Opensalt Docs Documentation

Release 2.3

Jan 08, 2020

Contents

1 1.0 Audience **5**
- 1.1 1.1 Platform Requirements . . . 5

2 2.0 OpenSALT Structure and Access **7**
- 2.1 2.1 OpenSALT Structure . . . 7
- 2.2 2.2 OpenSALT Home Page . . . 7
- 2.3 2.3 Log into OpenSALT . . . 8
- 2.4 2.4 Log off from OpenSALT . . . 8

3 3.0 User Roles and Permissions **11**
- 3.1 3.1 Super User . . . 12
- 3.2 3.2 Super Editor . . . 13
- 3.3 3.3 Organization Admin . . . 13
- 3.4 3.4 Organization Editor . . . 14
- 3.5 3.5 Organization Reviewer . . . 14
- 3.6 3.6 Public User . . . 15

4 4.0 Organization Management **17**
- 4.1 4.1 Show Organizations . . . 18
- 4.2 4.2 Add/Create Organization . . . 18
- 4.3 4.3 Edit Organization . . . 19
- 4.4 4.4 Delete Organization . . . 19

5 5.0 User Management **21**
- 5.1 5.1 Change Password . . . 21
- 5.2 5.2 Manage Users . . . 22

6 6.0 Framework Management **27**
- 6.1 6.1 Navigate and View Frameworks . . . 27
- 6.2 6.2 OpenSALT Frameworks . . . 34
- 6.3 6.3 Framework Items . . . 45
- 6.4 6.4 Exemplars for Items . . . 63
- 6.5 6.5 Derivative Frameworks - Copy Items . . . 65
- 6.6 6.6 Association Management . . . 70
- 6.7 6.7 Export Frameworks Packages . . . 86

7 7.0 Options **95**

7.1 7.1 Commenting Module 95
7.2 7.2 User Account Self-Creation 96

OpenSALT v.2.2 - User Manual

Revision History

Name	Organization	Date	Revision Notes	Document Version
J Kaufman	PCG	8/2017	Original Draft for v1.0	.0
B Dorman	ACT	11/2017	Updating Import Guide	.1
J Marks	PCG	03/2018	OpenSALT v2.0	.2
B Dorman	ACT	05/2018	OpenSALT v2.1	.3
B Dorman	ACT	07/2019	OpenSALT v2.2	.4
B Dorman	ACT	12/2019	OpenSALT v2.3	1.0

Table of Contents

Introduction
1.0 Audience
2.0 OpenSALT Structure and Access
- *2.1 OpenSALT Structure*
- *2.2 OpenSALT Home Page*
- *2.3 Log into OpenSALT*
- *2.4 Log off from OpenSALT*

3.0 User Roles and Permissions
- *3.1 Super User*
- *3.2 Super Editor*
- *3.3 Organization Admin*
- *3.4 Organization Editor*
- *3.5 Organization Reviewer*
- *3.6 Public User*

4.0 Organization Management
- *4.1 Show Organizations*
- *4.2 Add/Create Organization*
- *4.3 Edit Organization*
- *4.4 Delete Organization*

5.0 User Management
- *5.1 Change Password*

5.2 Manage Users
5.2.1 View User
5.2.2 Add/Create User
5.2.3 Show User
5.2.4 Edit User
5.2.5 Suspend User
5.2.6 Reinstate User
5.2.7 Delete User
6.0 Framework Management
6.1 Navigate and View Frameworks
6.1.1 OpenSalt Home / Contents View Page
6.1.1.1 Public View
6.1.1.2 Credentialed View
6.1.2 Display Frameworks Page
6.1.2.1 Public View
6.1.2.2 Credentialed View
6.1.2.3 Log View
6.1.3 Adoption Status Security Features
6.2 OpenSALT Frameworks
6.2.1 Import Frameworks
6.2.1.1 Import CASE-Compliant (JSON) File
6.2.1.2 Import framework from Spreadsheet
6.2.2 Manually Create Frameworks
6.2.3 Edit Frameworks
6.2.4 Delete Frameworks
6.2.5 Update Frameworks from spreadsheet
6.2.5.1 Creating a new Framework from a template spreadsheet
6.2.6 Create new associations via Spreadsheet Update Features
6.2.6.1 Update Associations on the same server
6.2.6.2 Update Associations from an external server
6.2.7 Copy Framework
6.2.7.1 Copy Framework only
6.2.7.2 Copy entire framework and Make Derivative Framework
6.3 Framework Items
6.3.1 Import Items with OpenSALT Template
6.3.1.1 CSV Loading Guide for CASE
6.3.1.2 Import Process
6.3.1.3 Error Log
6.3.2 Formatting for FullStatement and Notes
6.3.3 Create Items Manually
6.3.4 Edit Items
6.3.4.1 Edit Item Content
6.3.4.2 Edit Item Positions
6.3.5 Delete Items
6.3.5.1 Delete Child Item
6.3.5.2 Delete Parent Item
6.4 Exemplars for Items

6.4.1 Add Exemplar
6.4.2 Delete Exemplar
6.5 Derivative Frameworks - Copy Items
6.6 Association Management
6.6.1 Associations
6.6.1.1 Create Associations
6.6.1.2 Edit Associations
6.6.1.2 Delete Associations
6.6.1.2.1 Delete Associations from Tree View
6.6.1.2.2 Delete Associations from Association View
6.6.2 Association Groups
6.6.3 Crosswalk Associations
6.6.3.1 Create Crosswalk Associations
6.6.3.2 Edit Crosswalk Associations
6.6.3.3 Delete Crosswalk Associations
6.7 Export Frameworks Packages
6.7.1 Export Competency Framework Package (JSON) File
6.7.2 Export Spreadsheet
6.7.2.1 CF Doc Tab
6.7.2.2 CF Item Tab
6.7.2.3 CF Association Tab
6.7.3 Link for Browser View
7.0 Options
7.1 Commenting Module
7.1.1 Features
7.1.2 Viewing Comments
7.1.3 Commenting Configuration
7.1.4 License
7.2 Automated User Sign up
7.2.1 Features
7.2.2 Process
7.2.3 Configuration
7.3 Additional Fields
7.4 File/Image Upload
7.5 Clone Feature

CASE

CASE establishes a new, global technical standard for the exchange of machine readable, linked data versions of state and national academic standards, local learning objectives and targets, or any workplace, military, or higher education competencies representing skills, knowledge, or abilities. To learn more please visit:

OpenSALT

To support CASE, PCG Education (A national public sector consulting practice); ACT Inc and SchoolCity (both leading curriculum and assessment providers), partnered on the development of the open source project called OpenSALT, that provides a free, IMS-conformance certified tool that enables education organizations to manage and publish frameworks and crosswalks to other standards.

Developed as an open source project, OpenSALT provides for easy exchange of state standards data and provides for

and enables alignment services to align educational content to learning standards, as well as correlate (Crosswalk) between different state standards and competency frameworks.

Key features that distinguish OpenSALT:

- Easily create derivative frameworks from national standards for use within States
- Create customized competency frameworks with connections to national and state frameworks as needed
- Set learning progressions within a framework
- Allow different views of a framework (For example, view the Standards of Engineering Practices in the NGSS framework separately from the DCI statements)
- Filter frameworks by keyword
- Enabled for the common alignment of resources and assessments to standards
- A Consistent digital format that can be easily integrated into ed-tech products and internal documents alike (Export to the standardized CASE Format as well as spreadsheets)

CHAPTER 1

1.0 Audience

This document is intended as a complete guide for using OpenSALT. This document is specially designed for all users with advanced permissions or non-specialists who may find the document useful as a point of reference. By reading this guide, you will learn how to use OpenSALT through the elements of the graphical user interface (GUI) and explanations of some of the advanced features to provide best practices. This guide will help you to navigate and easily use OpenSALT. The functions and features described in this guide will indicate any role restrictions or limitations as to what users can access or interact with the given system function.

1.1 1.1 Platform Requirements

The user interface is designed using current web standards and supports most modern browsers, in their current supported versions, on a variety of desktop and mobile devices.

The user interface requires JavaScript to be enabled and internet access to be available (no support for an offline mode is currently available).

The user interface should support the latest versions of the following web browsers on desktop and mobile devices:

- Chrome
- Internet Explorer
- Edge
- Safari
- Firefox

CHAPTER 2

2.0 OpenSALT Structure and Access

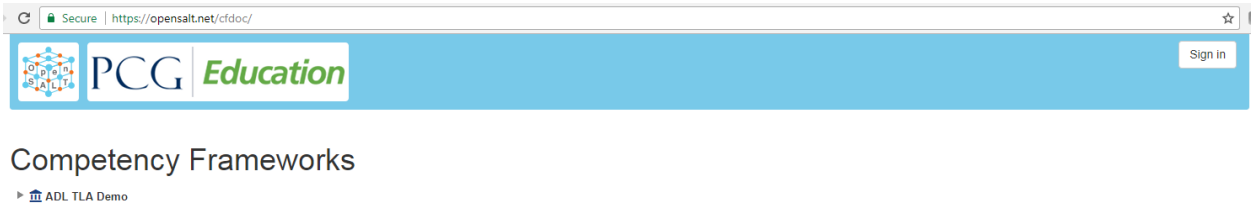
OpenSALT is designed to be an open, publicly accessible, framework site that allows transparency in frameworks and crosswalks. By design users are able to traverse and view the published frameworks and content by accessing the main url for the given site. OpenSALT then utilizes a login that allows provisioned users to create, edit and otherwise interact with the frameworks for the site.

2.1 2.1 OpenSALT Structure

OpenSALT goes to the home page by default, and also has a document tree view. The Doc Tree View shows a framework and information about the framework on the right hand side, where logged in users with proper permissions can edit items, associate with other framework items, and copy other framework items into the current framework.

2.2 2.2 OpenSALT Home Page

The image below is an example of an OpenSALT site, OpenSALT.net hosted by PCG. When user launches opensalt.net in a browser, the user is presented with the open view of the site.

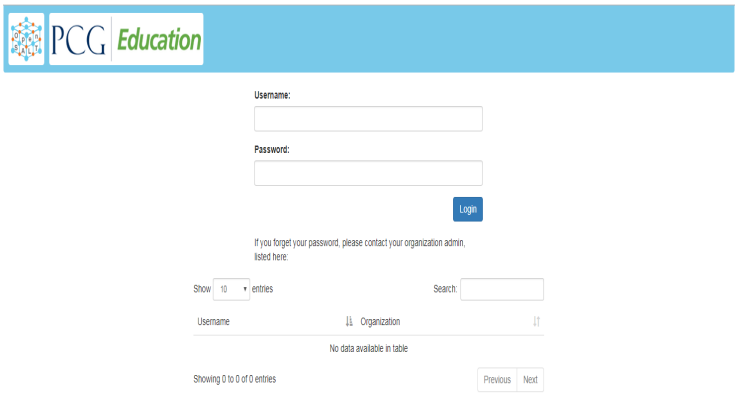


This is a view of an OpenSALT site hosted by ACT, located at frameworks.act.org

All public, non-provisioned users are able to view the frameworks and content with each form this view. Navigating the frameworks will be further discussed in a later section of this document.

2.3 2.3 Log into OpenSALT

For provisioned users the **Sign in** button will launch a login screen that will enable deeper system usage and open up available menus for further interaction with OpenSALT.

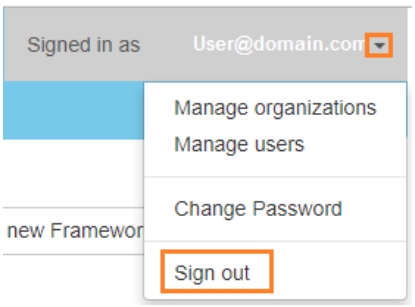


Provisioned users can enter in their *Username* (user's email) and *Password* then click the **Login** button to access the admin functions for OpenSALT.

After logging in a provisioned user will now see the *Sign In* button replaced by "Signed in as *username*" (email) with a menu icon and two new buttons on the main frame of the page: **Create a new Framework** and **Import Framework**

2.4 2.4 Log off from OpenSALT

All provisioned users should log out of OpenSALT when finished with their session. To logout, click on the expansion menu next to the user name and select **Sign out**.


Signed in as
User@domain.com
Manage organizations
Manage users
Change Password
new Framewor
Sign out

CHAPTER 3

3.0 User Roles and Permissions

OpenSALT currently has five user roles with specified permissions for the tool:

1. Super User
2. Super Editor
3. Organization Administrator
4. Organization Editor
5. Organization Reviewer (Account in org but no role)
6. Public (non-authenticated)

The Chart below provides an overview of the users and their provisioned roles and detailed descriptions follow.

User Type	Logged in User	Editor	Org Admin	Super Editor	Super User
Access Frameworks	X	X	X	X	X
Edit a framework in their own organization		X	X	X	X
Edit a framework		X	X	X	X
Export frameworks	x	x	X	X	X
View Comments	x	x	X	X	X
Manage Framework rights			X	X	X
Manage framework ownership			X	X	X
Edit Metadata across orgs				X	X
Add/edit Users within an Org			X		X
Add/Edit Orgs					X

Additionally, there are built in permissions regarding which users can edit organization-owned frameworks.

3.1 3.1 Super User

This is the top level in the user hierarchy for OpenSALT. A Super User has full permissions to the site and can create and manage organizations, manage users across all organizations, and manage all frameworks and content across all organizations.

- view any framework
- download (export) any framework in *Draft* or *Published* modes
- change their password
- add new, import, and edit frameworks in all organizations:
  - Personal frameworks
    - only editable by the creator (by default)
    - access can be granted to other editors to edit the framework
  - Organizational frameworks
    - editable by all editors/admins in the organization (by default), this includes Super Editors and Super Admins
    - can edit frameworks which they have permission to edit:
      - frameworks owned by any organization
      - other frameworks where the user has been explicitly been granted edit access
    - alter edit access to any frameworks
    - assign ownership of any personal framework created by an editor in any organization
- manage organizations in OpenSALT
  - add organizations
  - edit organizations
  - delete organizations
- add users to any organization (Super User, Super Editor, Organization Admin or Organization Editor)
  - suspend users in any organization (Super User, Super Editor, Organization Admin or Organization Editor)
  - unsuspend users in any organization (Super User, Super Editor, Organization Admin or Organization Editor)
- Note the Super Users cannot be excluded from edit access to a framework

3.2 3.2 Super Editor

Much like the Super User, the Super Editor has has permissions to work across organizations however this user is only permitted to manage all frameworks and content across all organizations. The Super Editor has no access to manage organizations, nor users.

The Organization Admin can manage users for their given organization, and manage and manage all frameworks and content for only their organization.

- view any framework
- download (export) any framework in *Draft* or *Published* modes
- change their password
- add new, import, and edit frameworks in all organizations:
  - Personal frameworks
    - only editable by the creator (by default)
    - access can be granted to other editors to edit the framework
  - Organizational frameworks
    - editable by all editors/admins in the organization (by default), this includes Super Editors and Super Admins
    - can edit frameworks which they have permission to edit:
      - frameworks owned by any organization
      - other frameworks where the user has been explicitly been granted edit access
    - alter edit access to any frameworks
    - assign ownership of any personal framework created by an editor in any organization
- Note the Super Editor cannot be excluded from edit access to a framework

3.3 3.3 Organization Admin

The Organization Admin can manage users for their given organization, and manage and manage all frameworks and content for only their organization.

- view any framework
- download (export) any framework in *Draft* or *Published* modes
- change their password
- add new, import, and edit frameworks:
  - Personal frameworks
    - only editable by the creator (by default)
    - access can be granted to other editors to edit the framework
  - Organizational frameworks
    - editable by all editors/admins in the organization (by default), this includes Super Editors and Super Admins
    - can edit frameworks which they have permission to edit:

            - frameworks owned by their organization open to all editors (no exclusions)
            - frameworks owned by their organization when the user is not in the exclusion list
            - other frameworks where the user has been explicitly been granted edit access
        * alter edit access to any frameworks owned by their organisation (personal frameworks)
        * assign ownership of any personal framework created by an editor in their organization
- add users to their organization (Organization Admin or Organization Editor)
  - suspend users in their organization (Organization Admin or Organization Editor)
  - unsuspend users in their organization (Organization Admin or Organization Editor)

3.4 3.4 Organization Editor

The Organization Editor has only access to its respective organization and is only permitted to manage frameworks and content with the assigned organization. An Organization Editor has no access to manage organizations, nor users. An Organization Editor can:

- view any framework
- download (export) any framework in *Draft* or *Published* modes
- change their password
- add new frameworks and import frameworks:
  - Personal frameworks
    * only editable by the creator (by default)
    * access can be granted to other editors to edit the framework
  - Organizational frameworks
    * editable by all editors/admins in the organization (by default), this includes Super Editors and Super Admins
    * can edit frameworks which they have permission to edit:
      - frameworks owned by their organization open to all editors (no exclusions)
      - frameworks owned by their organization when the user is not in the exclusion list
      - other frameworks where the user has been explicitly been granted edit access

3.5 3.5 Organization Reviewer

The Organization Reviewer is an org member with no roles assigned. They can log in and access private draft frameworks in an organization. They can not edit or manage frameworks. An Organization Reviewer can:

- view any framework in an org, private or not
- download (export) any framework in *Draft* or *Published* modes
- change their password
- Add comments if comments are turned on
- They can **not** make personal frameworks.

