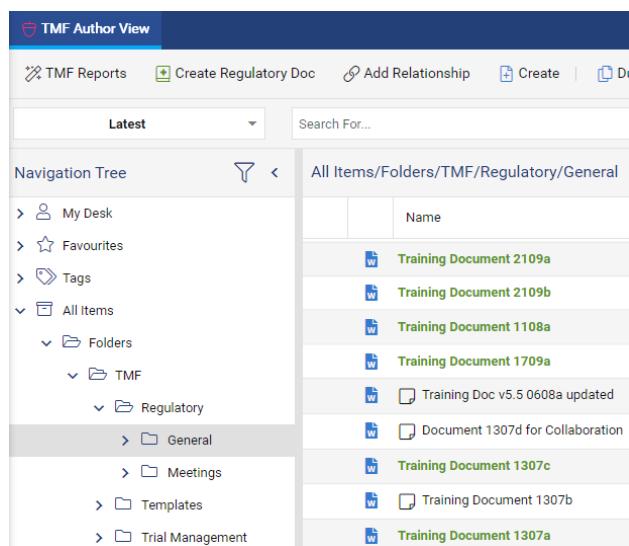
The screenshot shows a configuration window titled "New Legal Hold Items - TMF Studentx". It includes fields for "Active?", "Configuration Name:" (set to "regulatory_doc_hold"), "Condition:" (set to "doc.reg_status='Hold'"), and "Groups Allowed to Modify Documents on Legal Hold" (set to "Administrator (admin)"). At the bottom are "Save" and "Cancel" buttons.

6. **Save**, returning to the **Legal Hold Items** window, **Add** further legal hold items as required.

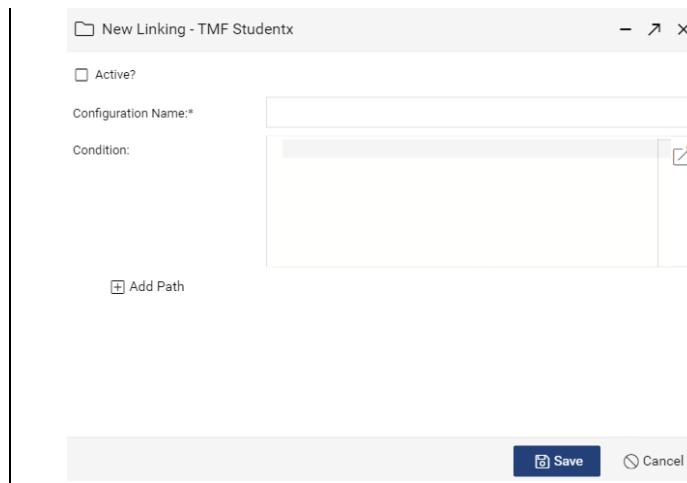
7. **Save** and close the **Legal Hold Items** window.

7.24. Linking

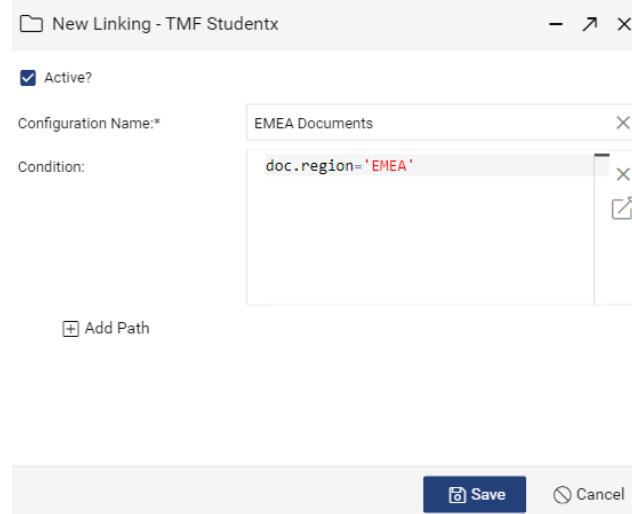
Description	<p>Linking configuration defines rules for automatically placing documents within a folder hierarchy. Typically users can browse a folder hierarchy in order to locate documents. Folder locations for documents are optional, they are convenient for users but not a configuration requirement. CARA automatically creates the folder paths as required by linking rules.</p> <p>Alternatively users can be allowed to create and select folders manually, in which case linking rules are not required.</p> <p>Whether folders are created automatically or manually is set during the initial type definition.</p>
--------------------	--



- | | |
|---------------------------------------|---|
| Creating Linking Configuration | <ol style="list-style-type: none"> 1. Within the Control Panel, having selected a type in the Type Configuration section, select Add > Processing > Linking. 2. In the Linking window click Add to open the New Linking window: |
|---------------------------------------|---|



3. Select if the new link configuration will be **Active**. Linking rules can be created but left in an inactive state until needed.
4. Enter a **Configuration Name**.
5. Enter a **Condition**, using a CARA Query Language **Where** statement. The where command is automatically included and does not need to be typed.
Example: doc.region='EMEA'



6. Click **+ Add Path**. The **Folder Paths** field displays. Folder paths can use static text, attribute values or a combination of the two.
7. Click **+ Add Element**. The **New Element** window opens:

New Element

Element Type: Static Text Attribute Value

Text:*

Save Cancel

8. Select the **Element Type** as either **Static Text** or **Attribute Value**.
9. If selecting **Static Text** enter a value in the **Text** field exactly as it should be displayed, including capital letters if required, for example Training Folder.
10. If selecting **Attribute Value**, select an attribute from the drop-down list. **Subsequent options are attribute type dependant.**
11. Enter an **Unspecified Value**, this will be used and included in the path if no valid value for the specified attribute is available:
12. If the **Attribute** selected was a **text** attribute:

New Element

Element Type: Static Text Attribute Value

Attribute:*

Zone - doc.zone (String)

Unspecified Value:*

unknown

Dictionary:

Dictionary Alias:

Case: Regular Upper-Case Lower-Case

Truncate After: Char(s) Replace Characters: With:

Save Cancel

- a. Optionally select a **Dictionary** and whether a **Dictionary Alias** should use a specified fixed alias (such as English, French, German etc..) or an alias based on the user's locale. This is

- useful where users may use multiple languages but a single language is used in the folder path for consistency.
- b. Select the **Case** for the folder element as either **Regular** (the text case will be the same as that entered, including mixed-case), all **Upper-case** or all **Lower-case**.
 - c. Optionally select to **Truncate** after x number of characters. This can be useful where attributes values lengths are potentially long but the folder path length should be kept within a specified limit.
 - d. Optionally where attributes values are a fixed format, characters in a specified position can be replaced, enter a **Replace Characters** value and the alternative values to replace them **With**.
 - e. Click **Save** to return to the **Linking** rule configuration window.

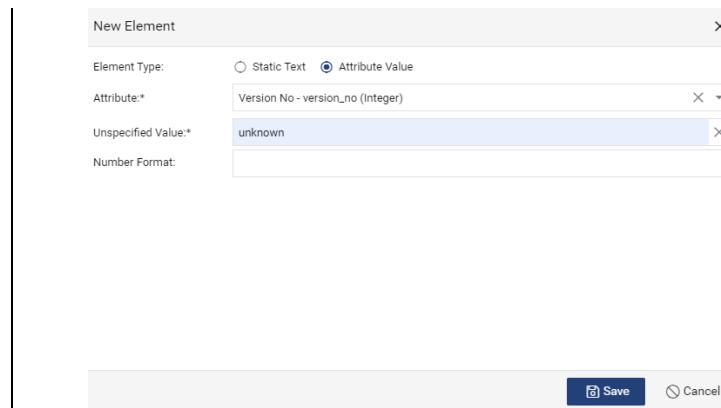
13. If the **Attribute** selected was a **date** attribute enter a **Date Format**.