3.6 3.6 Public User

The Public User is the default user for all who visit an OpenSALT site. This user has no provisioned credentials and therefore cannot log into the site to access administrative functionality Rather this user is able to navigate the published frameworks and content and has read-only rights to the information. The Public User can:

- view any framework
- download (export) any framework in *Draft* or *Published* modes

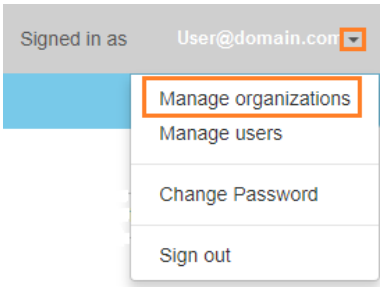
CHAPTER 4

4.0 Organization Management

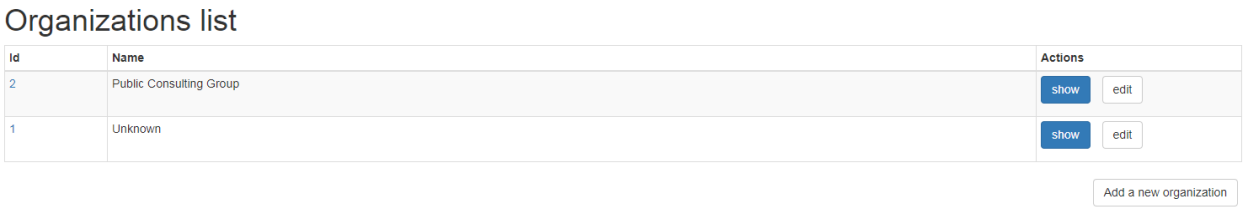
Audience: Super Users

As discussed in section 2.1 Open SALT is designed to b a multi-tenant environment housing multiple organizations and users, though keeping all securely separated. In order to establish the separation, the Super User will need to create and manage organizations.

To access the **Manage Organization** page, after logging in the user should click the triangular menu icon next to their user name in the header of the page. The menu will expand with several options depending on the user's role. Click on **Manage Organizations** to open the page.



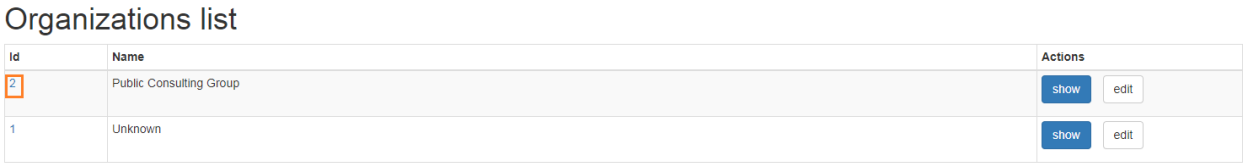
The **Organization List** is the default page displayed when a Super User accesses the **Manage Organization** function of OpenSALT. The list will display all organizations that the are currently in the given instance of OpenSALT. The **Organization List** provides the OpenSALT unique ID number, the Organization Name, and Actions for the organizations (**Show** and **Edit**). The page also has a button to **Add a new organization**.



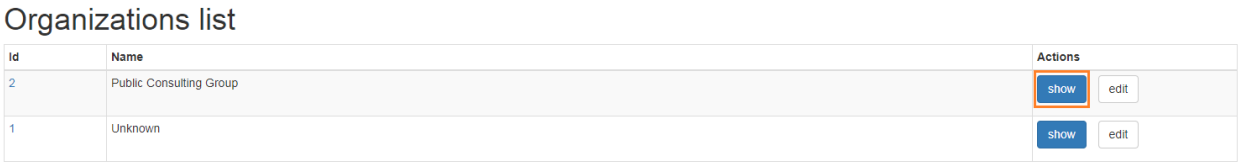
4.1 4.1 Show Organizations

There are two ways to view an individual organization's information beyond the **Organization list** table view.

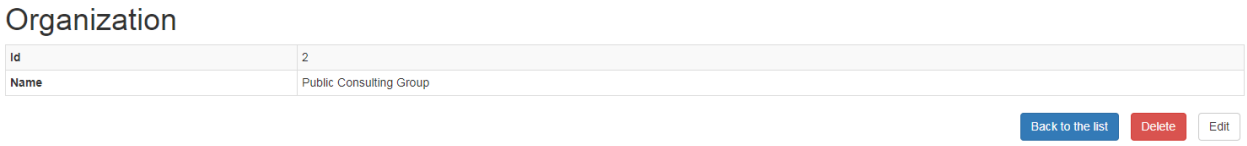
1. On the **Organization List** click on the linked **ID** number for the selected organization.



1. On the **Organization List** click on the **Show** button in the **Actions** section of the Organization table for the selected organization.



The **Organization** screen will display for the selected user.



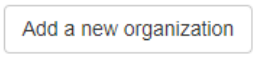
The Super User can see the Organization's unique ID value and the Organization's name.

From the **Organization** screen the Super User can:

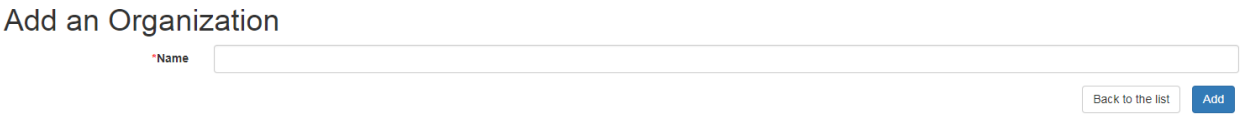
- return the **Organization List** by clicking the **Back to the list button**
- edit the organization by clicking on the **Edit** button
- delete the organization by clicking on the **Delete** button

4.2 4.2 Add/Create Organization

To create a new user, click on the **Add new user** button at the bottom right of the **User list**.



The Add an Organization page will display:



The admin will need to enter in the following required fields:

Name: Type the Organization name

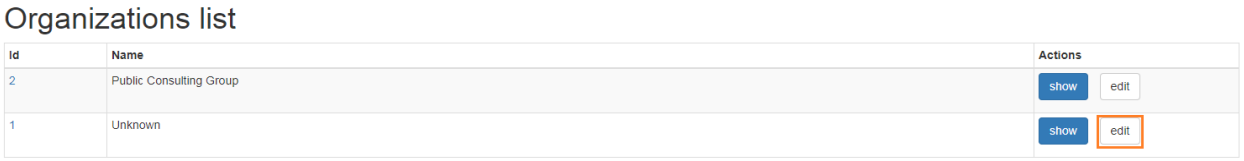
Then click on the **Add** button to create the organization.

If the Admin wants to cancel, and not create the organization, click the **Back to the list** button.

4.3 4.3 Edit Organization

An Admin can access the **Organization edit** screen using two paths:

1. On the **Organization List** click on the **Edit** button in the **Actions** section of the Organization table for the selected organization.



1. On the **Organization** screen click on the **Edit** button.

The **Organization edit** screen will display and allow the Admin to update the user's information.

The Admin can update/correct the following field:

Name: Type the Organization name

Then click on the **Save** button to save the changes.

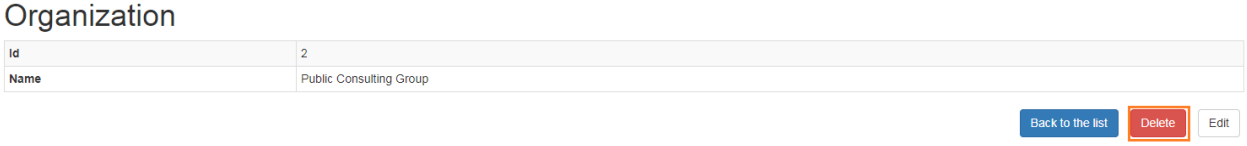
If the Admin wants to cancel, and not edit the the organization, click the **Back to the list** button.

The Admin can also delete the organization from this screen by clicking on the **Delete** button.

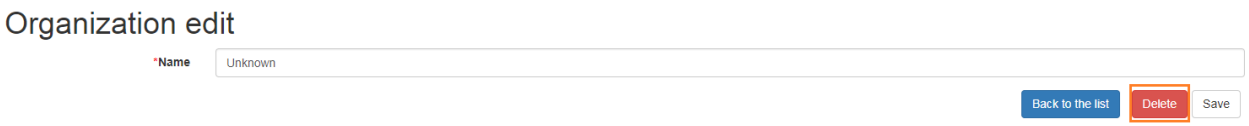
4.4 4.4 Delete Organization

If an organization needs to be completed removed from OpenSALT, the Admin should delete the organization. There are two methods to delete an organization.

1. From the **Organization List** click on the **Show** button for the selected organization. The **Organization** page will be displayed and the Admin can click the **Delete** button to terminate the organization. Once deleted the organization cannot be restored. If the organization is needed, the Admin will need to create a new organization.



1. From the **Organization List** click on the **Edit** button for the selected organization. The **Organization edit** page will be displayed and the Admin can click the **Delete** button to terminate the organization. Once deleted the organization cannot be restored. If the organization is needed, the Admin will need to create a new organization.


Organization edit
*Name
Unknown
Back to the list
Delete
Save

CHAPTER 5

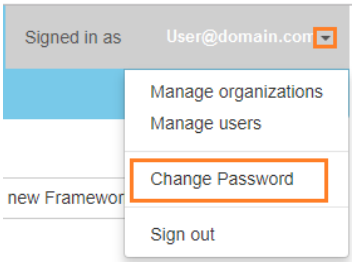
5.0 User Management

5.1 5.1 Change Password

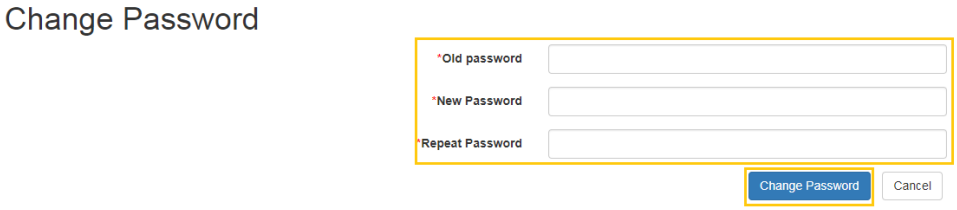
Audience: Super Users, Super Editor, Organization Admins, Organization Editor

All provisioned users have the ability to change their password from the temporary one created by the Admin or as part of a good practice to regularly update their password to maintain a good security protocol.

To access the **Change Password** page, after logging in the user should click the triangular menu icon next to their user name in the header of the page. The menu will expand with several options depending on the user's role. However all provisioned users will have the menu option: **Change Password**. Click on **Change Password** to open the page.



On the **Change Password** page the user will enter in the following required fields:



- **Old Password:** Existing current password
- **New Password:** Newly selected password

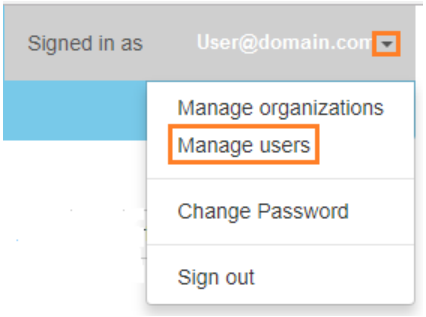
- **Repeat Password:** Re-type the newly selected password
- The user will then click the **Change Password** button.

If the entered values validate as correct, the old password is equal to the user's current password and the new and repeat passwords are the same, the system will change the user's password to the newly selected value. If alny of the data is not validated, the system will prompt the user to correct the information before the change password can be completed.

5.2 5.2 Manage Users

Audience: Super Users and Organization Admins

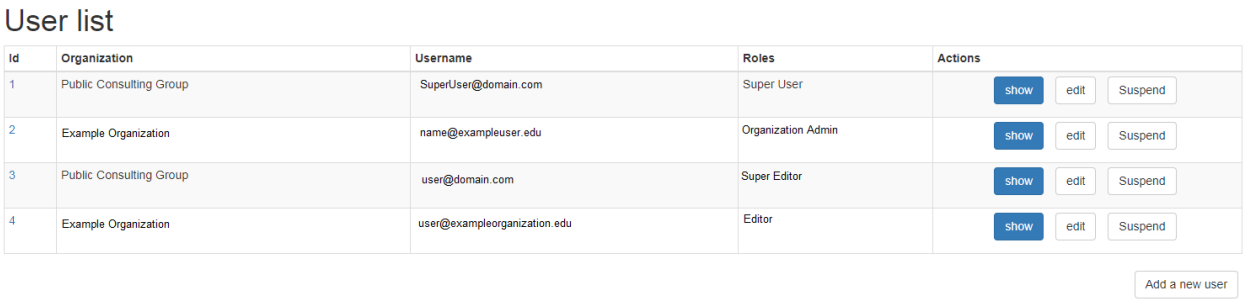
User account management is controlled by the Super Users and the Organization Admins. To access the **Manage Users** page, after logging in the user should click the triangular menu icon next to their user name in the header of the page. Click on **Manage Users** to open the page.



5.2.1 5.2.1 View User

Audience: Super Users and Organization Admins

The **User List** is the default page displayed when a Super User or Organization Admin accesses the **Manage User** function of OpenSALT. The list will display all users that the are currently in the given instance of OpenSALT. Super Users will see all users in all organizations, whereas Organization Admins will only see the users within their organization. The User List will display the OpenSALT unique ID for the user (as system generated incremented number), the Organization the user belongs to, the Username (email) The user's role, and available actions (**show**, **edit**, **Suspend**) for the user. The page also has a button to **Add a new user**.

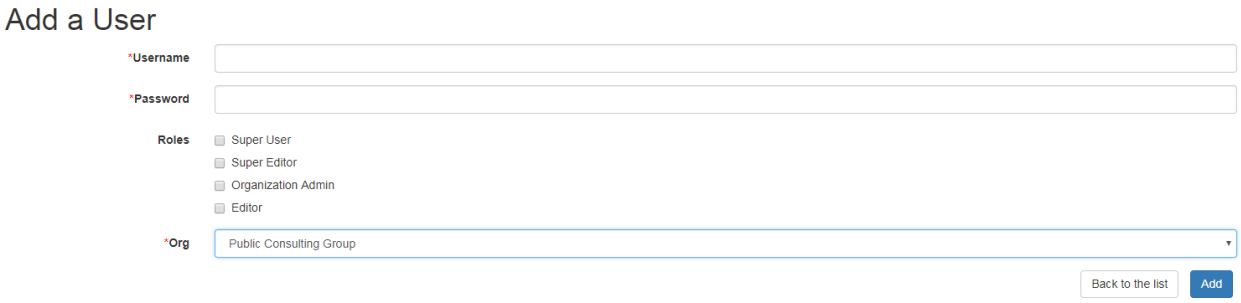


5.2.2 5.2.2 Add/Create User

Audience: Super Users and Organization Admins

To create a new user, click on the **Add new user** button at the bottom right of the **User list**.

The **Add a User** page will display:



The admin will need to enter in the following required fields:

Username: Type the user's email address

Password: Type in a temporary password for the user. This password will not be viewable after creating the user. The Admin needs to take note of the temporary password entered to share with the user and the user will need to enter this password to change their password to a non-temporary secure password.

Role: Select one role for the user and check the box accordingly

Org: Select the user's organization (note Organization Admins will only have their organization displayed, whereas Super Users will see all Organizations within the given OpenSALT site)

Then click on the **Add** button to create the user.

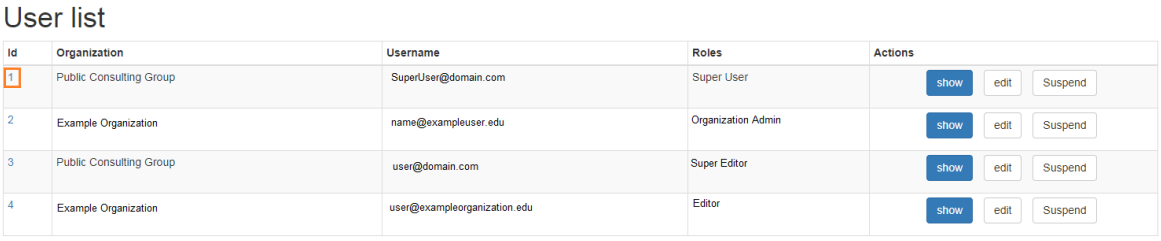
If the Admin wants to cancel, and not create the user, click the **Back to the list** button.

5.2.3 5.2.3 Show User

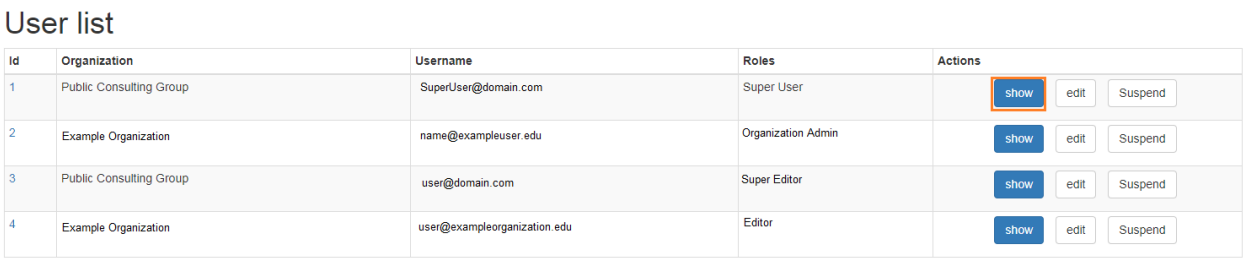
Audience: Super Users and Organization Admins

There are two ways to view an individual user's information beyond the **User list** table view.

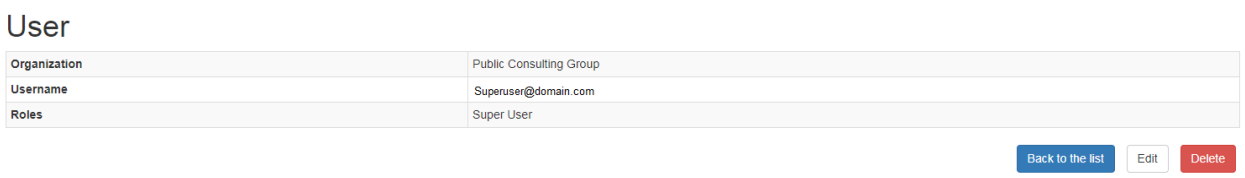
1. On the **User List** click on the linked **ID** number for the selected user.



1. On the **User List** click on the **Show** button in the **Actions** section of the User table for the selected user.



The **User** screen will display for the selected user.



The Super User or Organization Admin can see the user's Organization, Username, and assigned Role.