The screenshot shows the 'New Element' dialog box. It has the following fields:

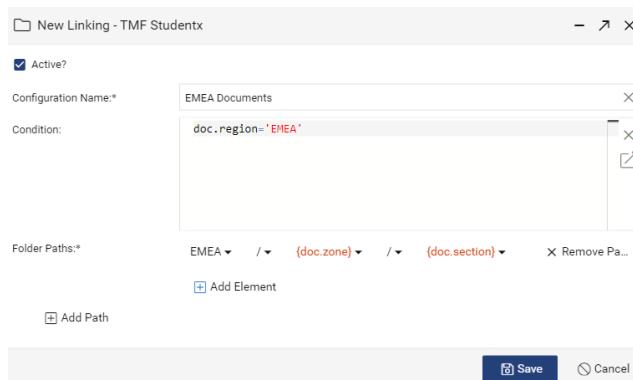
- Element Type:** A radio button group with 'Static Text' and 'Attribute Value' selected.
- Attribute:***: A dropdown menu showing 'Created - created (DateTime)'.
- Unspecified Value:***: A text input field containing 'unknown'.
- Date Format:** An empty text input field.

At the bottom right of the dialog are two buttons: a blue 'Save' button with a disk icon and a white 'Cancel' button with a circular arrow icon.

14. If the **Attribute** selected was a **numeric** attribute enter a **Number Format**.



15. Back in the **Linking** configuration window add additional elements to the **Folder Path** as required.
16. **Important note.** A forward slash (*/*) Static Text element must be added between folders in the path. There should be no forward slash at the beginning or end of the folder path, as in the following example:



17. Multiple paths can be added. Please note that the dimensions feature may remove a need to have documents linked to multiple folder paths. If required click the **Add Path** field and specify additional paths.
18. Individual paths can be removed using the **Remove Path** option.
19. Once all folder paths have been entered, **Save** the linking configuration, returning to the main Linking window:

Active?	Config Name	Condition	Folder Paths
✓	Document Templates	is_template=true	TMF/Templates
✓	EMEA Documents	doc.region='EMEA'	EMEA/{doc.zone}/{doc.section}
✓	Default User Documents		TMF/{doc.zone}/{doc.section}

⟳

Displaying 1 - 2 of 2

[+ Add](#) [Edit](#) [Delete](#) [Save](#) [Cancel](#)

20. The order of linking rules is significant and is applied from the top row down. For each document, the first active matching rule in the list determines the folder path used. The order of rules is changed using click and drag:

Active?	Config Name	Condition	Folder Paths
✓	Document Templates	is_template=true	TMF/Templates
✓	EMEA Documents	doc.region='EMEA'	EMEA/{doc.zone}/{doc.section}
✓	Default User Documents		TMF/{doc.zone}/{doc.section}

⟳

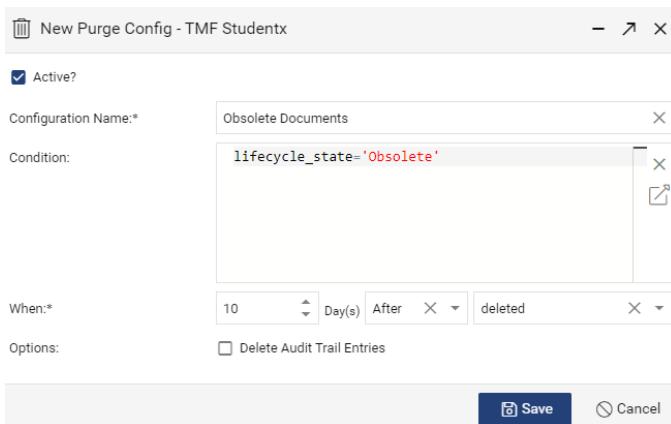
Displaying 1 - 2 of 2

[+ Add](#) [Edit](#) [Delete](#) [Save](#) [Cancel](#)

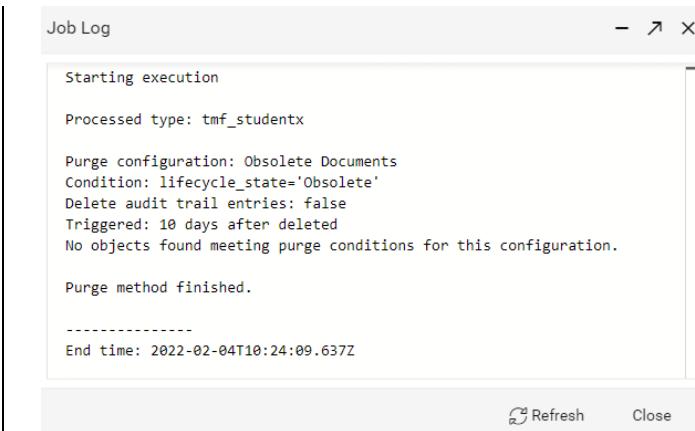
21. Once all required linking rules for the type have been created and placed in the correct order, click **Save** to return to the main Control Panel window.

7.25. Purge Configuration

Description	Purge configuration is used to automatically delete objects or documents after a set period of time. In addition to purging documents, it can also delete their audit trail entries. Each type can have its own purge configuration. All purge configurations must have a date field that acts as the primary trigger. One of the system default date fields can be used, or a type specific field can be created. Additional conditions can be created and based on any attribute type.
--------------------	---

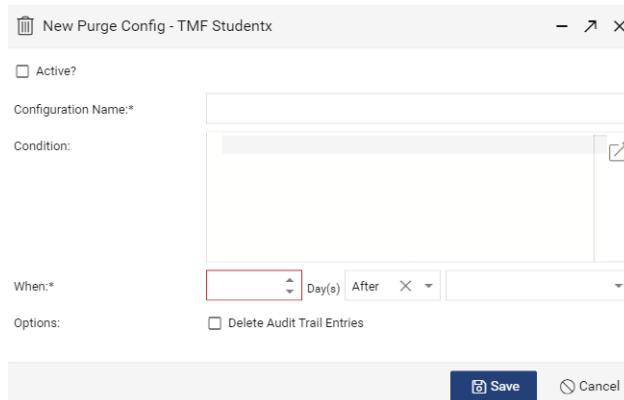


There is a **cara_purge** system job which when run purges documents which meet active purge criteria. By default the job has a daily run schedule, is inactive and set to test mode. In test mode the job only reports which documents meet purge configuration criteria. Test mode must be set to false in order for documents to be purged. Like all CARA jobs, in addition to running automatically when active, it can be run manually on-demand.