From the **User** screen the Admin can:

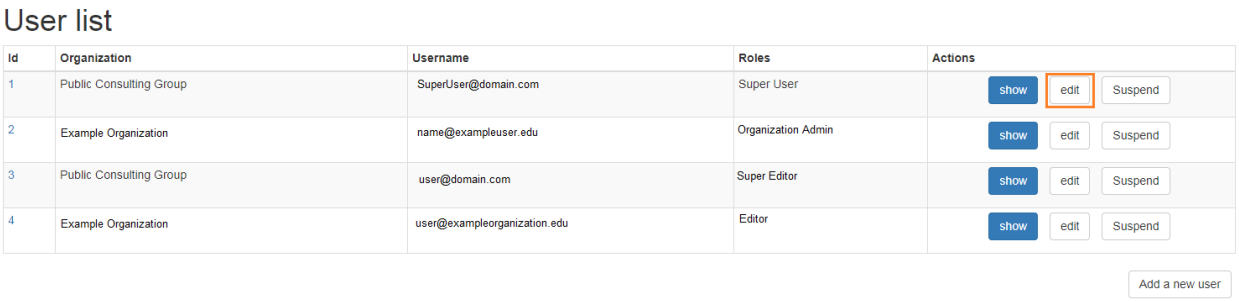
- return the **User List** by clicking the **Back to the list button**
- edit the user by clicking on the **Edit** button
- delete the user by clicking on the **Delete** button

5.2.4 5.2.4 Edit User

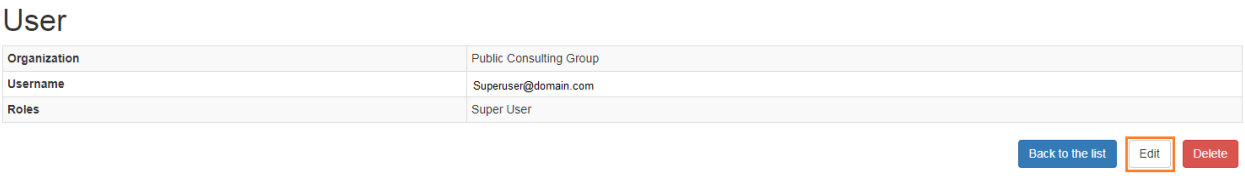
Audience: Super Users and Organization Admins

An Admin can access the **User edit** screen using two paths:

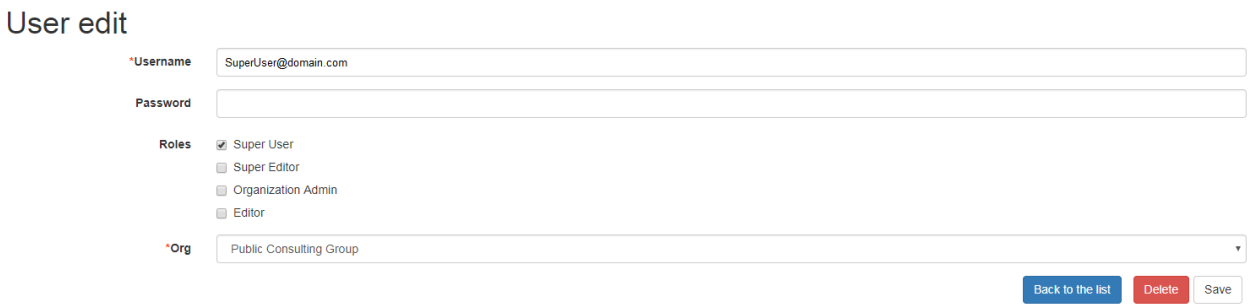
1. On the **User List** click on the **Edit** button in the **Actions** section of the User table for the selected user.



1. On the **User** screen click on the **Edit** button.



The **User edit** screen will display and allow the Admin to update the user's information.



The Admin can update/correct the following fields:

> *Username*: If needed, type the user's new email address. This will change the username that is entered when the user logs in.
>
> Password: If needed, type in a new temporary password for the user. This password will not be viewable after creating the user. The Admin needs to take note of the temporary password entered to share with the user and the user will need to enter this password to change their password to a non-temporary secure password.
>
> *Role*: If needed, select a new role for the user and uncheck the previous role.
>
> *Org*: If needed, change the organization Select the user's organization (note Organization Admins will only have their organization displayed, whereas Super Users will see all Organizations within the given OpenSALT site)

Then click on the **Save** button to save the changes.

If the Admin wants to cancel, and not edit the the user, click the **Back to the list** button.

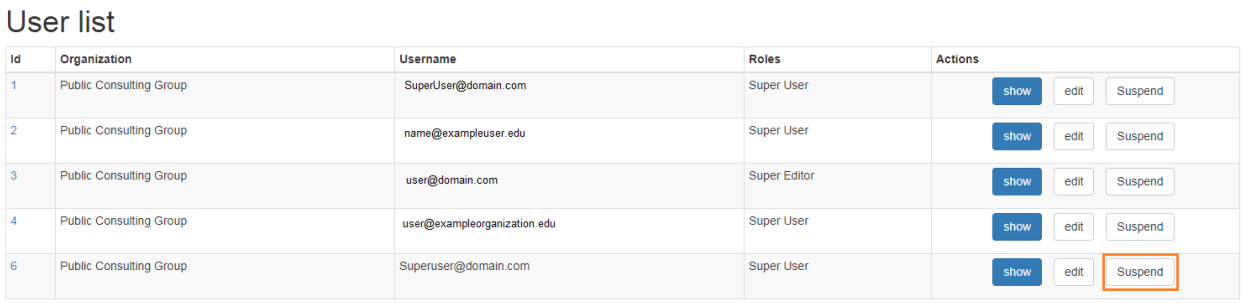
The Admin can also **delete** the user from this screen by clicking on the **Delete** button.

5.2.5 5.2.5 Suspend User

Audience: Super Users and Organization Admins

If an admin needs to prevent the user from accessing the OpenSALT instance, but needs to maintain the user's account in the system ,the admin can **Suspend** the user's account.

To suspend an account the Admin will need to be on the User List page and select the **Suspend** button for the selected user.



After clicking on the **Suspend** button, the user will be immediately suspended. The **User list** is updated to reflect the suspension and the account can now only be viewed through the **Show** button or unsuspended with the **Unsuspend** button.

5.2.6 5.2.6 Reinstate User

Audience: Super Users and Organization Admins

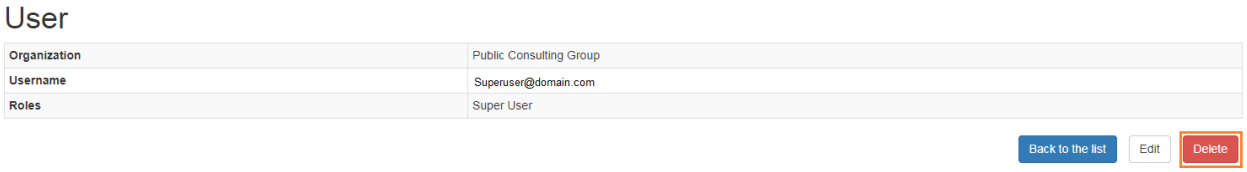
To reinstate a user's account the Admin will access the **User List** and click on the **Unsuspend** button for the selected user. The **User list** will update and the account will be immediately unsuspended and the user can log in with the original credentials. If the user needs to have their password reset, the Admin can the edit the user by clicking the **Edit** button for the selected user and updated the password.

5.2.7 5.2.7 Delete User

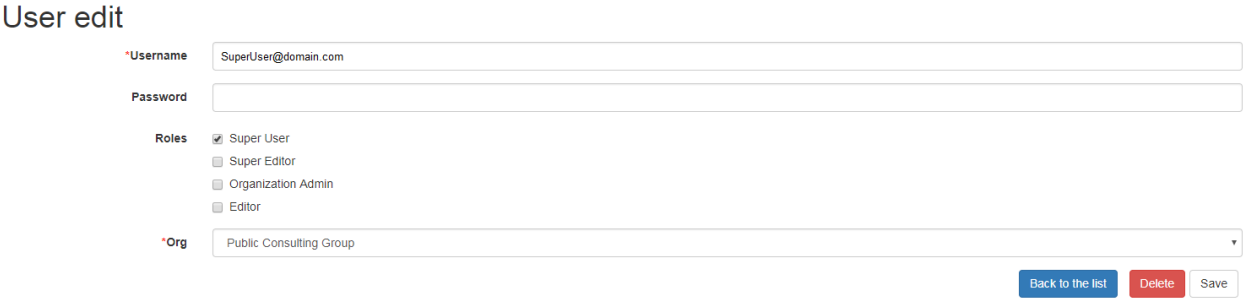
Audience: Super Users and Organization Admins

If a user needs to be completed removed from OpenSALT, the Admin should delete the user. There are two methods to delete a user.

1. From the **User List** click on the **Show** button for the selected user. The **User** page will be displayed and the Admin can click the **Delete** button to terminate the user account. Once deleted the account cannot be restored. If the account is needed, the Admin will need to create a new user account.



1. From the **User List** click on the **Edit** button for the selected user. The **User edit** page will be displayed and the Admin can click the **Delete** button to terminate the user account. Once deleted the account cannot be restored. If the account is needed, the Admin will need to create a new user account.



CHAPTER 6

6.0 Framework Management

6.1 6.1 Navigate and View Frameworks

OpenSALT is designed to be a simple application with few UI screens. The Application has two (2) primary screens for users to view and manage Frameworks and items:

1. OpenSalt Home / Contents View Page
2. Framework Display Page

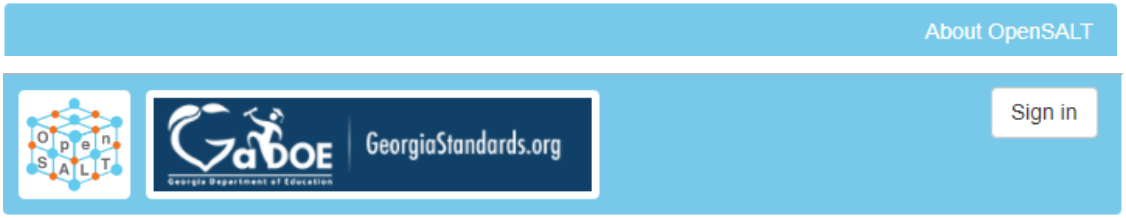
6.1.1 6.1.1 OpenSalt Home / Contents View Page

6.1.1.1 Public View

When a user launches an OpenSALT instance through their browser they will reach the Public view of the OpenSALT site. The following images represent a few of the current OpenSALT sites in operation.

Competency Frameworks

- CCSSO
- Norm Webb

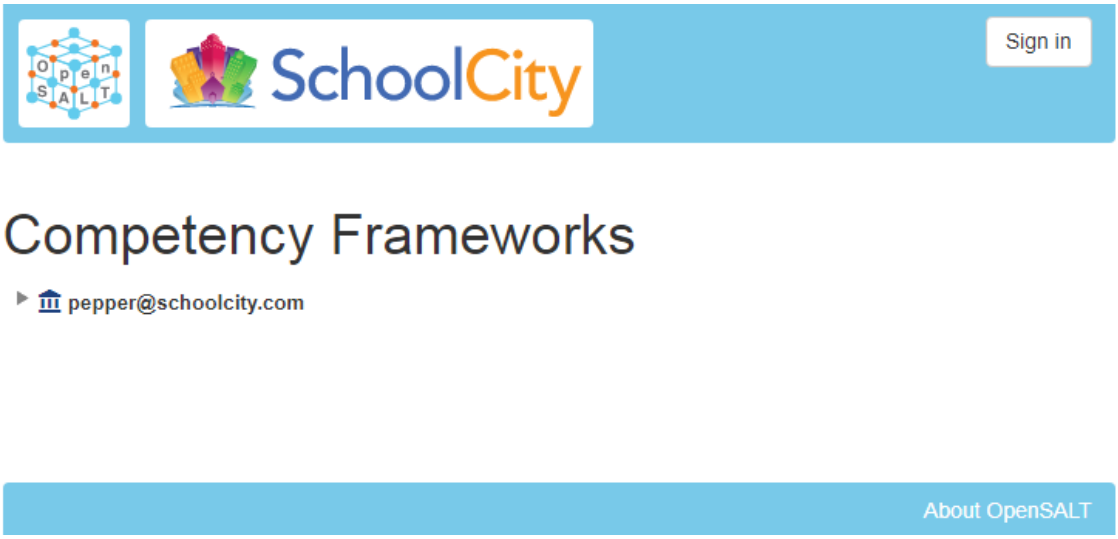


Competency Frameworks

- Georgia DOE

Competency Frameworks

- Australian National Curriculum
- OpenEd

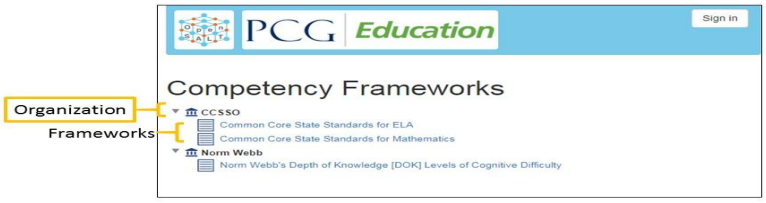


Note the default view has consistency from site to site. The variation is in the organization's ability to add their organization logo and the content of the frameworks in the list.

Public users can view all Organizations with Draft and Adopted Frameworks on the OpenSALT site.

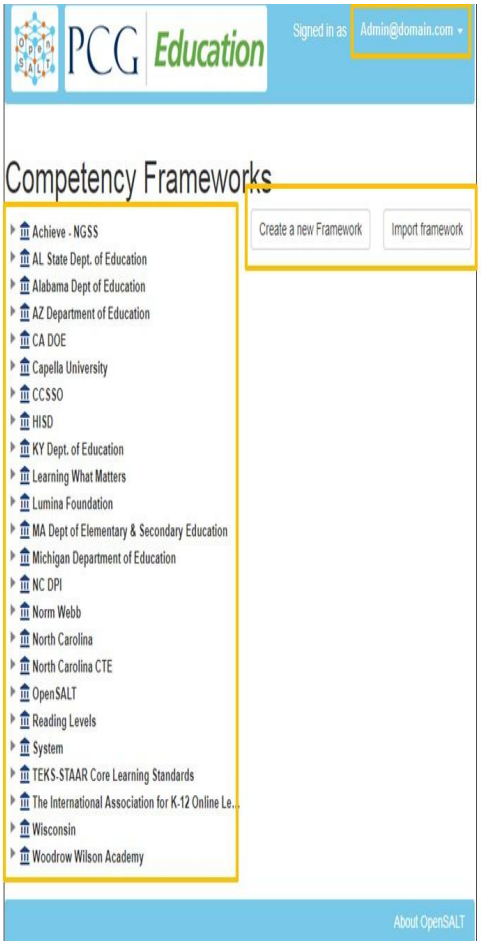
The default view is for all of the Organizations to be listed and their frameworks to be collapsed.

The user can expand the Organizations to see all available Frameworks by clicking on either the arrow to the left of the Organization name or on the name itself.



6.1.1.2 Credentialed View

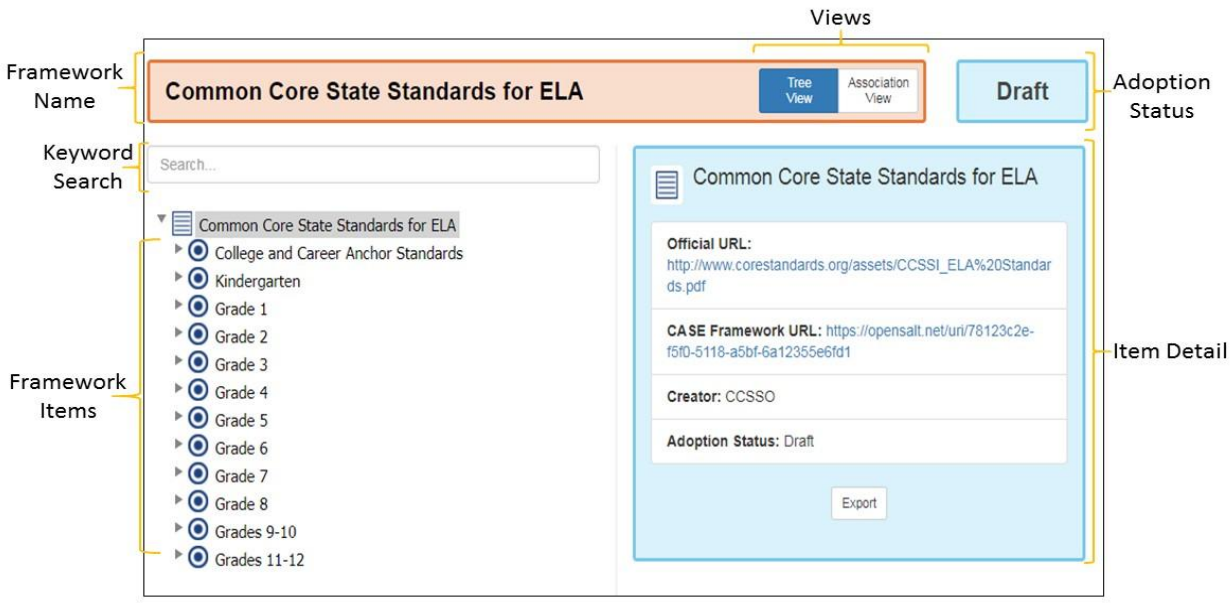
If a user has credentials and logs into OpenSALT they will have additional buttons on this page, depending on their role and permissions. Additionally they will be able to see all frameworks that are in a Private Draft status as well.