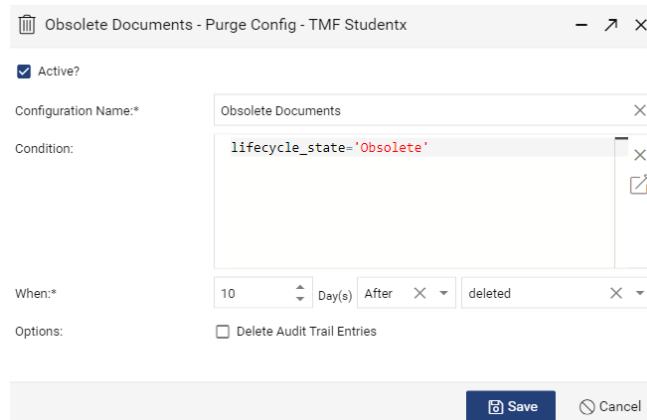
Creating Purge Configuration

1. Within the **Control Panel**, having selected a type in the **Type Configuration** section, select **Add > Processing > Purge Configuration**.
2. In the **Purge Configuration** window click **Add** to open the **New Purge Configuration** window:

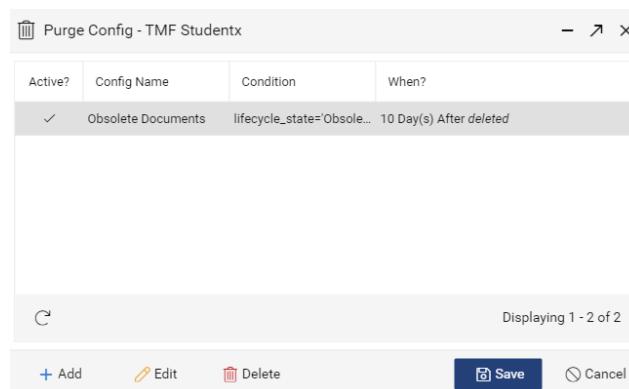


3. Select if the purge configuration will be **Active**. Only active purge configurations are processed when the cara_purge job is run.
4. Enter a **Configuration Name**.
5. Enter a **Condition** using CARA Query Language syntax. Any condition entered here is in addition to the primary date field condition beneath.

6. **When.** Select the date & time field which will act as the primary condition, a default system date field can be used, or a type specific field can be selected. Specify the number of **Days, After** or **Before** the chosen date attribute, which the purge will trigger at.



7. Select if **Audit Trail Entries** are to be deleted.
8. Click **Save** and return to the **Purge Configuration** window, **Add** further purge configuration rules as appropriate.



9. **Save** and close the **Purge Configuration** window.

Please see the jobs section of this manual for details of the cara_purge job.

7.26. Queuing

Description	<p>In CARA queues are used to pass requests to a service such as those for generating renditions or full-text indexing of content text.</p> <p>Queues are configured in the Admin section of the Control Panel, then individual types have their own Queuing configuration for when to send requests to the Queues, which is covered in this chapter.</p> <p>However note that where a type definition has the option to Render PDF Automatically enabled it is not necessary to create a queuing configuration in order to have PDF renditions automatically generated. Also where a type definition has the Indexed for Search option enabled it is not necessary to create a queuing configuration to have documents full-text indexed.</p> <p>DocSecure requests are sent automatically and do not need a separate queuing configuration to be created.</p> <p>Queuing configuration for a type can be used to configure which requests will be sent to custom queues, any parameters to be passed with the requests, and the conditions required to trigger those requests.</p>
--------------------	--

The screenshot shows the 'New Queuing - TMF Studentx' configuration window. It includes fields for 'Active?' (checked), 'Configuration Name:' (Custom Queuing 1), 'Condition' (empty), 'Queue Name:' (Custom Queue1), 'Parameters' (empty), and 'Events' (checkboxes for Create, Update, Version, Set Content, where Create is checked). At the bottom are 'Save' and 'Cancel' buttons.

Creating Type Queuing Configuration

1. Within the **Control Panel**, having selected a type in the **Type Configuration** section, select **Add > Processing > Queuing**, the **Queuing** window opens:

The screenshot shows the 'Queuing - TMF Studentx' window with a table header including columns for Active?, Config Name, Condition, Queue Name, and Events. Below the table is a toolbar with buttons for Add, Edit, Delete, Save, and Cancel.

2. No queue requests are configured by default. Click **Add**, the **New Queuing** window opens:

The screenshot shows a configuration dialog titled "New Queuing - TMF Studentx". It includes fields for "Active?", "Configuration Name", "Condition" (with a placeholder icon), "Queue Name", "Parameters", and a list of "Events" (Create, Update, Version, Set Content). At the bottom are "Save" and "Cancel" buttons.

3. Tick **Active** to the make queue request active once saved. If the service to be requested has not yet been made available it can be left as inactive.
4. Enter a **Configuration Name**.
5. Optionally enter a **Condition** in CARA Query Language syntax, for example status="Approved". If left blank, every time the trigger **Event** selected occurs a request will be generated.
6. Select the **Queue Name** from the drop-down list. Custom queues are created in the Administration > Queues section of the control panel.
7. Enter any **Parameters** required in the request e.g. a rendition request may need to specify the format such as .pdf.
8. Tick which **Events** will trigger the request. Multiple can be selected:
 - a. Create.
 - b. Update.
 - c. Version.

- d. Set Content.
9. **Save the New Queuing** configuration.

7.27. Security

Description

In security configuration, specify the ACL (access control list) which will be applied to documents or objects of the selected type. Multiple security configuration rules can be created and each can include attribute based conditions, for example a document with status = draft would typically have a different ACL (and therefore permissions) applied than one where status = approved.

Active?	Config Name	Condition	Security
✓	Approved Documents	lifecycle_state='Approved'	approved_acl
✓	Draft Documents	lifecycle_state='Draft'	draft_acl
✓	Default User Docs		studentx_acl

Displaying 1 - 1 of 1

+ Add Edit Delete Save Cancel

Please note that Access Control Lists, which detail the permissions available to users and groups, are configured separately in the Admin section of the Control Panel. Access Control Lists can themselves contain conditions which provide alternative permission levels based on an attribute.

Configuring Security Configuration

1. Within the **Control Panel**, having selected a type in the **Type Configuration** section, select **Add > Processing > Security Configuration**.
2. In the **Security** window click **Add** to open the **New Security** window:

The screenshot shows a configuration window titled 'New Security - TMF Studentx'. It contains the following fields:

- Active?**: A checkbox that is currently unchecked.
- Configuration Name:** A text input field containing the placeholder text 'Configuration Name:*.
- Condition:** A dropdown menu or list box with a small edit icon on the right.
- ACL Template:** A dropdown menu or list box with a small edit icon on the right.

At the bottom of the window are two buttons: **Save** (blue) and **Cancel**.

3. Select if the security configuration will be **Active**.
4. Enter a **Configuration Name**.
5. Enter a **Condition**, for example status='Draft'.
6. Select the **ACL Template** which will be applied.

The screenshot shows the same configuration window as above, but with the following changes:

- Active?**: A checkbox that is now checked.
- Configuration Name:** A text input field containing 'Draft Documents'.
- Condition:** A dropdown menu or list box containing the value 'lifecycle_state='Draft''.
- ACL Template:** A dropdown menu or list box containing 'Draft ACL (draft_acl)'.

At the bottom of the window are two buttons: **Save** (blue) and **Cancel**.

7. **Save** and return to the **Security** window.
8. **Add** further security configuration rules as required and **Save**.

9. The order of security rules is significant and is applied from the top row down. For each document, the first active matching rule in the list determines the ACL applied.

The order of the security rules is changed using click and drag:

Active?	Config Name	Condition	Security
✓	Approved Documents	lifecycle_state='Approved'	approved_acl
✓	Draft Documents	lifecycle_state='Draft'	draft_acl
✓	Default User Docs	Default User Docs	udentx_acl

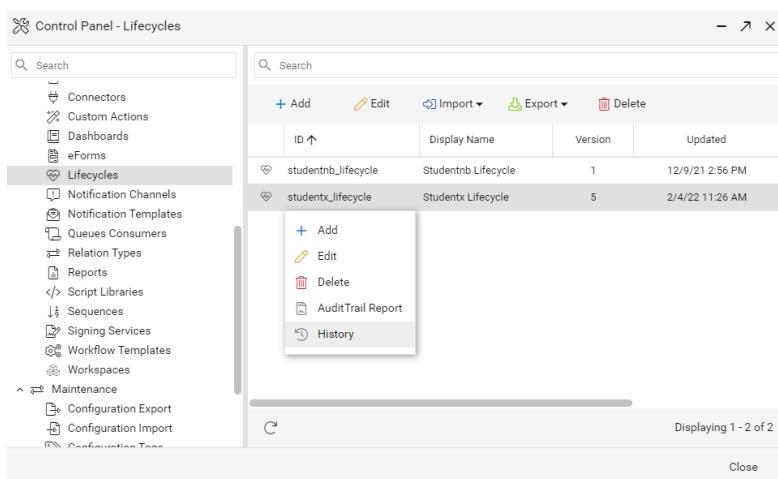
Displaying 1 - 1 of 1

+ Add Edit Delete Save Cancel

8. Configuration History, Restore & Audit

Description

A history of configuration changes is automatically recorded. Changes can be viewed, compared and reverted to as needed for most configuration elements within the control panel. To view the history of a configuration element, right-click and select **History**:



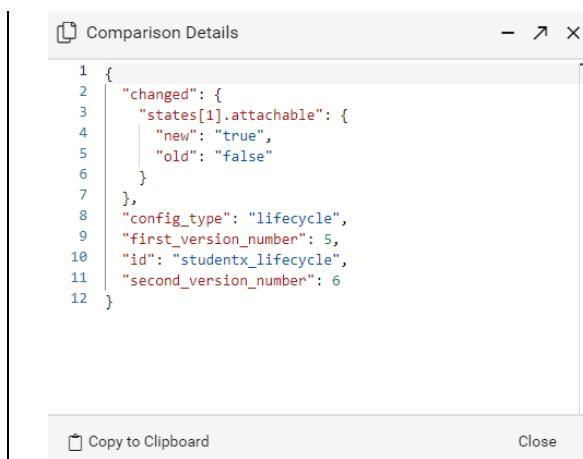
The **Configuration History** window details the configuration element **Display Name**, **Version**, **Updated date**, **Updated By** and any **Tags** applied:

Display Name	Version ↓	Updated	Updated By	Tags
Studentx Lifecycle	6	2/4/22 11:31 AM	Barry Prince	
Studentx Lifecycle	5	2/4/22 11:26 AM	Barry Prince	
Studentx Lifecycle	4	9/20/21 12:01 PM	Barry Prince	
Studentx Lifecycle	3	5/28/21 10:16 AM	Barry Prince	
Studentx Lifecycle	2	3/19/21 2:02 PM	cadmin	UAT_Complete
Studentx Lifecycle	1	3/19/21 2:02 PM	cadmin	

Compare selected

Revert **Close**

Click **Compare Selected** with 2 or more configuration versions highlighted to open the **Comparison Details** window. The Comparison window details the changes and additions between versions.



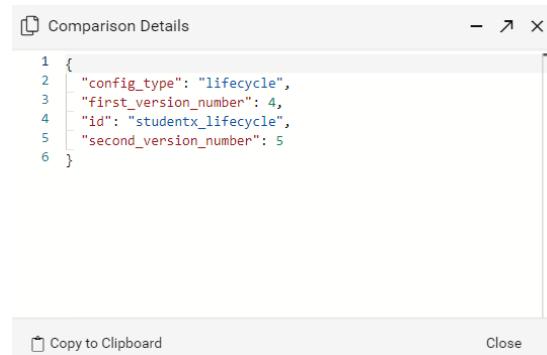
```

1 {
2   "changed": {
3     "states[1].attachable": {
4       "new": "true",
5       "old": "false"
6     }
7   },
8   "config_type": "lifecycle",
9   "first_version_number": 5,
10  "id": "studentx.lifecycle",
11  "second_version_number": 6
12 }

```

Close

If no changes were made between versions, the Comparison Details window will open but only show the configuration version numbers selected:



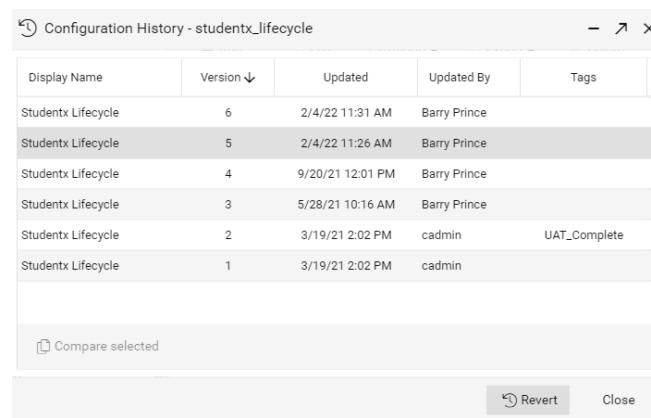
```

1 {
2   "config_type": "lifecycle",
3   "first_version_number": 4,
4   "id": "studentx.lifecycle",
5   "second_version_number": 5
6 }

```

Close

Configuration can be reverted, select the version to be reverted to and select **Revert**. Click **Yes** in the **Confirmation** window:



Display Name	Version	Updated	Updated By	Tags
Studentx Lifecycle	6	2/4/22 11:31 AM	Barry Prince	
Studentx Lifecycle	5	2/4/22 11:26 AM	Barry Prince	
Studentx Lifecycle	4	9/20/21 12:01 PM	Barry Prince	
Studentx Lifecycle	3	5/28/21 10:16 AM	Barry Prince	
Studentx Lifecycle	2	3/19/21 2:02 PM	cadmin	UAT_Complete
Studentx Lifecycle	1	3/19/21 2:02 PM	cadmin	

Configuration Audit Trail Report

An audit trail of configuration changes is also available to view.
Right-click a configuration element and select Audit Trail Report:

The screenshot shows the 'Control Panel - Lifecycles' interface. On the left, there's a sidebar with various configuration categories like Connectors, Custom Actions, Dashboards, eForms, Lifecycles, etc. The 'Lifecycles' category is selected. On the right, a list of lifecycle configurations is displayed, with two entries: 'studentnb_lifecycle' and 'studentx_lifecycle'. A context menu is open over the second entry, showing options: Add, Edit, Import, Export, Delete, and AuditTrail Report. The 'AuditTrail Report' option is highlighted.

Report filters are provided in the right-hand side. The date filter defaults to the last month. A specific date range can be chosen, select **Range** within the date filter drop-down list.

Right-click an individual event and select **Details** to open the separate Details window, which includes changes to properties on a separate tab:

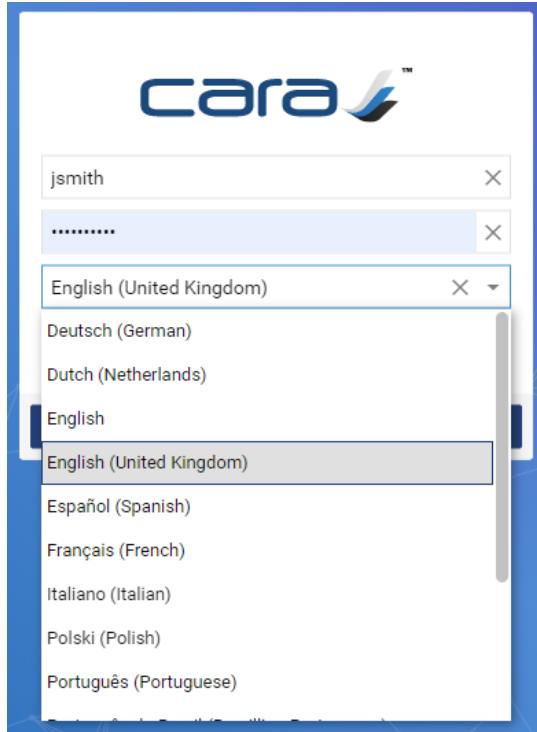
The screenshot shows the 'Audit Trail Report - studentx_lifecycle' interface. On the left, there are filters for Range, Countries, Events, Modified Attribute, and Types. In the center, a list of events is shown, with one event selected: 'System: Update'. A details window is open over this event, titled 'Audit Trail - Details'. The window has tabs for General and Properties Change. The General tab shows the event type as 'Modified' and the User Name as 'System: Update'. The Properties Change tab displays a table of modified attributes with their old and new values:

Attribute	Old Value	New Value
modified	2022-02-04T11:26:38.303984Z	2022-02-04T11:31:56.569346Z
states[1].attachable	false	true
version	5	6

9. Supported Locales

Description

Users can select a user interface display language when logging into CARA:



The following locales are supported by CARA Version 5.6 and have built in translation of the main user-interface menus and labels.