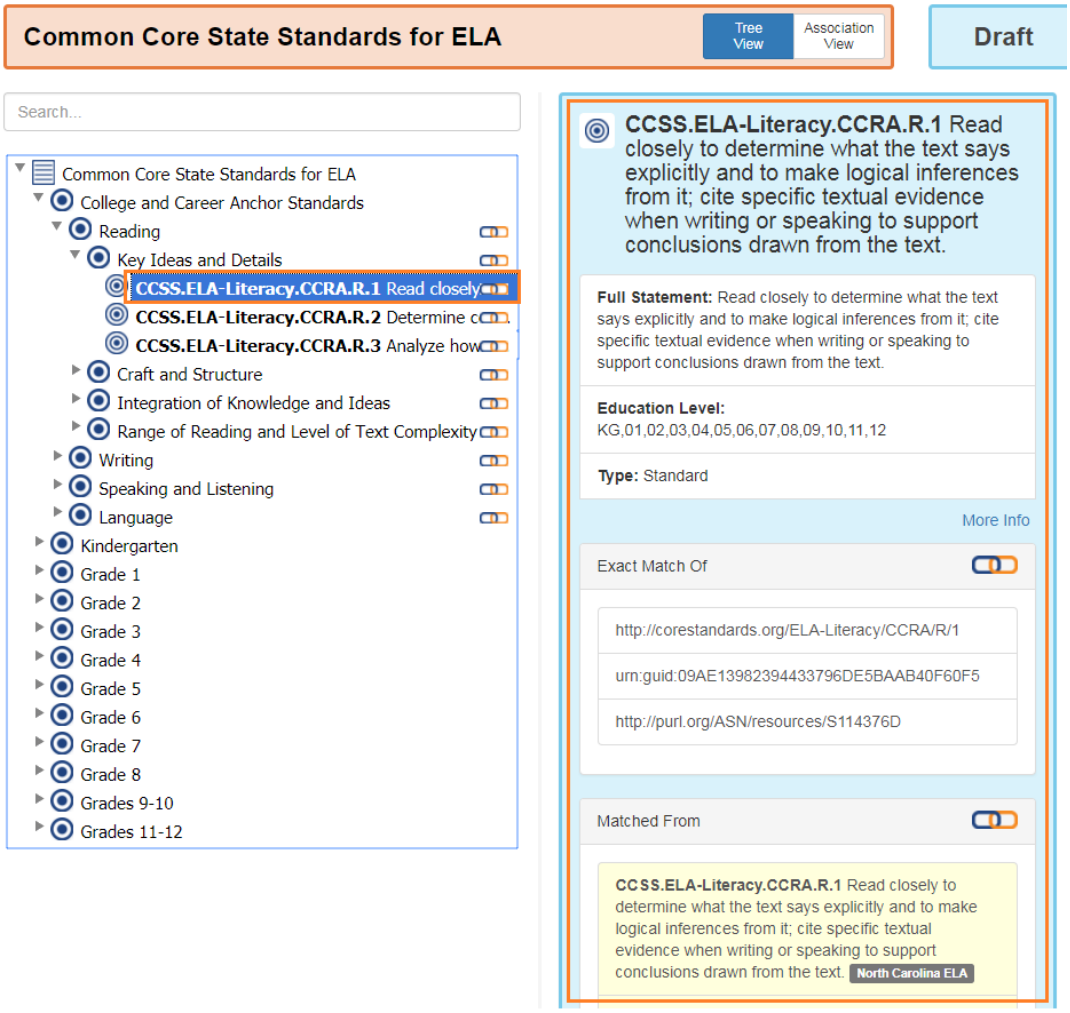
6.1.2 6.1.2 Display Frameworks Page

When a user click on a framework on the **OpenSalt Home / Contents View Page** OpenSALT will open the **Framework Display** page for the select framework.

6.1.2.1 Public View

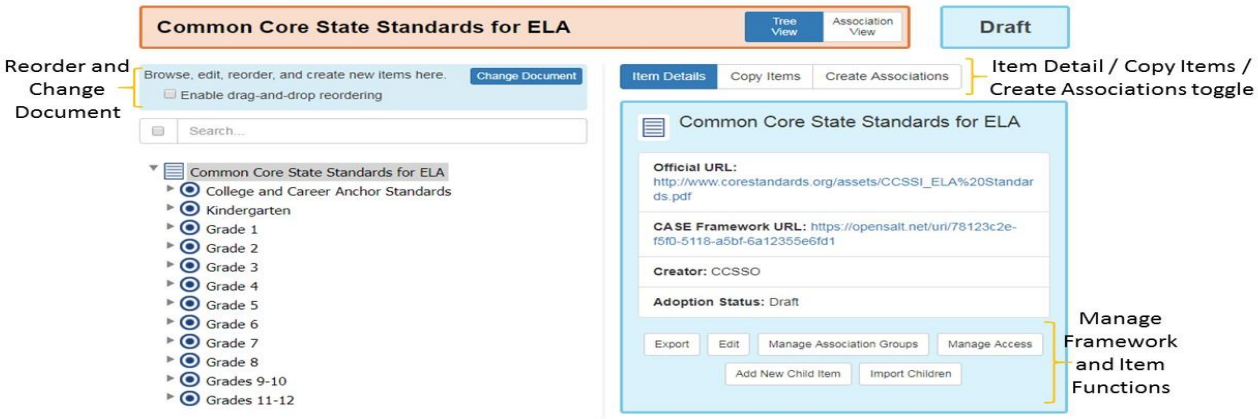


As a user clicks on a framework item, the Item Details frame will update to reflect the specifics for the selected item.



6.1.2.2 Credentialed View

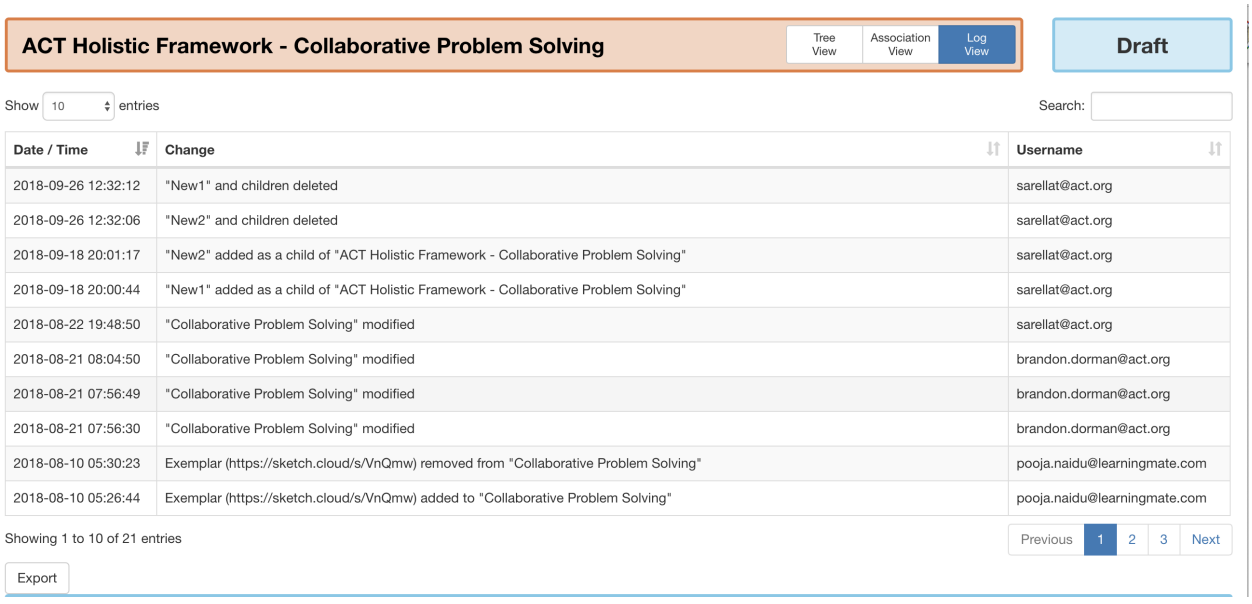
Credentialed users will have additional administrative controls and functions on the Display Framework page, depending on their role and permissions.



6.1.2.3 Log View

Logged in Editors will have additional insights into what changes have been made on a framework by selecting the Log View Tab.

This view provides a comprehensive history of changes made to the framework in general as well as a method of exporting as a CSV file.



6.1.3 Adoption Status Security Features

Draft Status	Private Draft	Draft	Adopted	Deprecated
Public Can View	Only with specific link	No	No	No
Org User can edit	Yes	Yes	No	No
Non-Org editor can edit	Must be shared	Must be shared	No	No

6.2 6.2 OpenSALT Frameworks

6.2.1 6.2.1 Import Frameworks

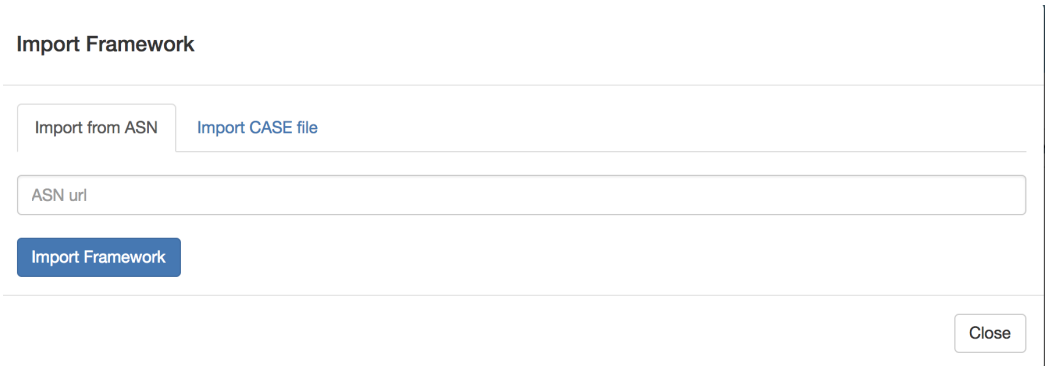
Audience: Super Users, Super Editor, Organization Admins, Organization Editor

OpenSALT has two methods for importing in an existing CASE-compliant framework into the site.

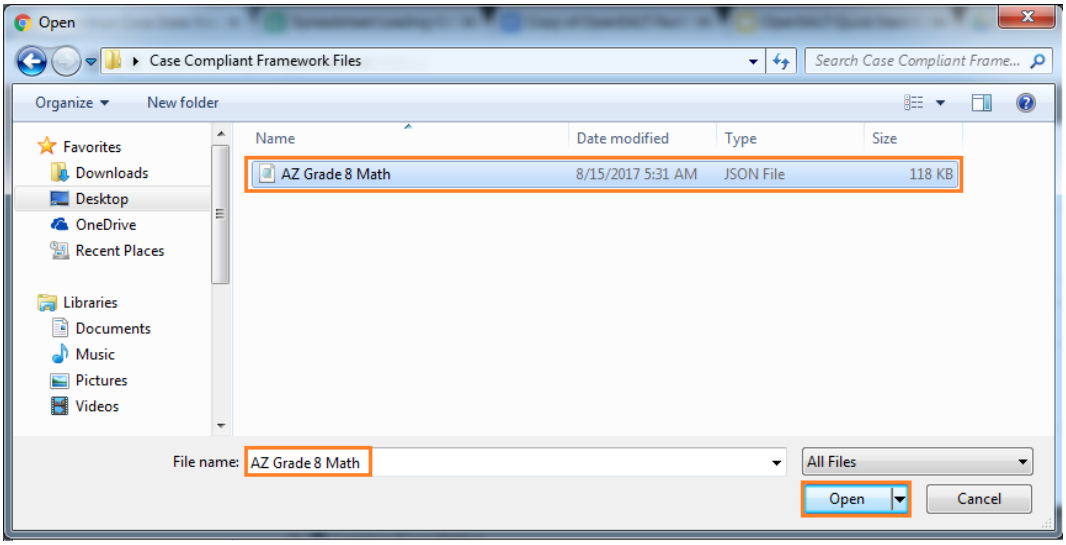
1) Import a CASE Compliant (JSON) file
2) Import a non-CASE compliant Spreadsheet file

6.2.1.1 Import CASE-Compliant (JSON) File

OpenSALT allows users to import known CASE-compliant framework files into the system. Click on the **Import CASE file** tab in the **Import Framework** window. Then click **Choose File.**



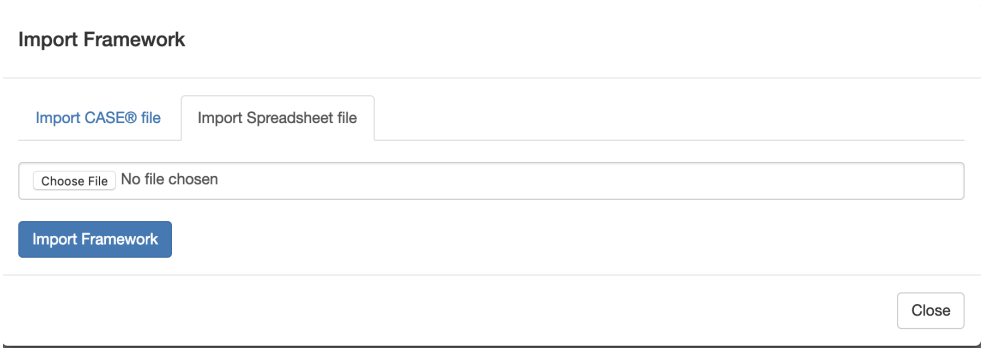
Navigate on your computer to the correct Case-compliant JSON file to import then click **Open**.



The filename will display in the Choose File box, then click the **Import Framework** button to import the framework.

To cancel the action and not import the file, click on the **Close** button.

6.2.1.2 Import Framework from spreadsheet



On the Import screen, you are able to import a spreadsheet. For the spreadsheet importer, FullStatement is the primary required field. The spreadsheet template is here:

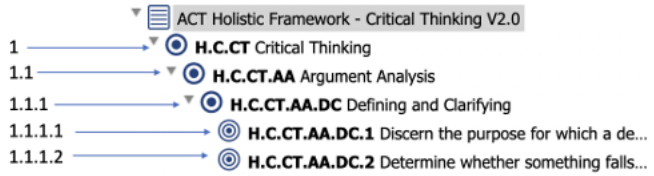
Additionally, a Sample Spreadsheet is available as well:

To create a basic framework via the spreadsheet import template, you must populate the FullStatement and SmartLevel in the CF Item sheet as well as Creator and Title on the CF Document sheet. The smart level field works like this:

The spreadsheet view of correctly entering a statement with smartlevel looks as follows:

fullStatement	humanCodingSch	smartLevel
Critical Thinking is: reasonably a	H.C.CT	1
Argument analysis is the use of l	H.C.CT.AA	1.1
Using definitions and clarificatic	H.C.CT.AA.DC	1.1.1
Discern the purpose for which a	H.C.CT.AA.DC.1	1.1.1.1
Determine whether something f	H.C.CT.AA.DC.2	1.1.1.2

On a live framework in OpenSALT, you would see this represented as:



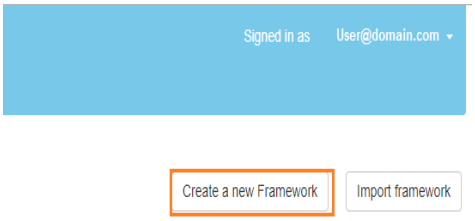
Once you have populated these two fields, upon import OpenSALT will create identifiers and relevant IsChildOf associations automatically.

If the framework you are importing already exists on the server, it will update that framework and let you know with a message.

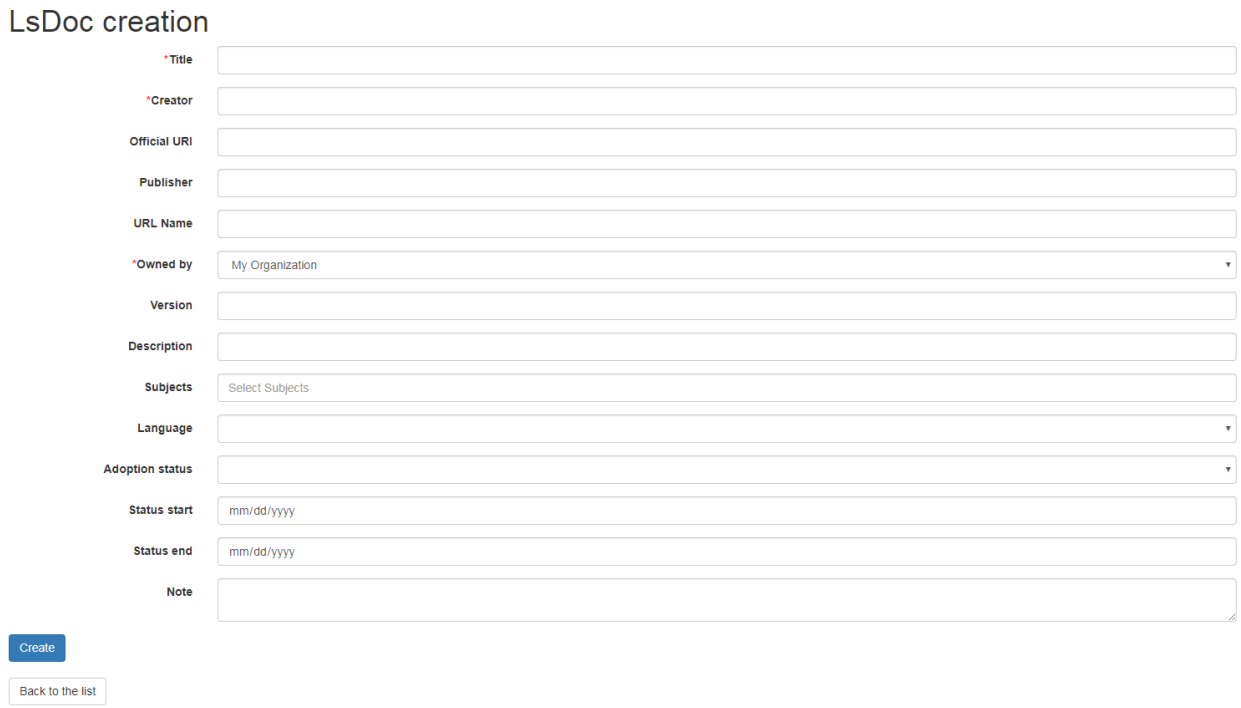
6.2.2 6.2.2 Manually Create Frameworks

Audience: Super Users, Super Editor, Organization Admins, Organization Editor

To Create a **Framework** manually the credentialed user will click on the Create a new Framework button on the **Framework Contents/OpenSalt Homepage** after logging into OpenSALT.