- Brazilian Portuguese
- Chinese
- Czech
- Dutch
- English
- English UK
- French
- German
- Italian
- Japanese
- Polish

- Portuguese
- Russian
- Spanish

Customer-specific user interface labels, such as those that are displayed on a property form for a custom document or object type, are customised by the use dictionary key aliases. Please see the dictionaries chapter for details.

Active?	Key	en	fr
<input checked="" type="checkbox"/>	Regulatory	Regulatory	Régulateur
<input checked="" type="checkbox"/>	Site Management	Site Management	Gestion du Site
<input checked="" type="checkbox"/>	Trial Management	Trial Management	Gestion des Essais

Buttons at the bottom: Add, Edit, Delete, Export to Excel, Import from Excel, Save, Cancel.

Language Auto Detection

Automatic detection of the user's language, based on the language of their operating system, is available. In the configuration file cara-ui.yml set autoDetectLanguage to true.

The following order is used to determine the user interface display language applied:

1. User's saved language from the previous session.
2. Use the OS language where supported if autoDetectLanguage is true.
3. If autoDetectLanguage is false or the OS language is not available in cara-ui.yml locales, use the locale default language.
4. If none of the above are available, use English.

10. CARA Query Language - CQL

Description

CQL (CARA Query Language) is a SQL-based query language used in the CARA query editor, report queries and as value assistance in forms. Currently it supports **select** and **execute** statements.

CQL commands are not case-sensitive. Select and SELECT are both valid.

Select

The select statement runs queries against types created in CARA and also the default system types. Note that type-specific attributes are prefixed with .doc. Attributes in a select statement are separated with a comma.

From

The from statement is used to specify the object type to be returned

Use:

`select <attributes or wildcard> from <type>`

Examples:

`select object_name, creator, created from type1`

Replace type1 with your custom type.

`select * from type1`

Returns default system attributes for type1, it does not include type-specific attributes.

`select object_name, creator, doc.subject from type1`

subject is a custom attribute of type1, therefore it is prefixed with doc.

System attributes such as creator are not prefixed.

Where

The where statement used to limit the objects returned for a type

Both apostrophes and speech marks can be used to enclose text within a query, for example:

where creator="bob"

and

where creator='bob'

are both valid.

Be very careful if copying queries from MS Word document as characters such as quotes may be automatically altered.

Enclosed text is case-sensitive:

where lifecycle_state='Draft'

will return different results than

where lifecycle_state='draft'

Examples:

select object_name, created from type1 where creator='bob'

Returns objects created by bob.

select object_name, creator from type1 where doc.subject = 'Test Document'

returns objects where subject equals Test Document.

Operators:

<, <=, >, >=, (), =, !=, like, and, or, is null, is not null, now() = current_datetime(), today() = current_date(), user, between, in

Like

Wildcard characters in a where statement use **like**

Multi-character wildcard character is *

Single-character wildcard is %

Examples:

select object_name, creator from type1 where doc.subject like
'*Document*'

Returns objects where subject contains Document and any number of prefix or suffix characters.

```
select object_name, creator from type1 where doc.subject like  
'Document%'
```

Returns objects where subject begins with Document and is followed by any single character.

```
select object_name from type1 where creator = 'Bob' or creator  
='Sue'
```

Returns objects created by Bob or Sue.

```
Select object_name from type1 where creator !='Bob'
```

Returns objects not created by Bob.

```
select object_name from type1 where doc.subject is not null
```

Returns objects where subject is not empty.

user indicates the current logged in user.

today indicates the current date.

Examples:

```
select object_name, created from type1 where creator=user  
returns objects created by the currently logged in user.
```

```
select object_name, creator from type1 where created=today  
returns objects created today.
```

IN

Allows selecting from a sub-query. The sub-query can be a defined list of values or another select.

Examples:

```
select * from type1 where doc.region in ('Europe','America','Asia')  
returns objects where doc.region is Europe, America or Asia.
```

```
select * from type1 where creator in (select owner from type1)
returns objects where the value for creator is within the values for
owner.
```

All

By default select statements only return the latest version of objects.
Add 'all' or "all" after the type name to return all versions.

Examples:

```
Select object_name, version_label from type 1
Returns only the latest versions of objects.
```

```
select object_name, version_label from type1 'all'
Returns all versions of objects.
```

Group By

Groups returned objects by an attribute. Use count(*) to return the number of objects for each value. A maximum of 2 attributes can be grouped by.

Example:

```
select creator, count(*) from type1 group by creator
Returns objects grouped by creator, with a count of the number of
objects per creator.
```

TODATE and TODATETIME

Used to specify a date or date and time.

Examples:

```
select object_name, created from type1 where created >
todate('2021-01-18')
Returns objects created after January 18th 2021.
```

```
select object_name, created from type1 where created >
todatetime('2021-01-18' 23:00)
Returns objects created after 11pm on January 18th 2021.
```

DATEADD

Used to return an offset date or time value.

Example:

```
select object_name, created from type1 where created >
```

```
DATEADD(CURRENT_DATE(), days, -5)
```

Returns objects created in the last 5 days:

Execute

The execute statement runs the given procedure on the server.

Syntax:

```
execute <procedure_name>
```

parameter names are prefixed with @

Available procedures:

clear_cache. Clears the system cache

clear_queue. Remove all items from the specified queue

show_cache. Lists cached items

show_sessions. Returns a list of active user sessions

purge_audit. Removes audit entries for a specified object. See the audit chapter for full syntax details and examples.

Example:

```
execute clear_queue @queue_name="_fulltext"
```

Runs the clear queue procedure on the queue named _fulltext.

11. Query Editor

Description

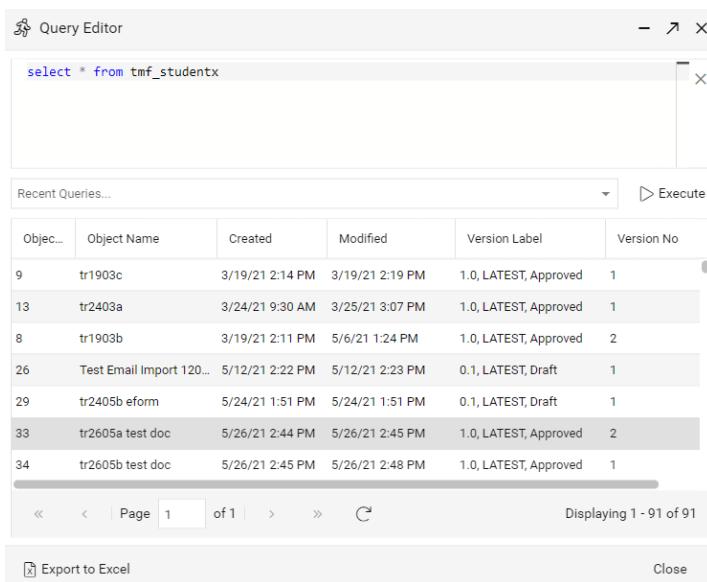
CARA provides a query editor for running CQL queries within the user interface.

Query Editor is available within the **Utilities** drop-down menu, which is displayed in the top right corner of the main user interface.

Users must be a member of the `_query_executor` system group in order for the query editor to be available in the Utilities drop down menu.

Users are still constrained by their ACL permissions when using the query editor. A user who has no permissions on an object will not see that object when queried through query editor.

Enter the query in the top field and click **Execute**.



The columns displayed can be customised by clicking the drop-down arrow visible in the column headings when hovered over with the mouse, this is useful when using the `*` wildcard to return all attributes but only a selection needs to be displayed.

Object...	Object Name	Created	Modified	Version Label	Version No
9	tr1903c		2:19 PM	1.0, LATEST, Approved	1
13	tr2403a		3:07 PM	1.0, LATEST, Approved	1
8	tr1903b			T, Approved	2
26	Test Email Import 121			T, Draft	1
29	tr2405b eform	5/24/21 1:51 PM	5/24/21	T, Draft	1
33	tr2605a test doc	5/26/21 2:44 PM	5/26/21	T, Approved	2
34	tr2605b test doc	5/26/21 2:45 PM	5/26/21	T, Approved	1
38	Site Management Tem...	5/26/21 7:15 PM	5/26/21	T, Draft	1
42	tr2705c	5/27/21 2:45 PM	5/27/21	T, Draft	1
41	tr2705b	5/27/21 12:45 PM	5/27/21	T, Draft	1

Sort Ascending
Sort Descending
Columns >

Page 1 of 1 >> Close

Export to Excel

Displaying 1 - 91 of 91

Object Id
Object Name
Owner
Creator
Modified
Modifier
Acl Name
Version Label
Version No
Root Version Id

The most recent queries entered are selectable in the Recent Queries drop-down menu.

Query results can be directly exported to Excel. Click **Export to Excel** and select an appropriate location for the export file when prompted.

Close the Query Editor when finished.

12. Scripting Utilities

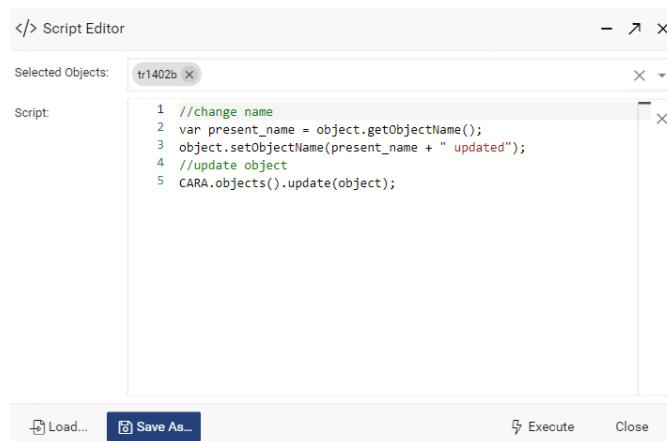
Description

CARA supports the use of scripts within multiple configuration areas, including custom actions, lifecycles and workflow templates. CARA scripts use Groovy, a superset of Java. The CARA Scripts Guide provides details and examples of use.

Several utilities are provided to assist with the use of scripts within CARA and are described below.

Script Editor

CARA provides a script editor for writing, editing, saving and executing scripts:



The script editor is available from the Utilities drop-down menu in the top-right corner of the user interface.

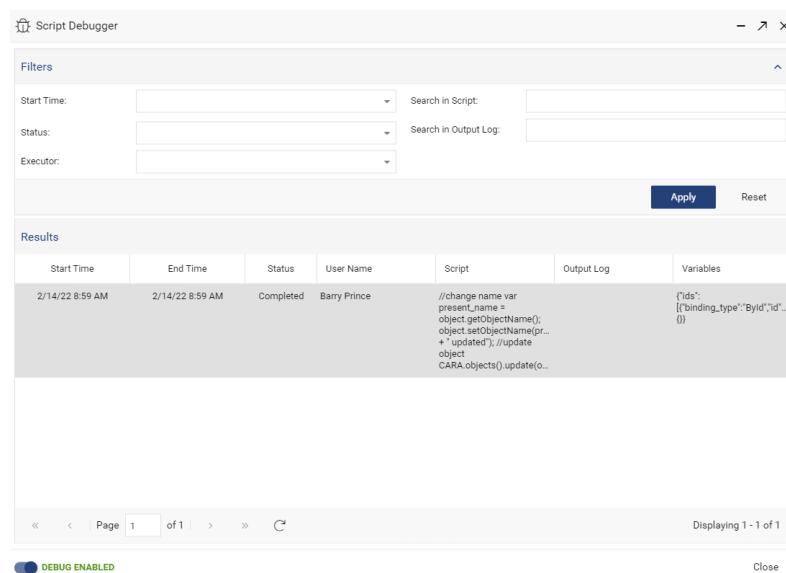
Users must be a member of the `_script_executor` system group in order for the script editor to be available in the Utilities drop down menu.

If a document is highlighted before the script editor is launched, the document's name is displayed in the **Selected Objects** field and scripts will be applied to that document when executed. Multiple documents can be selected. If no document is selected when the script editor is launched, the Selected Objects field is hidden.

The script editor provides options to **Load** a previously saved script, **Save As** the script currently displayed within the editor, **Execute** the current script and **Close**.

Script Debugger

A script debugger is provided and is available from the Utilities menu, Users must be members of the `_script_executor` group in order for to access the script debugger utility.



The option to enable and disable the script debugger is located in the bottom left corner. When enabled, all script executions are written into the dedicated `cara_script_log` index. The index is monthly-based and removed with the log purge method.

The following script execution information is provided:

- Status
- Start time
- End time
- User name
- Script
- Variables
- Output log

Filters are provided for narrowing the returned results, including script start time, status and executor.

Script Library

The script library is available within the CARA control panel.

ID	Display Name	Version	Updated	Updated By
CommonUtils	Common Utilities	104	11/01/2021, 20:10	cadmin
d0663333	Script_JLibrary	1	20/02/2020, 08:01	cadmin
dtff	dtff	2	18/09/2019, 20:57	cadmin
DocuSignSDKIntegration	DocuSignSDKIntegration	20	27/07/2021, 07:45	Bartosz Czyzowski
Employee	Employee POJO	101	14/09/2021, 07:39	Vadim Popa
GenericTypeLibrary	Library for the Generic Type@ Complex	6	14/11/2019, 20:38	Larissa Bezrukova
Global-test	dfdf	1	19/06/2019, 07:18	cadmin
GoogleTranslation	Google Translation Library	3	18/08/2020, 16:17	Yoshihiko Aschi
HelloWorldScript	ScriptForDemo	2	04/09/2020, 11:35	Barry Prince
ImportedtestWW	ImportedtestWW	6	07/12/2020, 11:17	Tomasz Porański
new	new	1	19/06/2019, 08:25	cadmin
PasswordGenerator	Password Generator	1	28/04/2020, 10:54	Yoshihiko Aschi
PDFExtraction	PDF Extraction Demo	2	06/04/2020, 15:02	Yoshihiko Aschi
ScriptValidation	ScriptValidation	44	07/01/2021, 08:29	Bartosz Czyzowski
ScriptTest	Script Test	8	05/05/2020, 06:26	Yoshihiko Aschi

From the General > Scripts section of the control panel select Add to add a script to the library:

Scripts must be provided with an **ID** and **Display Name**.