On the **Framework Creation** page, completed at a minimum the mandatory fields, though all fields should be filled in by best practice.



The following fields will display on the **LsDOC Creation** page:

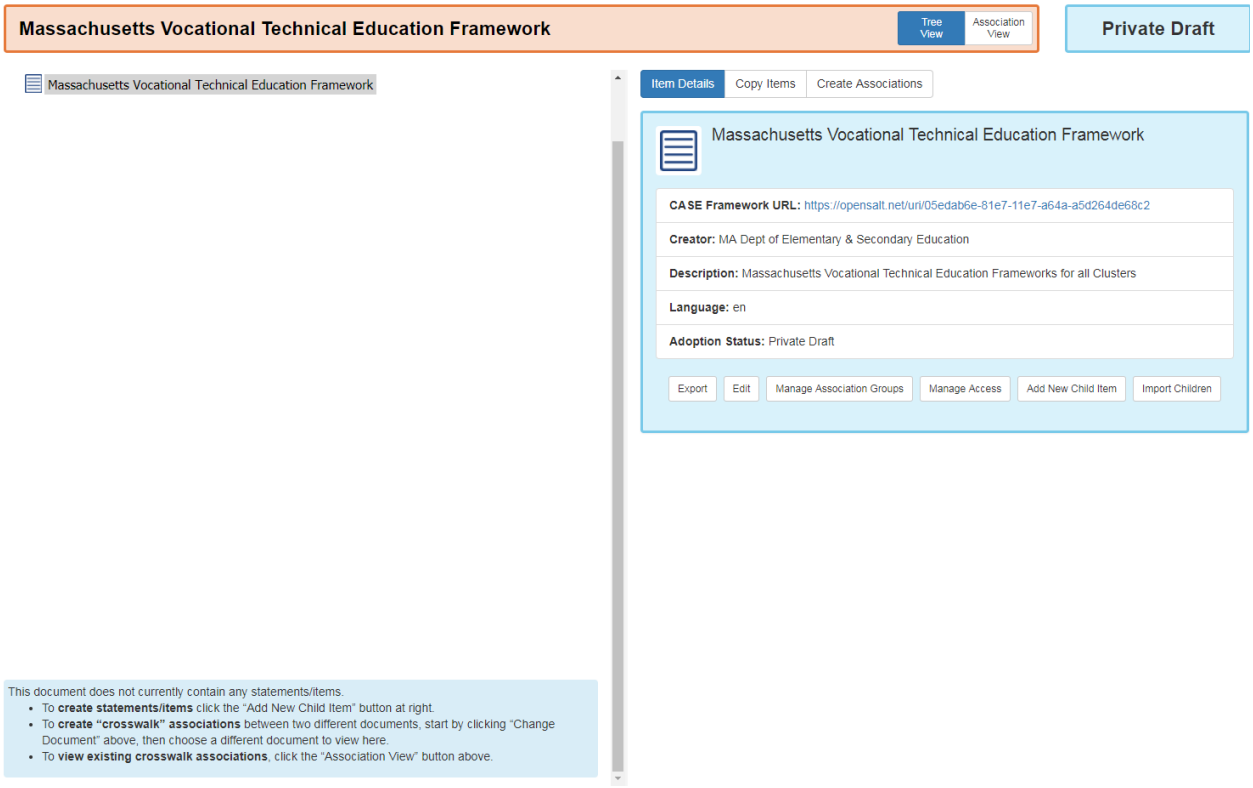
- *Title:* The title as it appears on the cover of the Official Source artifact, although it may be a title created by the Publisher. This is a mandatory field in OpenSALT. .
- *Creator:* The the entity that authorized or created the competency framework. It could be an education agency, higher education institution, professional body. It is the owner of the competency framework (e.g CCSSO, TEA, NGSS). This is a mandatory field for OpenSALT and will act as the Organization Folder on the Framework Contents/OpenSALT Home page.
- *Official URI:* The URL of the artifact adopted by the Standard Setting Entity. Often this document is published in html and/or as pdf and is used by the standard setting entity as part of its approval process. Since it is not the intent of this specification to fully reproduce the human-facing content and formatting of the source document, it is recommended that this document be transmitted as part of the competency framework package. This is an optional field in OpenSALT, though best practices indicate it should be filled in.
- *Publisher:* The entity that loads and publishes the Framework. Note that in many cases, the Standard Setting Entity may lack technical capabilities to publish the Competency Framework in a standard format so a third party may be displayed. This is an optional field in OpenSALT, though best practices indicate it should be filled in.
- *URL Name:* This field allows users to enter in a user friendly URL name. example: vs
- *Owned By:* Users will have the option to select from a few choices to indicate the Organization that created the Framework in OpenSALT.
  - *Me* - Private Framework created by an individual user
  - *My Organization* - Default to the user's organization
  - *Other Organization (Named in the Dropdown)* - Super Editor or Super User can select any Organization in OpenSALT.
- *Version:* This is used to separate any version information expressed by the Official Source artifact. Once and CF Pkg has been approved and published, any changes to an CF Item will constitute a new version of the CF Doc. This is an optional field in OpenSALT. If best practices are not followed, this field may be blank.

- *Description:* The description is typically created by the the Publisher as a standard description of the Competency Framework.This is an optional field in OpenSALT. If best practices are not followed, this field may be blank.
- *Subjects:* This is a string expressing the general subject area of the Competency Framework (e.g. Mathematics). This is an optional field in OpenSALT. If best practices are not followed, this field may be blank.
- *Language:* HTML Language Country Code VIA- country code from . This is an optional field in OpenSALT, though best practice indicates the filled should be filled in. However OpenSALT assumes English if not other language value is entered.
- *Adoption Status:* Adoption status displays the Framework's current status as Draft, Private Draft, Adopted, or Deprecated. OpenSALT assumes Adopted as the default if no status is specifically selected for the framework. This is an optional field in OpenSALT. If best practices are not followed, this field may be blank. OpenSALT assumes Adopted as the default if no status is specifically selected for the framework.
  - *Draft*: Able to be edited by Editors and Admins in an organization. Able to be viewed by the public.
  - *Private Draft*: Able to be viewed and edited by Editors and Admin in the owning organization
  - *Adopted*: Not able to be edited by Editors or Admin
  - *Depreciated:* Was once published, however it is now out of date
- *Status Start Date:* The date that the CF Doc status started. This is an optional field in OpenSALT.
- *Status End Date:* This date is often only known when a new status is started. This is an optional field in OpenSALT.
- *Note:* Notes or comments generated by the Framework Publisher about the context of the Framework. This is an optional field in OpenSALT.

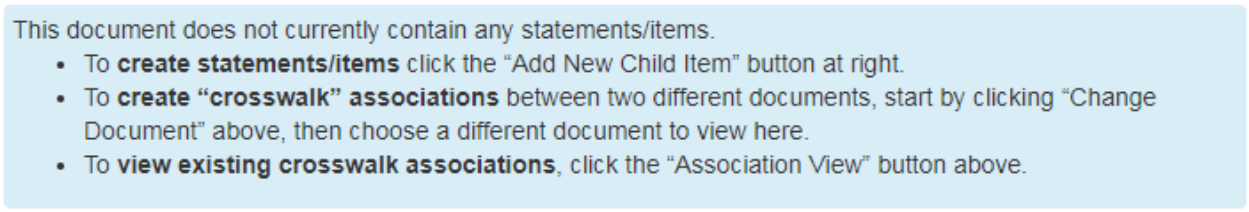
To **create** the Framework, click the **Create** button.

To **cancel** and abandon any changes, click the **Back to the list** button.

When a Framework is created the **Framework Display** page will refresh and will indicate that are loaded with the framework.



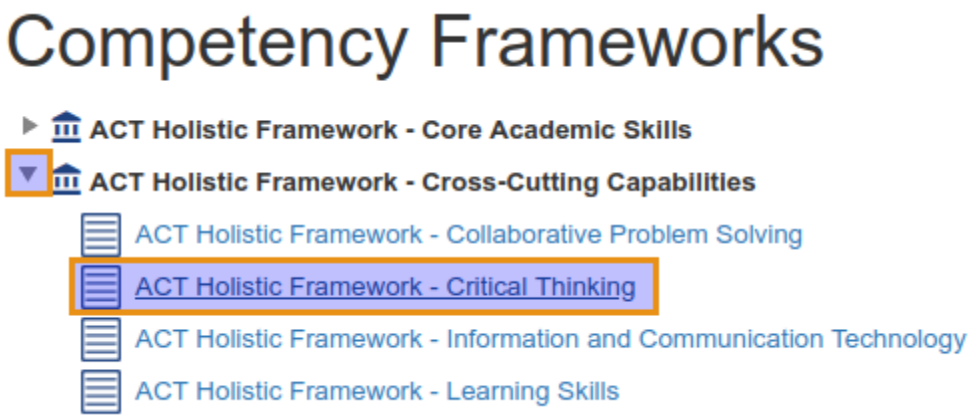
Open SALT Provides the user with suggestions for how to add items to the framework.



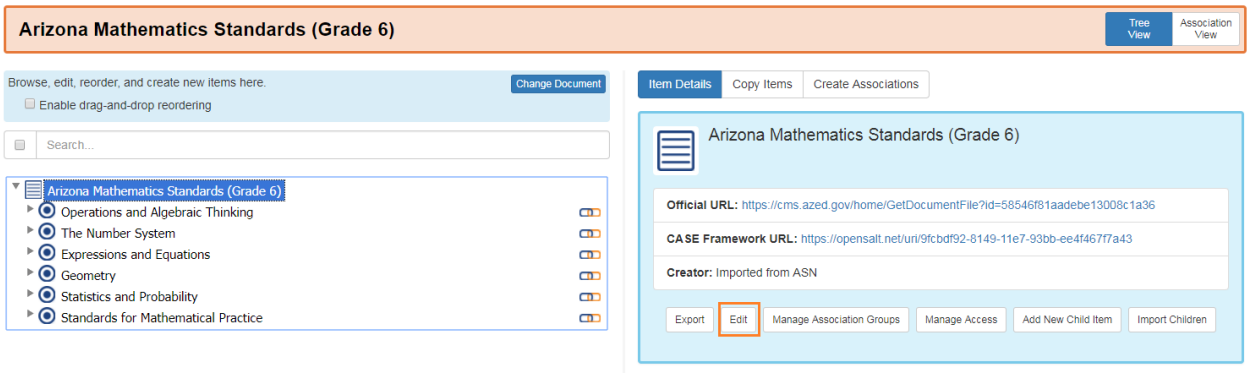
6.2.3 6.2.3 Edit Frameworks

Audience: Super Users, Super Editor, Organization Admins, Organization Editor

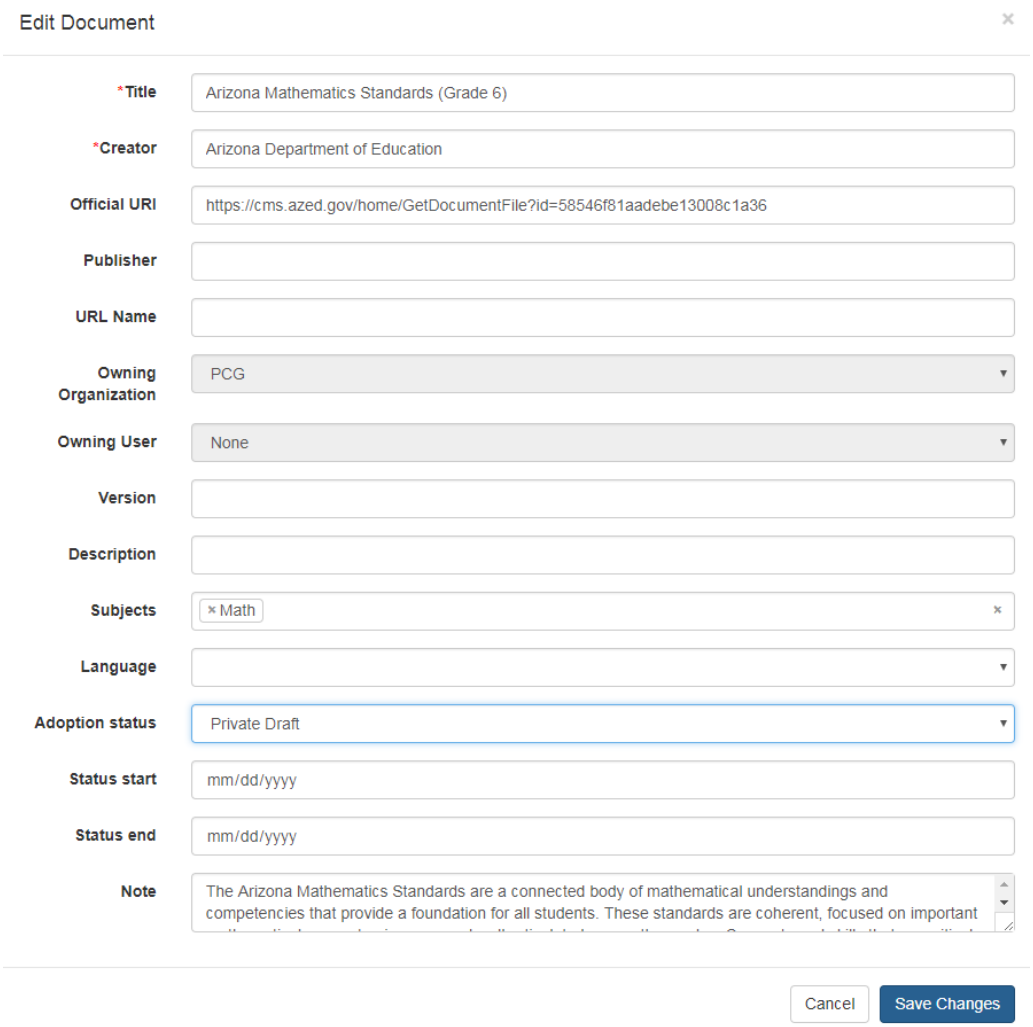
To **Edit** a **Framework** the credentialed user will select the appropriate **Framework** from the **Framework Contents/OpenSalt Home** Page by expanding the **Organization** and clicking on the desired **Framework**.



On the **Framework Display** page, click on the **Edit** button in the **Detail Frame**.



The **Edit Document** window will open and allow the user to update the necessary fields.



The following fields will display on the Edit Document window. Note some may be edited and others have fixed values:

- *Title:* The title as it appears on the cover of the Official Source artifact, although it may be a title created by the Publisher. This is a mandatory field in OpenSALT.
- *Creator:* The the entity that authorized or created the competency framework. It could be an education agency, higher education institution, professional body. It is the owner of the competency framework (e.g CCSSO, TEA, NGSS). This is a mandatory field for OpenSALT and will act as the Organization Folder on the Framework Contents/OpenSALT Home page.
- *Official URI:* The URL of the artifact adopted by the Standard Setting Entity. Often this document is published in html and/or as pdf and is used by the standard setting entity as part of its approval process. Since it is not the intent of this specification to fully reproduce the human-facing content and formatting of the source document, it is recommended that this document be transmitted as part of the competency framework package. This is an optional field in OpenSALT, though best practices indicate it should be filled in.
- *Publisher:* The entity that loads and publishes the Framework. Note that in many cases, the Standard Setting Entity may lack technical capabilities to publish the Competency Framework in a standard format so a third party may be displayed. This is an optional field in OpenSALT, though best practices indicate it should be filled in.
- *URL Name:* This field allows users to enter in a user friendly URL name. example: vs

- *Owning Organization:* If a Framework is not a personal Framework, and rather is an Organizational Framework, the associated Organization that created or imported the Framework will be displayed. *Owning User:* If the Framework is a personal Framework owned by an individual user, the user's name will be displayed. Otherwise the OpenSALT will assume a value of none because the Framework is not owned by a single user rather is owned by an Organization.
- *Version:* This is used to separate any version information expressed by the Official Source artifact. Once and CF Pkg has been approved and published, any changes to an CF Item will constitute a new version of the CF Doc. This is an optional field in OpenSALT. If best practices are not followed, this field may be blank.
- *Description:* The description is typically created by the the Publisher as a standard description of the Competency Framework.This is an optional field in OpenSALT. If best practices are not followed, this field may be blank.
- *Subjects:* This is a string expressing the general subject area of the Competency Framework (e.g. Mathematics). This is an optional field in OpenSALT. If best practices are not followed, this field may be blank.
- *Language:* HTML Language Country Code VIA- country code from . This is an optional field in OpenSALT, though best practice indicates the filled should be filled in. However OpenSALT assumes English if not other language value is entered.
- *Adoption Status:* Adoption status displays the Framework's current status as Draft, Private Draft, Adopted, or Deprecated. OpenSALT assumes Adopted as the default if no status is specifically selected for the framework. This is an optional field in OpenSALT. If best practices are not followed, this field may be blank. OpenSALT assumes Adopted as the default if no status is specifically selected for the framework.
  - *Draft*: Able to be edited by Editors and Admins in an organization. Able to be viewed by the public.
  - *Private Draft*: Able to be viewed and edited by Editors and Admin in the owning organization
  - *Adopted*: Not able to be edited by Editors or Admin
  - *Depreciated:* Was once published, however it is now out of date
- *Status Start Date:* The date that the CF Doc status started. This is an optional field in OpenSALT.
- *Status End Date:* This date is often only known when a new status is started. This is an optional field in OpenSALT.
- *Note:* Notes or comments generated by the Framework Publisher about the context of the Framework. This is an optional field in OpenSALT.

To **save** the updated, click the **Save Changes** button.

To **cancel** and abandon any changes, click the **Cancel** button.

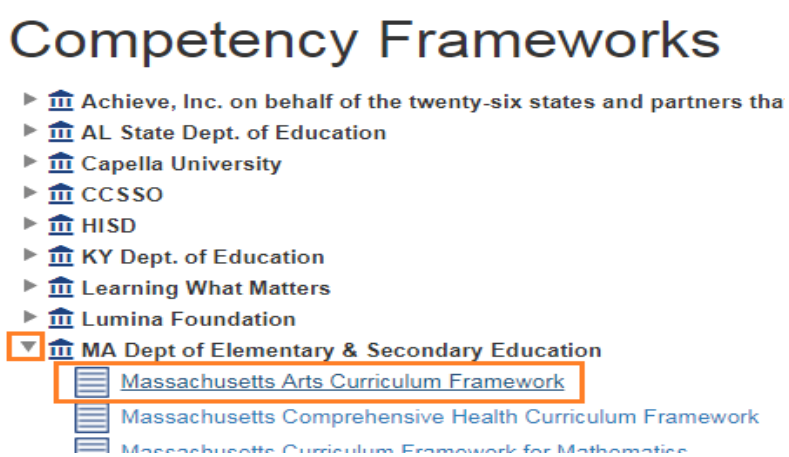
6.2.4 6.2.4 Delete Frameworks

Audience: Super Users, Organization Admins

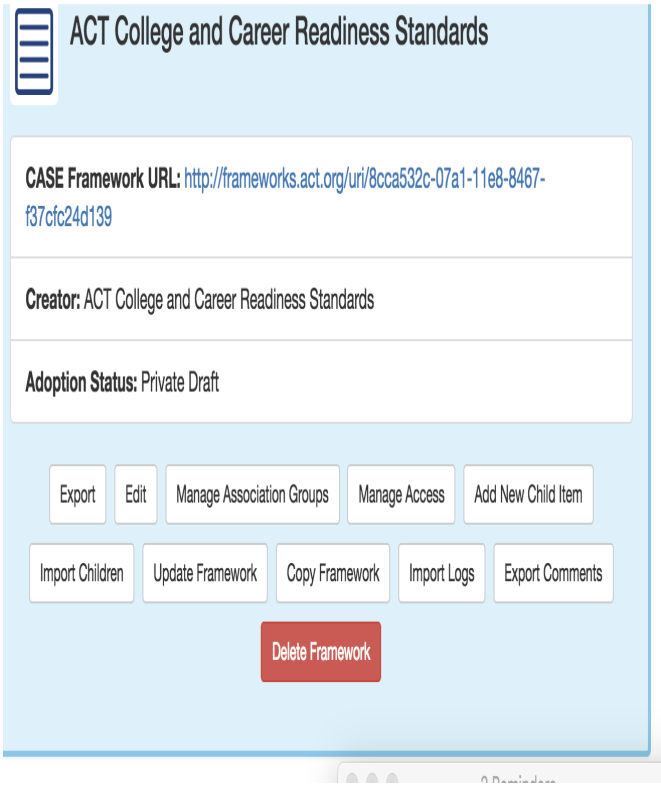
OpenSALT allows Super Users and Organization Admins to delete Frameworks if needed. Note once a Framework is deleted the action can not be undone. The framework and all associations will be permanently removed from the database.

To delete a framework the Super User or Organization Admin will first need to log into OpenSALT.

Next view the selected Framework by expanding the organization and clicking on the selected Framework.



Once you are sure the correct Framework is selected and you are ready to **delete**, click the **Delete** button.



6.2.5 6.2.5 Update Frameworks from a spreadsheet

OpenSALT allows organization admins and above to download a framework as an excel document and make the following changes that can be merged back into the framework on the server: - Update an item's text based on the same guid (will overwrite any other field in that line) - Add an item to the the spreadsheet - do not enter an identifier; one will be created for you on update - Delete an item when removing the identifier

- If a parent statement is deleted, those children statements will be ungrouped as "orphans"

Note that the framework should only be updated from within the Document itself using the Update button. Furthermore, note that currently these changes are irreversible within the application. The Update code searches for the Cf

Item identifier and then performs the update procedures described above based on that business logic. (Eg if it finds a new identifier, it adds the CF Item. If it does not find an identifier previously present, that item is removed from the server).

How-To Video

6.2.6 6.2.6 Create new associations via Spreadsheet Update Features

OpenSALT is able to use this function as well to update/create associations with generally the same user actions. Note that you have to be using the 'latest' spreadsheet download each time you update a framework.

6.2.6.1 Update Associations for frameworks on the same server

Simply enter the guids and association type you want in the appropriate columns on the Cf Association Tab of the downloaded spreadsheet. Do not enter an identiifer for the association action itself; this will be created by OpenSALT on upload.

54ca2fc6-f	83515050-f460-11e8-9	Is Related To	6b9ba9a3-d7cc-11e8-824f-0242ac160002
54cb3308-	83519f06-f460-11e8-a	Is Related To	6b9c3725-d7cc-11e8-824f-0242ac160002
	83510988-f460-11e8-9	Is Related To	6ba138d5-d7cc-11e8-824f-0242ac160002
	83515730-f460-11e8-l	Is Related To	6ba1b8b8-d7cc-11e8-824f-0242ac160002
	83516536-f460-11e8-l	Is Related To	6ba224e2-d7cc-11e8-824f-0242ac160002
	83516c34-f460-11e8-9	Is Related To	6ba26d02-d7cc-11e8-824f-0242ac160002
	83517314-f460-11e8-9	Is Related To	6b9d769c-d7cc-11e8-824f-0242ac160002
	83517a26-f460-11e8-8	Is Related To	6ba13b7c-d7cc-11e8-824f-0242ac160002
	83518110-f460-11e8-l	Is Related To	6ba1bbb1-d7cc-11e8-824f-0242ac160002
	8351880e-f460-11e8-a	Is Related To	6ba227ec-d7cc-11e8-824f-0242ac160002
	83518eee-f460-11e8-b	Is Related To	6ba2703e-d7cc-11e8-824f-0242ac160002
	83519830-f460-11e8-9	Is Related To	6b9e4aa8-d7cc-11e8-824f-0242ac160002
	8351a5e6-f460-11e8-9	Is Related To	6b9dead9-d7cc-11e8-824f-0242ac160002
	8345f106-f460-11e8-8	Has Skill Level	6b9c8faf-d7cc-11e8-824f-0242ac160002

6.2.6.2 Update Associations from an external server

This process is the same as for frameworks on the same server with two steps prior to the download, change, and update.

Steps: 1) Load an external framework within the framework you are working in. 2) Make one association from that document to the framework you are working on now. 3) Now use the guids from the external framework in the spreadsheet update tool as detailed above and the associations will load with connections to that external framework.

6.2.7 Copy Framework Only

When viewing a framework in the tree view, with sufficient rights on the CF Document item detail pane you will see a button to Copy Framework. This will open up an additional dialog box for multiple options to copy the entire framework documents

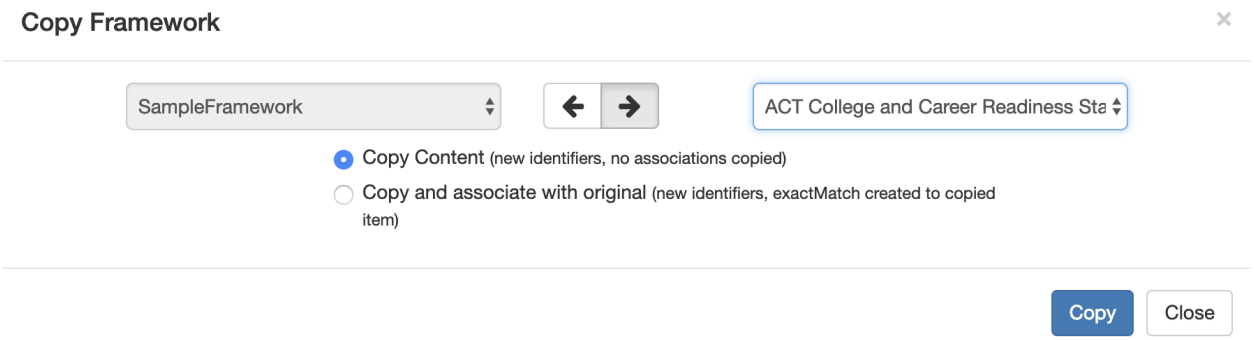
height 100 px

width 150 px

6.2.7.1 Copy Content Only

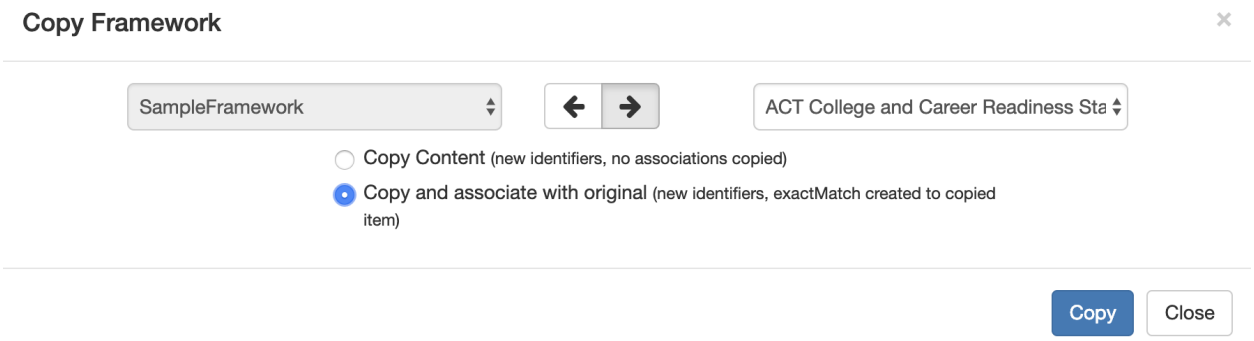
If you are intending on creating a new framework from the existing one, or simply with to add the current items to an existing framework, use this option. It will copy the framework items and structure and create new GUIDS for the

items in the new framework. Associations are not transferred or created.



6.2.7.2 Copy and Make Derivative Framework

This will copy the framework items into a the target document and create Exact Match Of associations between origin and target items. No other associations will be copied. **There is a bug in OpenSALT 2.2 where a hard refresh is necessary before the copied CFItems are viewable to the user undertaking the copy action.**



6.3 6.3 Framework Items

Items can be created for Frameworks either by importing existing data using the OpenSALT CSV template or by manually creating the items.

6.3.1 6.3.1 Import Items with OpenSALT Template

Users can create their own CASE-compliant data files to load into OpenSALT using a provided CSV template and guide. This allows users to easily import their items or standards into their created framework and eliminates the need for manually entering the data into the system. The template can be found here: .

6.3.1.1 CSV Loading Guide for CASE

- Note that since OpenSALT 2.0 it is preferred to use the Spreadsheet Importer instead of this CSV Import.

General notes when loading/creating CSV files for bulk upload of child statements:

The only required field to upload a framework is FullStatement, but to build a hierarchy you will need Human Coding Scheme as well as the two layers for IsChildOf.

If an identifier exists, it will be transferred to the OpenSALT system. If it does not exist one will be created upon upload and creation.

SequenceNumber may be used to denote a particular order within parent nodes.

Currently the only fields with Rich Text capability through markdown are FullStatement and Notes.

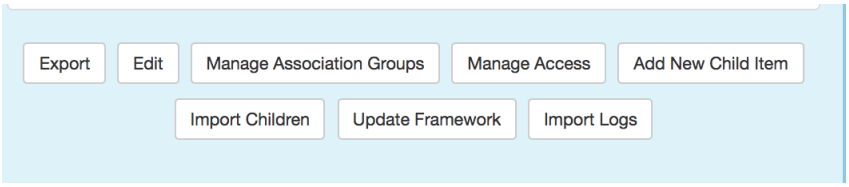
See "CASE.csv" in the /opensalt/sample files/.

Tab 1: Step 1 Read This

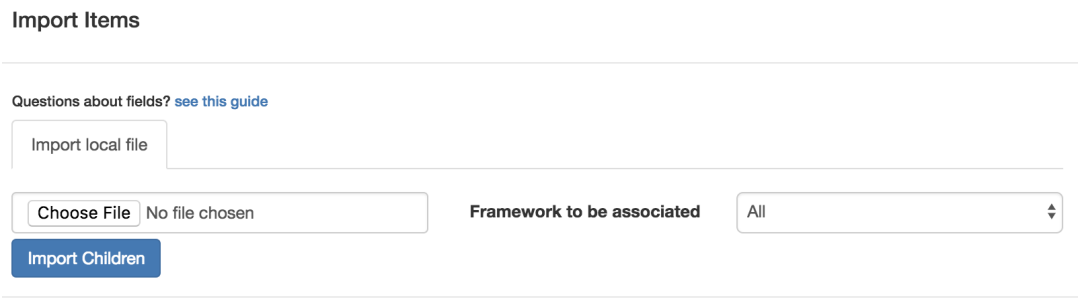
An Overview of the process for creating the CSV to import the data.

Step 1: Create New Framework

Step 2: Select "Import Children"



Step 3: Select your CSV



Step 4: If your CSV has associations to external frameworks in the full human readable terms (ie CCSS.MATH.Content.K.CC.A.1) select the framework you want to associate it too. If nothing just leave alone

Step 5: Select Import Children

Tab 2: CF DOC

The only required fields are Creator and title. Creator is the folder that will be created.

Tab 3: CF ITEM

Explanation of the item fields that will be included in template for the Framework. Note that technically only fullStatement and humanCodingScheme are required.

The only required field is fullstatement.

To create a hierarchy however, use human coding scheme, ischildOf and SequenceNumber

example:

fullStatement	humanCodingScheme	IsChildOf	SequenceNumber
Parent Item	P		1
Child 1	P.C	P	1
Grandchild 1	P.C.G	P.C	1
Child 2	P.C2	P	2
Grandchild 2	P.C2.G	P.C2	1

Tab 4: CF Association

The process for associations which can be associated in the template if desired, though associations may be easier managed in the UI itself after the Framework is created and imported.

Association Types

For most people, if you are correlating these standards with another framwork, it is probably easier to do that in the tool than in the spreadsheet originally. But if you do have associations to Common Core for example, enter the full statement id here, such as CCSS.MATH.CONTENT.K.MD.A.1 under a column for "IsRelatedTo" or "ExactMatchOf" depending on if your standard is indeed the same statement just with a different numbering scheme or not. Also, your standards should have some way of identifying how they should be displayed under "IsChildOf"

M=Mandatory, O= Optional

CF Association	Property	M/O/R	posed IMS Descrip
CF Association	CFDocumentURI	M	An unambiguous reference to the standards document which contains this CF Item using a network-resolvable URI. [CEDS Element: Learning Standard Document Identifier URI, ID:000693]
CF Association	identifier	M	An unambiguous, synthetic, unique reference to the association.
CF Association	uri	M	A URI that establishes uniqueness of an association between a learning standard and another learning standard or other objects such as learning resources. [CEDS Element: Learning Standard Item Association Identifier URI, ID:000871]
CF Association	originNodeIdentifier	M	Identifier of the origin node when the Learning Standard Item Association is used as a connector in a learning map. [CEDS Element: Learning Standard Item Association Origin Node URI, ID: 001406]
CF Association	originNodeURI	M	URI of the origin node when the Learning Standard Item Association is used as a connector in a learning map. [CEDS Element: Learning Standard Item Association Origin Node URI, ID: 001406]
CF Association	destinationNodeIdentifier	M	Identifier of the destination node when the CF Association is used as a connector in a learning map. [CEDS Element: Learning Standard Item Association Destination Node URI, ID: 001404]
CF Association	destinationNodeURI	M	URI of the destination node when the CF Association is used as a connector in a learning map. [CEDS Element: Learning Standard Item Association Destination Node URI, ID: 001404]
CF Association	lastChangeDateTime	M	A system generated log of the most recent change to this record
CF Association	associationType	M	A controlled vocabulary used to express the types of associations used to describe the relationship between CF Docs and between CF Items.
CF Association	CFAssociationGroupingIdentifier	O	An identifier to allow associations to be grouped together. Different values only have to be unique within the document.
CF Association	CFAssociationGroupingIdentifierURI	O	An attribute of CF Item Type to establish a controlled vocabulary.
Association Type	isPeerOf	-	The source/origin is a peer of of the target/destination.
Association Type	isChildOf	-	To represent the structural relationship in a taxonomy between parent and child.
Association Type	isPartOf	-	The origin of the association is included either physically or logically in the item at the destination of the association. This classifies an item as being logically or semantically contained as a subset of the destination.
Association Type	exactMatchOf	-	Equivalent to. Used to connect derived CF Item to CF Item in original source CF Doc (e.g. Common Core State Standards)
Association Type	precedes	-	The origin of the association comes before the destination of the association in time or order.
Association Type	isRelatedTo	-	The origin of the association is related to the destination in some way that is not better described by another association type.
Association Type	replacedBy	-	The origin of the association has been supplanted by, displaced by, or superseded by the destination of the association.
Association Type	exemplar	-	The destination of the association is a learning resource that exemplifies this standard. In this case, the destination need not be another item, but can be any URL that points to the resource in question.
Association Type	hasSkillLevel	-	The destination of this association is understood to define a given skill level i.e. Reading Lexile 100, Depth Knowledge 2, or Cognitive Level (Blooms Taxonomy) etc.

Tab 6: Template

The template itself that will be used to add the data into the correct format and saved as a CSV to import into OpenSALT.