ID:

Display Name:

```
package library;
class {Name} {
```

Save **Cancel**

Once added, **Save** the script.

Scripts can be edited, imported, exported and deleted from the Scripts Library panel. Scripts which are in use cannot be deleted.

13. CARA Condition Language - CCL

Description	CARA Condition Language is a SQL-based API that will accept CaraObject and query as parameters and return true or false. The query must support: Regular operators like: AND, OR, IN, NOT IN, IS NULL, IS NOT NULL, BETWEEN, NOT BETWEEN, LIKE, NOT LIKE Brackets () Comparators: = != <> <> Arithmetic operations * / + - Keywords: NULL, TRUE, FALSE
Statement	Details
DATEADD (date/datetime, unit, value)	Supported units for adding/subtracting from given date/datetime: Unit.Seconds, Unit.Minutes, Unit.Hours, Unit.Days, Unit.Weeks, Unit.Months, Unit.Quarters, Unit.Years
INLIST (list, value)	Function that matches a value against a list of values and evaluates to an indicator of membership in the list. For example: INLIST(SPLIT('1,2,3', ','), '2')
MAKELIST (...)	Function that creates a list with each argument being one element in the list. For example: MAKELIST('1', '2', '3')
REGEX (pattern, value)	Function that matches a value against a regular expression. For example: REGEX('A.B', 'A-B'). Note that the regular expression is not anchored; use the anchor characters, ^ and \$, as-needed. For example, REGEX('AA', 'XAAx') evaluates to true while REGEX('^AA\$', 'XAAx') evaluates to false.

REPLACE (string, pattern, replacement)	Function which replaces regular expression matches in a source string to a replacement literal. For example: REPLACE('1,2/3', '[./]', ';') returns '1;2;3'
SPLIT (string, pattern [, limit])	Function which splits a string into a list of strings given a regular expression for the separator.
TODATE(...) and TODATETIME(...)	When those functions are invoked with one argument that is of type LocalDate or ZonedDateTime they will act like a cast function: TODATE(ZonedDateTime) produces LocalDate and TODATETIME(LocalDate) produces ZonedDateTime

14. CARA System Object Model

Object Names and Attributes

User

Index name: .cara_user

Attributes:

name
login_name
active
locked
description
email_address
source
password
password_salt
failed_login_attempt
last_login
photo
ldap_dn
ldap_id

Role

Index name: .cara_role

Attributes:

name
description
source
user_members
role_members
admins
modified
modifier
ldap_dn
ldap_id

User Token

Index name: .cara_user_token

Attributes:

user_name
token_name
token_key
ip_address
expiration_date

Type

Index name: .cara_type

Attributes:

type_name
description
icon
status
index_name
index_version
version
modified
modifier

Session

Index name: .cara_session_data-{now/M{YYYY_MM}}

Attributes:

ip_address
connection_id
user_name
auth_method
auth_outcome
login_timestamp
logout_timestamp

Object

Index name - based on type and its version

Attributes:

object_id

object_name
owner
created
creator
modified
modifier
lock_owner
lock_date
in_transaction
is_template
template_id
lifecycle_name
lifecycle_state
acl_name
acl_readers
root_version_id
previous_version_id
version_no
version_label
is_latest
status
deleted
content
type
format
length
checksum
data
data_indexed
renditions
type
format
length
checksum
identifier
title

Relation**Index name:** .cara_relation**Attributes:**

type
description
integrity_kind
parent_type
parent_root_version_id
parent_id
binding_type
child_type
child_root_version_id
child_id
child_label
retain_across_versions
order_no
tags

ACL**Index name:** .cara_acl**Attributes:**

acl_name
acl_description
accessors
name
permission

Config**index name:** .cara_config**Attributes:**

config_type
id
description
icon
version
status

definition
modifier
modified
tag

Queue Item

Index name: .cara_queue_item

Attributes:

queue_name
type_name
object_id
parameters
priority
user_name
created
status
status_date
status_message

User Type Settings

Index name: .cara_user_type_settings

Attributes:

user_name
type_name
recent_items
subscriptions

User Tag

Index name: .cara_user_tag

Attributes:

user_name
type_name
name
color
accessors
items

Monitoring Log**Index name:** .cara_monitoring_log-{now/M{YYYY_MM}}

15. Configuration Files

Below are the main CARA server configuration files.

See the CARA installation guide for further details.

CARA Primary Configuration File: cara.yml

Please note that numbered explanatory notes have been added below, these are not present in the file when deployed:

#CARA Base URL

appBaseUrl: <https://some.domain.com/cara/>

1. Base CARA URL - used in notifications and generally in links generated by the system, required

#CARA cluster name - needs to be changed to a unique value

clusterName: develop

2. Cluster Name - unique cluster identified, required

#Elasticsearch connection

elasticsearch:

3. Elasticsearch/OpenSearch connection config, required

scheme: http

hostname: localhost

port: 9200

userName: admin

password: !keystore es.password

redis:

4. Redis connection config, required

url: localhost

port: 6379

useSSL: false

mq:

5. Message Broker config

brokerType: RabbitMQ

#Possible values: RabbitMQ or ActiveMQ

6. Message Broker Type - RabbitMQ is recommended, ActiveMQ can be used as well

nodes:

7. Message Broker Nodes

```
host: b-8e330066-9d67-465e-b400-f24f2f947591.mq.us-east-1.amazonaws.com
port: 5671
useSSL: true
username: user
password: password
authentication:
```

8. Enabled Authenticators - inline and token are required, ldap is optional, but those default values are recommended being enabled

authenticators:

- inline
- ldap
- token

#Cache timeouts

cache:

9. Configuration Cache in minutes, in production it is recommended to keep it as 60, in dev it can be lower

config: 10

#Configuration cache in minutes

stagingFolder: /opt/cara/staging

#Content-related options content:

10. Staging Folder Path - required for bulk import

skipChecksumVerification: false

#skip verification of content checksum when retrieving content

11. Checksum Verification - option to disable file checksum verification

auditing:

disabled: false

12. Auditing - option to disable writing audit trails

enabledEvents:

13. Auditing - Enabled Events - list of events that should be audited, optional

- _create
- _update

disabledEvents:

14. Auditing - Disabled Events - list of events that should not be audited, optional

```
- _authentication_system  
scripting:  
  sandboxEnabled: false
```

15. Sandbox Enabled - enable the script execution sandbox which encapsulates the script execution and disallows using dangerous code, recommend always enabled
`timeout: 5`

16. Script Execution Timeout - time in minutes after which the script execution is terminated

`allowedPackages:`

17. Allowed Packages - list of packages allowed to be used in scripts

```
- com.fasterxml  
- com.springframework  
- com.apache.pdfbox  
- com.pdftron  
- org.slf4j
```

`allowedClasses:`

18. Allowed Classes - list of classes allowed to be used in scripts

```
- java.lang.Object  
- java.lang.Exception  
- java.lang.Throwable
```

`googleDocsConfig:`

```
enabled: false  
importFormats:  
  application/vnd.ms-powerpoint: "application/vnd.google-apps.presentation"  
  application/vnd.openxmlformats-officedocument.presentationml.presentation:  
    "application/vnd.google-apps.presentation"  
  text/tab-separated-values: "application/vnd.google-apps.spreadsheet"  
  application/vnd.sun.xml.writer: "application/vnd.google-apps.document"  
  application/vnd.ms-excel: "application/vnd.google-apps.spreadsheet"  
  application/vnd.openxmlformats-officedocument.spreadsheetml.sheet:  
    "application/vnd.google-apps.spreadsheet"  
  application/vnd.openxmlformats-officedocument.wordprocessingml.document:  
    "application/vnd.google-apps.document"  
  application/rtf: "application/vnd.google-apps.document"  
  application/msword: "application/vnd.google-apps.document"  
  application/vnd.oasis.opendocument.text: "application/vnd.google-apps.document"  
  text/plain: "application/vnd.google-apps.document"  
  text/csv: "application/vnd.google-apps.spreadsheet"  
  text/rtf: "application/vnd.google-apps.document"
```

application/x-vnd.oasis.opendocument.spreadsheet:
"application/vnd.googleapps.spreadsheet"