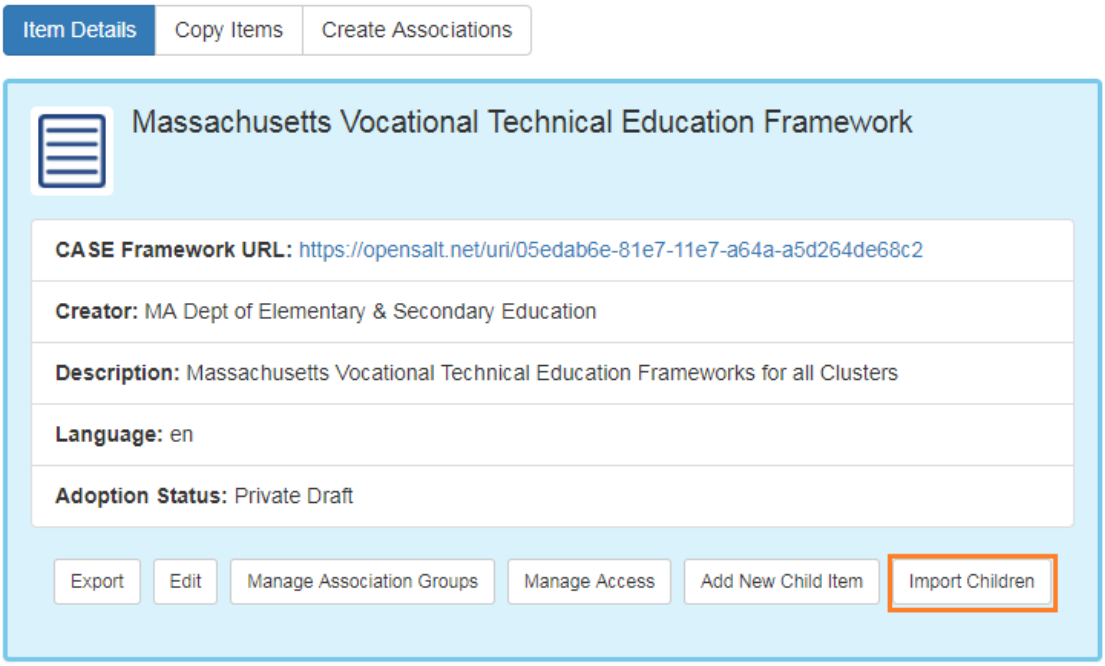
You may use these fields as your template and save into a CSV. +——————+——————+——————+————+——————-+ |Identifier |fullStatement | humanCodingscheme | IsChildOf | SequenceNumber | +——————+——————-+————+——————-+

6.3.1.2 Import Process

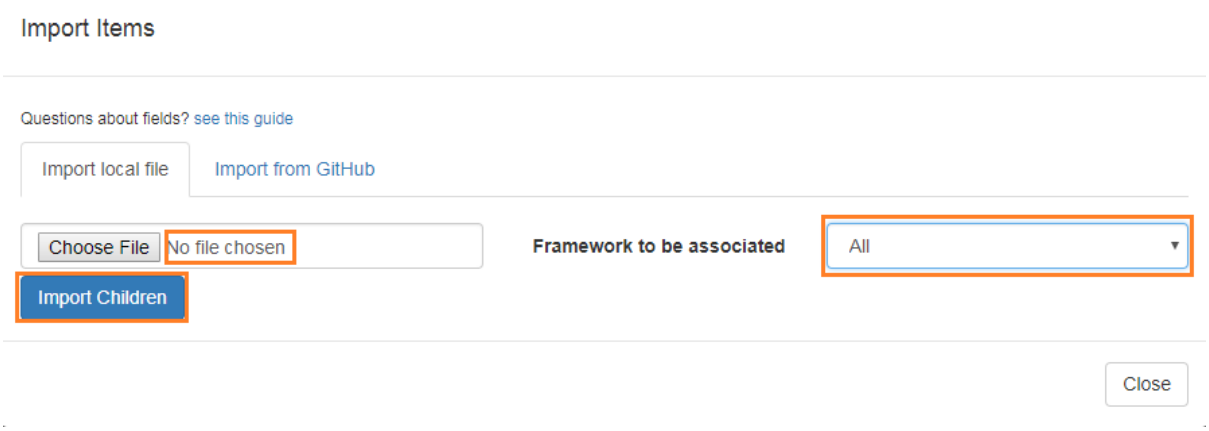
After reviewing the **CSV Loading Guide for CASE** and the user will need to format their data into the Template Tab then save the file locally to their computer as a CSV file. This will save only the Template Tab and convert the data from an Excel file to a CSV file for import.

identifier	fullStatement	HumanCodingScheme	Legacy Human Coding Scheme	SMART Level	Abbreviated Statement	ConceptKeywords	Notes	Language	Educa
	Strand 1: Safety and Health Knowledge and Skills	1		1.1.1.1				English	9-12
	Fundamentals of Health and Safety	1.A		1.1.1.1.1				English	9-12
	Describe and apply health and safety regulations.	1.A.1		1.1.1.1.1.1				English	9-12
	Identify, describe and apply health and safety regulations that apply to specific tasks and jobs. Stud	1.A.1.1		1.1.1.1.1.2				English	9-12
	Identify, describe and apply Environmental Protection Agency (EPA) and other environmental protect	1.A.1.2		1.1.1.1.1.3				English	9-12
	Identify, describe and apply Right-To-Know (Hazard Communication Policy) and other communicativ	1.A.1.3		1.1.1.1.1.4				English	9-12
	Explain procedures for documenting and reporting hazards to appropriate authorities.	1.A.1.4		1.1.1.1.1.5				English	9-12
	Identify and describe potential consequences for non-compliance with appropriate health and safet	1.A.1.5		1.1.1.1.1.6				English	9-12
	Identify and list contact information for appropriate health and safety agencies and resources.	1.A.1.6		1.1.1.1.1.7				English	9-12
	Demonstrate appropriate health and safety practices based on the specific occupational area.	1.A.2		1.1.1.1.2				English	9-12
	Identify, describe and demonstrate the effective use of Safety Data Sheets (SDS).	1.A.2.1		1.1.1.1.2.1				English	9-12
	Read and interpret chemical, product and equipment labels to determine appropriate health and saf	1.A.2.2		1.1.1.1.2.2				English	9-12
	Identify, describe and demonstrate personal, shop and job site safety practices and procedures.	1.A.2.3		1.1.1.1.2.3				English	9-12
	Demonstrate safe dress and use of relevant safety gear, personal protective equipment (PPE) and erg	1.A.2.4		1.1.1.1.2.4				English	9-12
	Demonstrate appropriate safe body mechanics, including appropriate lifting techniques and ergonoi	1.A.2.5		1.1.1.1.2.5				English	9-12
	Locate emergency equipment, first aid kit, SDS information directories and emergency action/respons	1.A.2.6		1.1.1.1.2.6				English	9-12
	Demonstrate the safe use, storage, and maintenance of every piece of equipment in the lab, shop and	1.A.2.7		1.1.1.1.2.7				English	9-12
	Describe safety practices and procedures to be followed when working with and around electricity, e	1.A.2.8		1.1.1.1.2.8				English	9-12
	Handle, store, dispose of and recycle hazardous, flammable and combustible materials, according t	1.A.2.9		1.1.1.1.2.9				English	9-12
	Demonstrate appropriate workspace cleaning, sanitation, disinfection and sterilization procedures (	1.A.2.10		1.1.1.1.2.10				English	9-12

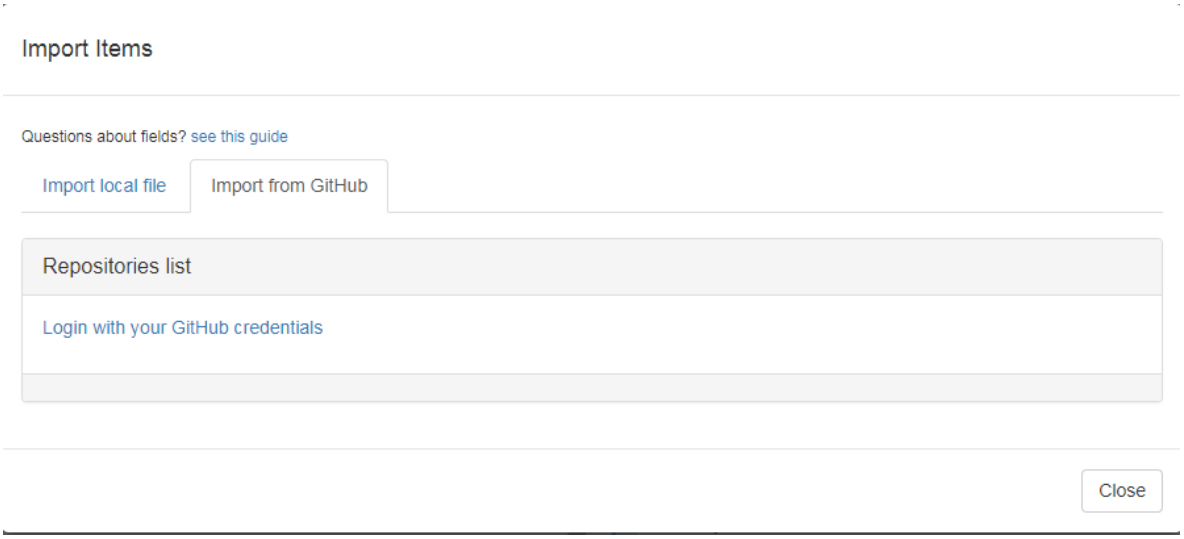
Next the user will need to Import the items/children for the Frameworks. Click on the **Import Children** button in the **Item Details Frame** on the right of the selected framework to open the **Import Items** window.



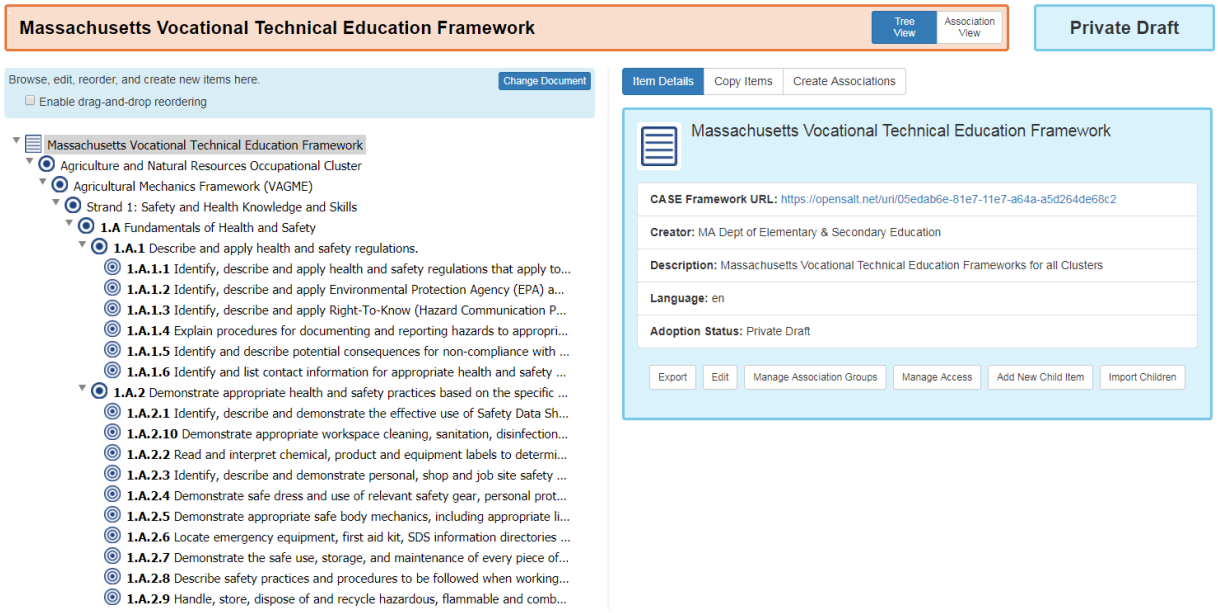
Select the tab for where your template file is located. If the CSV is stored locally on your computer, click on the Import local File tab. Browse and select your file with the Choose File button. If a specific Framework to be associated is preferred, mark that selection otherwise the default is All. Then click the Import Children button to import the items for the framework.



Alternatively if the template is stored in your GitHub repository, click on the Import from GitHUb tab and login to connect to your file and import the items for the framework.

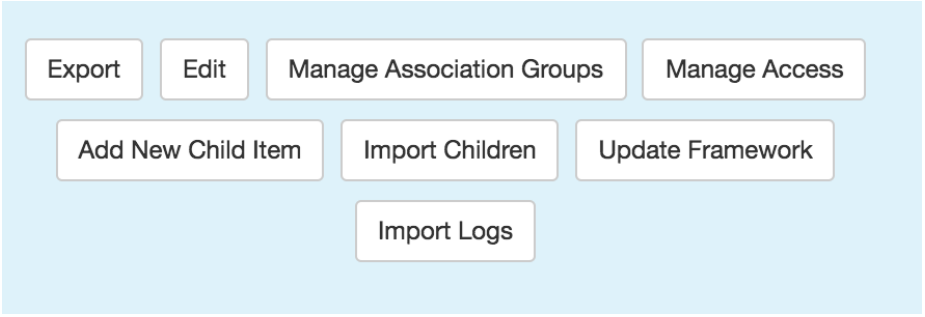


The items will be loaded and the Framework Display page will be refreshed.

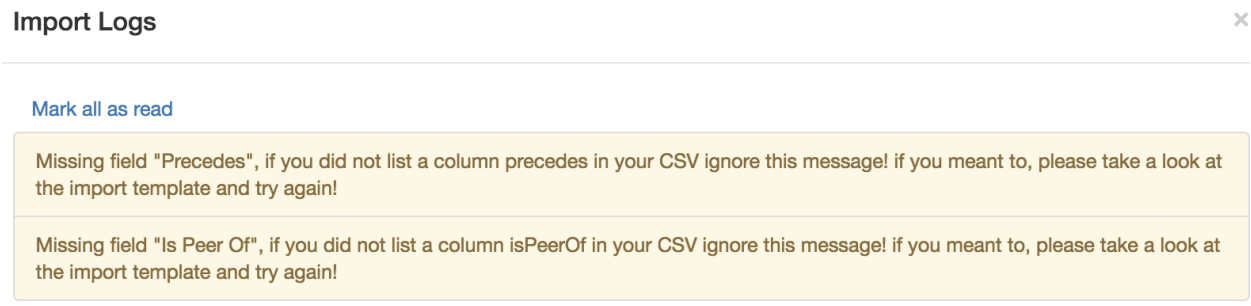


6.3.1.3 Error Log

On import, error messages will display when a file is missing fields. These error messages can be retrieved afterwards by clicking on "error log" in the admin console.



(Sample log)



6.3.2 6.3.2 Formatting for Full Statement and Notes

Item full statements can be formatted using three different methods / pure UTF8 text can be enhanced with:

- Markdown
- Katex
- Limited HTML tags

These three formatting options for text can be combined in-line with limitations. Also note that HTML and Katex will not render consistently in the Tree and top of the Item Detail pane.

6.3.2.1 Using HTML Tags

As of OpenSALT build 1.3. Some HTML tags are allowed despite HTML sanitization. The tags that will render are:

ul, ol, li, b, i, u, br, p

6.3.2.2 Using Markdown

Note that since OpenSALT uses markdown() as the primary formatting language, there could be unintended formatting consequences when importing from external sources.

6.3.2.3 Using Katex

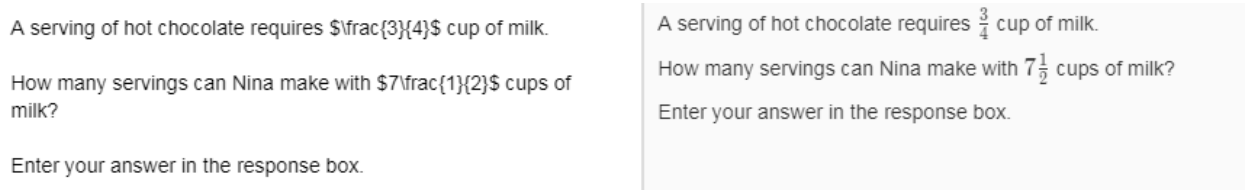
The Katex system supports plain text writing of all KaTeX functions listed . In-line Katex is formatted as braced between dollar signs ($):

Alternatively, block text Katex is formatted as braced between double dollar signs ($$).

(no screenshot available)

6.3.2.4 Combining text and Katex in-line

Katex formatting may be combined in-line with plain text in the markdown editor:

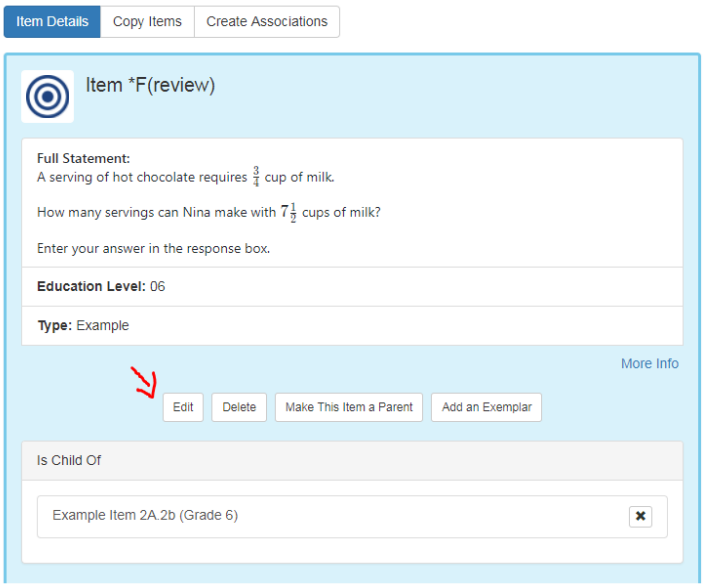


6.3.2.5 Constraints for using HTML tags

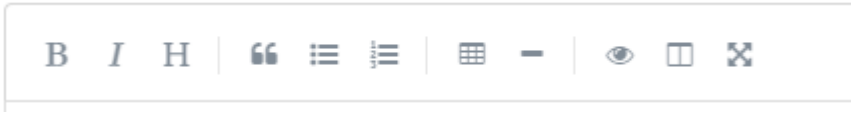
HTML and Katex cannot be combined in-line with anything else.

6.3.2.6 Using the Modes of Markdown Editor UI

The markdown editor is accessed by clicking on the "Edit" button for an item:



The markdown editor has 11 functional buttons to manipulate Full Statement text, from left to right they are: Bold, Italics, Heading, Quote, Generic List, Numbered List, Insert Table, Insert Horizontal Line, Toggle Preview, Toggle Side by Side, Toggle Full Screen. These text-editing options are depicted below:



In the subsections below example text is showin in two columns; the first column shows the Markdown special characters, while the right column shows the text in the way it presents to the end user.