File System Connector: cara_connector-fs.yml

```
#Default content store path
path: /tmp/data
#Whether to encode file content, if true, the keystore must be provided
encryptionEnabled: false
#Algorithm: AE/DES/DESede
encryptionAlgorithm: AES
#Mode: CTR/CBC/GCM
encryptionMode:CBC
#Padding: NO_PADDING, PKCS5_PADDING
encryptionPadding: PKCS5_PADDING
encryptionPassword:
```

Amazon S3 Connector: cara_connector-s3.yml

```
#Default content store path
bucketName: s3-bucket-name
#AWS region name
regionName: us-east
#Whether to encode file content, will utilize server-side S3 encryption
encryptionEnabled: false
```

Box Connector: cara_connector-box.yml

```
verboseLogging: true
boxSettings:
# each entry contains Box configuration for specific appId + document type;
#special entry '*' can be used to match any type and will be used as fall-back in case
there is no type-specific configuration), for example:
#
#app1/pubmed_fs: ...
#app1/*: ...
```

```
#/hr_docs: ...
#!/*: ...
/*/*:
caraUsersGroupName: ""
caraUsersGroupId: ""
clientId: ""
clientSecret: ""
publicKeyId: ""
privateKey: ""
passphrase: ""
enterpriseld: ""
```

Azure Blob Storage Connector: cara_connector-azure-blob-storage.yml

```
containerName:
connectionString: ""
```

LDAP Authentication Configuration: ldap-ad-authentication.yml

```
id: ldap-ad1
type: ldap
```

authentication:

```
# This flag enables use of this LDAP subsystem for authentication. It may be
# that this subsystem should only be used for synchronization, in which case
# this flag should be set to false.
```

```
active: true
```

```
# This properties file brings together the common options for LDAP authentication
rather than editing the bean definitions
```

```
#
```

```
allowGuestLogin: true
```

```
# How to map the user id entered by the user to taht passed through to LDAP
# In Active Directory, this can either be the user principal name (UPN) or DN.
```

```
# UPNs are in the form <sAMAccountName>@domain and are held in the
userPrincipalName attribute of a user
userNameFormat: "%s@domain"

# The LDAP context factory to use
initialContext: com.sun.jndi.ldap.LdapCtxFactory

# The URL to connect to the LDAP server
providerUrl: ldap://localhost:389

# The authentication mechanism to use for password validation
authenticationType: simple

# Escape commas entered by the user at bind time
# Useful when using simple authentication and the CN is part of the DN and contains
commas
escapeCommasInBind: false

# Escape commas entered by the user when setting the authenticated user
# Useful when using simple authentication and the CN is part of the DN and contains
commas, and the escaped \, is
# pulled in as part of an LDAP sync
# If this option is set to true it will break the default home folder provider as space
names can not contain \
escapeCommasInUid: false

# Comma separated list of user names who should be considered administrators by
default
defaultAdministratorUserNames: cadmin

# Enable FTP authentication using LDAP
authenticateFTP: true

# Set to 'ssl' to enable truststore configuration via subsystem's properties
(java.naming.security.protocol)
protocol:
```

```
# LDAPS truststore configuration properties
trustStore:
    path:
    passphrase:
    type:

security:
# This flag enables use of this LDAP subsystem for user and group
# synchronization. It may be that this subsystem should only be used for
# authentication, in which case this flag should be set to false.
active: true

# The authentication mechanism to use for synchronization
authenticationType: simple

# The default principal to bind with (only used for LDAP sync). This should be a UPN
or DN
principal: cn=directory manager

# The password for the default principal (only used for LDAP sync)
credentials: !keystore ldap.password

# If positive, this property indicates that RFC 2696 paged results should be
# used to split query results into batches of the specified size. This
# overcomes any size limits imposed by the LDAP server.
queryBatchSize: 1000

# If positive, this property indicates that range retrieval should be used to fetch
# multi-valued attributes (such as member) in batches of the specified size.
# Overcomes any size limits imposed by Active Directory.
attributeBatchSize: 1000

# The query to select all objects that represent the groups to import.
groupQuery: (objectclass=groupofuniqueNames)
```

```
# The query to select objects that represent the groups to import that have changed  
since a certain time.  
groupDifferentialQuery: (&(objectclass=groupofuniquenames)(!(whenChanged<={0})))  
  
# The query to select all objects that represent the users to import.  
personQuery: (&(title=GENERIS)(objectclass=inetOrgPerson))  
  
# The query to select objects that represent the users to import that have changed  
since a certain time.  
personDifferentialQuery: (&(objectclass=*)(!(modifyTimestamp<={0})))  
  
# The group search base restricts the LDAP group query to a sub section of tree on  
the LDAP server.  
groupSearchBase: ou=internal,ou=groups,dc=generiscorp,dc=com  
  
# The user search base restricts the LDAP user query to a sub section of tree on the  
LDAP server.  
userSearchBase: ou=users,dc=generiscorp,dc=com  
  
# The name of the operational attribute recording the last update time for a group or  
user.  
modifyTimestampAttributeName: whenChanged  
  
# The timestamp format. Unfortunately, this varies between directory servers.  
timestampFormat: yyyyMMddHHmmss'.0Z'  
  
# The attribute name on people objects found in LDAP to use as the unique user id in  
CARA  
userUniqueIdAttributeName: nsuniqueid  
  
# The attribute name on people objects found in LDAP to use as the uid in CARA  
userIdAttributeName: uid  
  
# The attribute on person objects in LDAP to map to the first name property in CARA  
userFirstNameAttributeName: givenName
```

```
# The attribute on person objects in LDAP to map to the last name property in CARA
userLastNameAttributeName: sn

# The attribute on LDAP user objects to map to the authority display name property in
CARA
userDisplayNameAttributeName: displayName

# The attribute on person objects in LDAP to map to the email property in CARA
userEmailAttributeName: mail

# The attribute on person objects in LDAP to map to the organizational id property in
CARA
userOrganizationalIdAttributeName: title

# The default home folder provider to use for people created via LDAP import
defaultHomeFolderProvider: largeHomeFolderProvider

# The attribute on LDAP group objects to map to the authority name property in CARA
groupIdAttributeName: cn

# The attribute on LDAP group objects to map to the authority display name property
in CARA
groupDisplayNameAttributeName: description

# The group type in LDAP
groupType: groupofuniqueNames

# The person type in LDAP
personType: inetOrgPerson

# The attribute in LDAP on group objects that defines the DN for its members
groupMemberAttributeName: uniqueMember

# If true progress estimation is enabled. When enabled, the user query has to be run
twice in order to count entries.
enableProgressEstimation: true
```

synchronization:

```
# The attribute on person objects in LDAP to map to the user account status  
(OpenLDAP: pwdAccountLockedTime, Oracle/RedHat/389: nsAccountLock)  
userAccountStatusAttributeName: nsAccountLock
```

```
# id of theaccount status interpreter, available options: ad, 389  
userAccountStatusInterpreter: openldap
```

common:

```
# Requests timeout, in miliseconds, use 0 for none (default)  
readTimeout: 0
```