Bold 6.3.2.6.1

Text may be **bolded** by clicking icon 1/11 in the Full Statement toolbar:

In the Markdown editor, Bold special character text is indicated by ** on either end of the statement:

How much water, in cups, will the full container hold?	How much water, in cups, will the full container hold?

Italics 6.3.2.6.2

Text may be *italicized* by clicking icon 2/11 in the Full Statement toolbar:

In the Markdown editor, Italics special character text is indicated by * on either end of the statement:

How much water, in cups, will the full container hold?	How much water, in cups, will the full container hold?

Heading 6.3.2.6.3

Text may be converted into a header by clicking icon 3/11 icon in the Full Statement toolbar:

In the Markdown editor, Heading special character text is indicated by #, which precedes the statement:

```
# This is a test
```

This is a test

Quote 6.3.2.6.4

Text may be converted into a header by clicking icon 4/11 icon in the Full Statement toolbar:

In the Markdown editor, Quote special character text is indicated by >, which precedes the statement:

```
> How much water, in cups, will the full container hold?
> This is a test
> Enter your answer in the response box.
```

> How much water, in cups, will the full container hold?
> This is a test
> Enter your answer in the response box.

Generic List 6.3.2.6.5

Text may be converted into a bulleted list by clicking icon 5/11 icon in the Full Statement toolbar:

In the Markdown editor, Generic List special character text is indicated by >, which precedes the statement:

```
* How much water, in cups, will the full container hold?
* This is a test
* Enter your answer in the response box.
```

- How much water, in cups, will the full container hold?
- This is a test
- Enter your answer in the response box.

Numbered List 6.3.2.6.6

Text may be converted into an enumerated list by clicking icon 6/11 icon in the Full Statement toolbar:

In the Markdown editor, Generic List special character text is indicated by *, which precedes the statement:

```
1. How much water, in cups, will the full container hold?
1. what!?
1. Enter your answer in the response box.
```

1. How much water, in cups, will the full container hold?
2. what!?
3. Enter your answer in the response box.

Insert Table 6.3.2.6.7

A table may be inserted by clicking icon 7/11 icon in the Full Statement toolbar:

In the Markdown editor, Columns headings and associated text are separated by inserting a Horizontal Line:

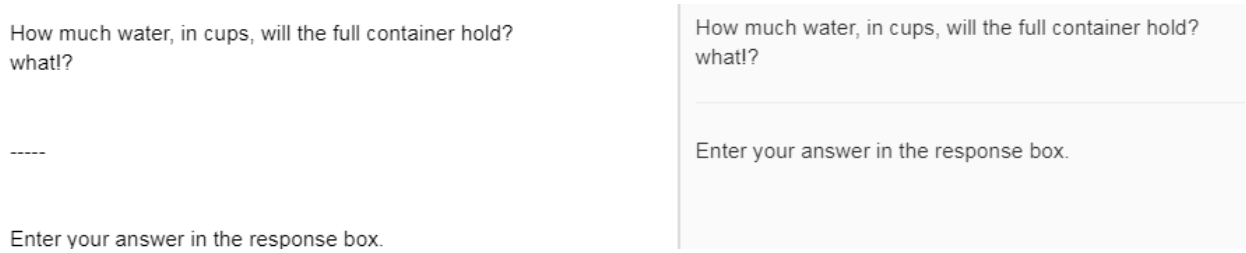
```
| Column 1 | Column 2 | Column 3 |
| -------- | -------- | -------- |
| Text     | Text     | Text     |
```

Column 1	Column 2	Column 3
Text	Text	Text

Insert Horizontal Line 6.3.2.6.8

A horizontal line may be inserted by clicking icon 8/11 icon in the Full Statement toolbar:

In the Markdown editor, a solid horizontal line is indicated by —– between text statements:



Toggle Preview 6.3.2.6.9

An editor toggle between Markdown special characters vs. end user view of text by clicking icon 9/11 icon in the Full Statement toolbar:

The toggle shifts between preview mode and pure text edit mode.

Toggle SIde by Side 6.3.2.6.10

An editor may view Markdown pure text and its rendered view simultaneously by clicking icon 10/11 icon in the Full Statement toolbar:

This two column view is depicted in all examples above and is the easiest way to edit and format text.

Toggle Full Screen 6.3.2.6.11

An editor may expand the Full Statement textarea by clicking icon 11/11 icon in the Full Statement toolbar:

This does not cause the area to expand to the full monitor width, bur rather to the width of the pop-up dialogue.

Underline 6.3.2.6.12

Underline may be utilized by entering Markdown's emphasis tag, or placing _underscore_ on either end of a word or phrase.

6.3.2.7 Other Considerations

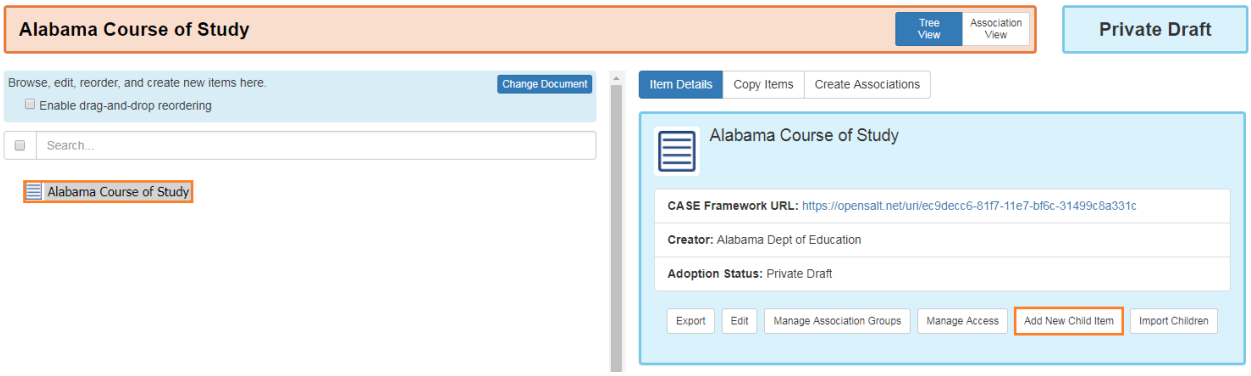
Line Break after markdown table requires the following input: $~$ or an HTML

6.3.3 6.3.3 Create Items Manually

Audience: Super Users, Super Editor, Organization Admins, Organization Editor

When a Framework is created it is item neutral and is open for organizational content to be added based on the user's needs. In addition to the Item import in section , OpenSALT allows users to manually create items and organize the data through the OpenSALT UI.

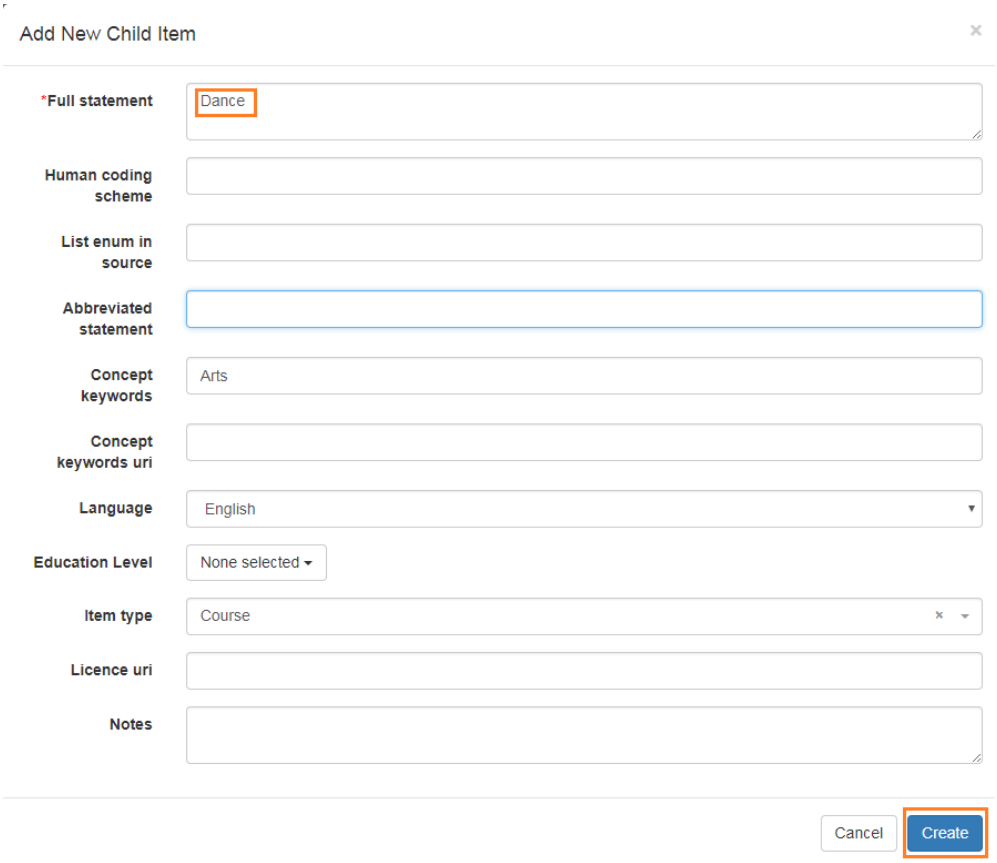
On the **Framework Display** page, click on the Item or Framework name that will act as a parent for the item to be added. Then in the **Item Details Frame** click on the **Add New Child Item** button.



The **Add New Child Item** window will display.

Enter in the appropriate values for all available fields. At a minimum enter in mandatory **Full Statement.**

- *FullStatement*: The the main content of the CF Item. It is used to express both nodes and granular statements. If the statement is part of a list, the list enumeration should not be included in the statement and should instead be contained in the List Enumeration in Source Document. This is a mandatory field in OpenSALT.
- *HumanCodingScheme*: The ID sometimes used by humans to identify a CF Item. It often will use concatenated codes expressing its position in the taxonomy and abbreviations to convey other classification information (e.g. K.CC.1.1). This is an optional field in OpenSALT.
- *ListEnumeration*: Used to parse out enumerations or bullets that precede CF Item statements. This is an optional field in OpenSALT..
- *AbbreviatedStatement:* Abbreviated or summary statement provided by the Publisher. This is an optional field in OpenSALT and may be blank.
- *ConceptKeywords:* Upper level CF Item node statements may be used to populate Concept Keywords of lower level nodes. upper The concepts data structure allows a master list of keywords to be defined which can then be parsed down specific to a node. This works as usually concepts will be a less granular hierarchy representation of of the more detailed nodes in CF items. A node could be 'Geometry' and the lower node is 'Tangents' but the keywords for 'tangents' could include the word geometry. This is an optional field in OpenSALT and may be blank.
- *Language:* HTML Language Country Code VIA- country code from . This is an optional field in OpenSALT. If best practices are not followed, this field may be blank. However OpenSALT assumes English if not other language value is entered.
- *EducationLevel:* The current US K12 defined vocabulary is to use CEDS https://ceds.ed.gov/cedselementdetaiCF.aspx?termid=8267. Multiple values are allowed via comma delimitation and should be used to express grade spans. This is an optional field in OpenSALT and may be blank.
- *ItemType:* e.g., "Standard," "Benchmark," "Strand," or "Topic." or "Level 1, Level 2,…" This is an optional field in OpenSALT and may be blank.
- *License uri:* Systems may filter for content with particular licences to support discovery. This is an optional field in OpenSALT and may be blank.
- *Notes:* In some cases, this can be used to contain additional information found in the original source document. This is an optional field in OpenSALT and may be blank.

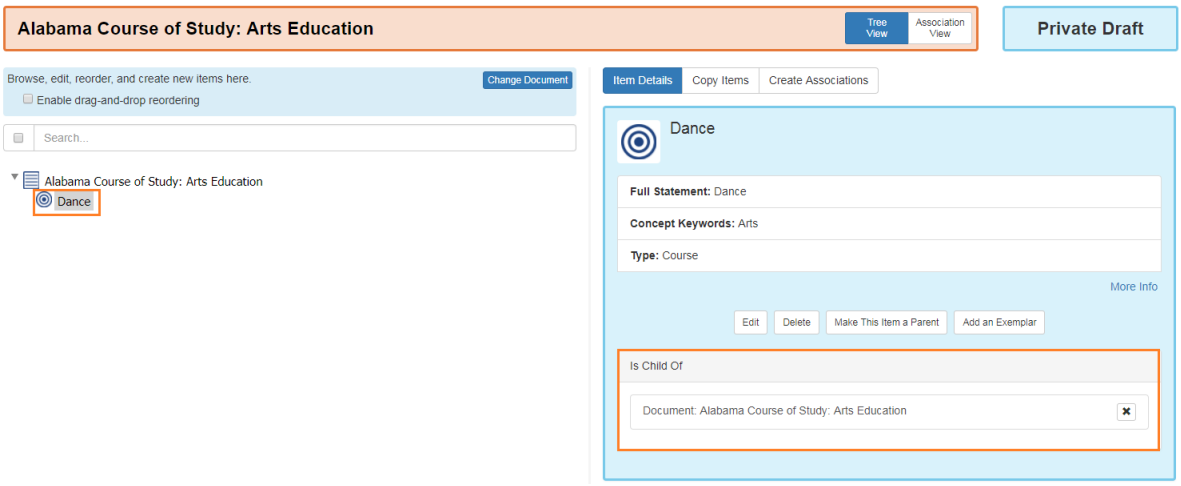


To **create** the item, click the **Create** button.

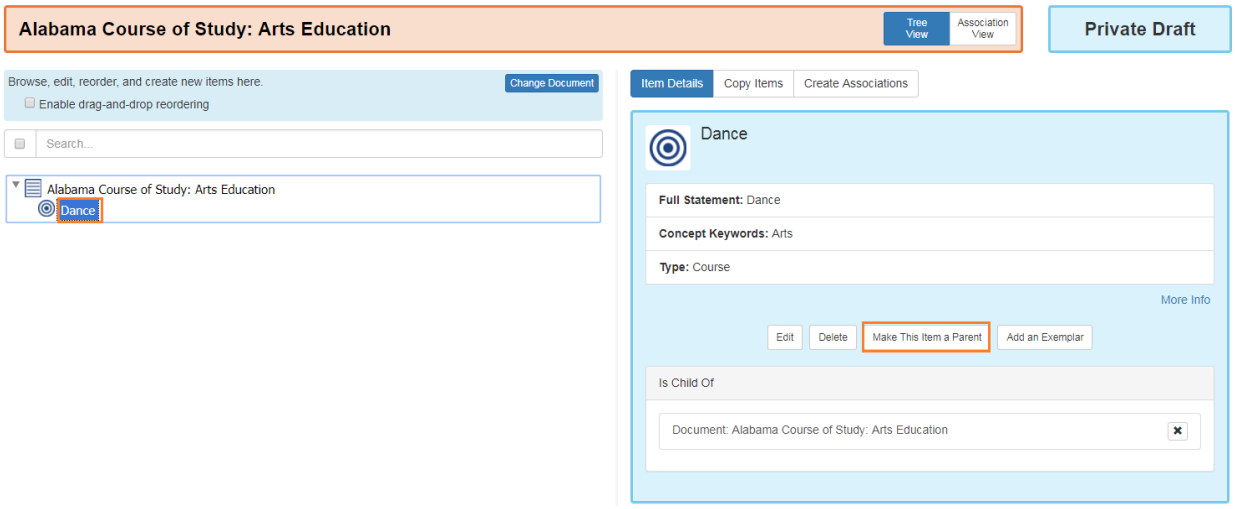
To **cancel** and discard the changes, click the **Cancel** button.

6.3.4 6.3.4 Parent vs Child Items

Once the Item is created it will by default only be a child of the item it was created for.



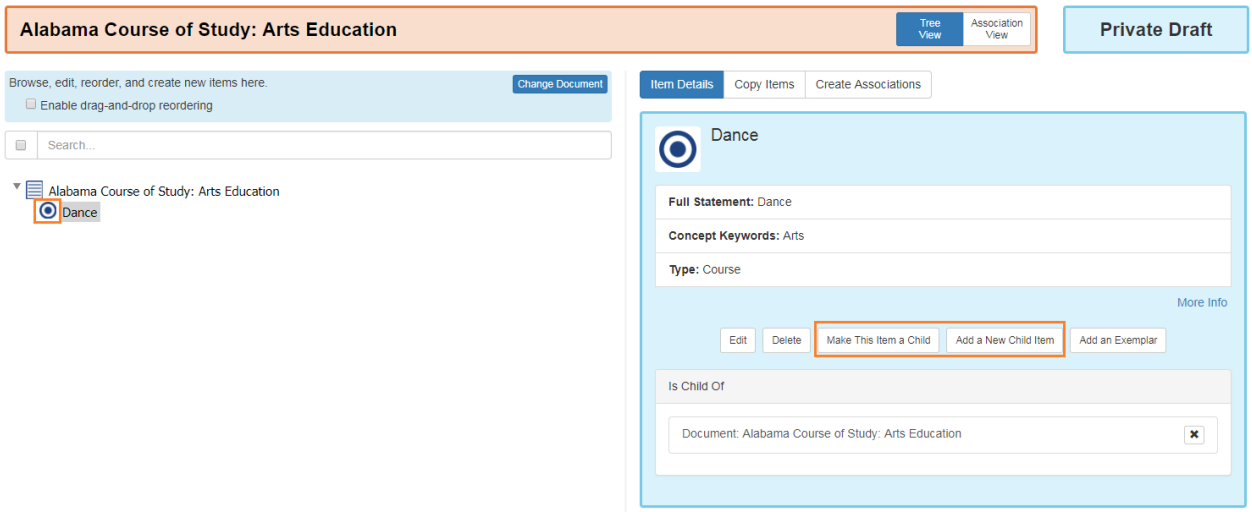
To add items nested below the item created, the user will need to make this item a Parent item. Highlight the item on in the **Framework Display** and click on the **Make This Item a Parent** button in the **Item Detail Frame**.



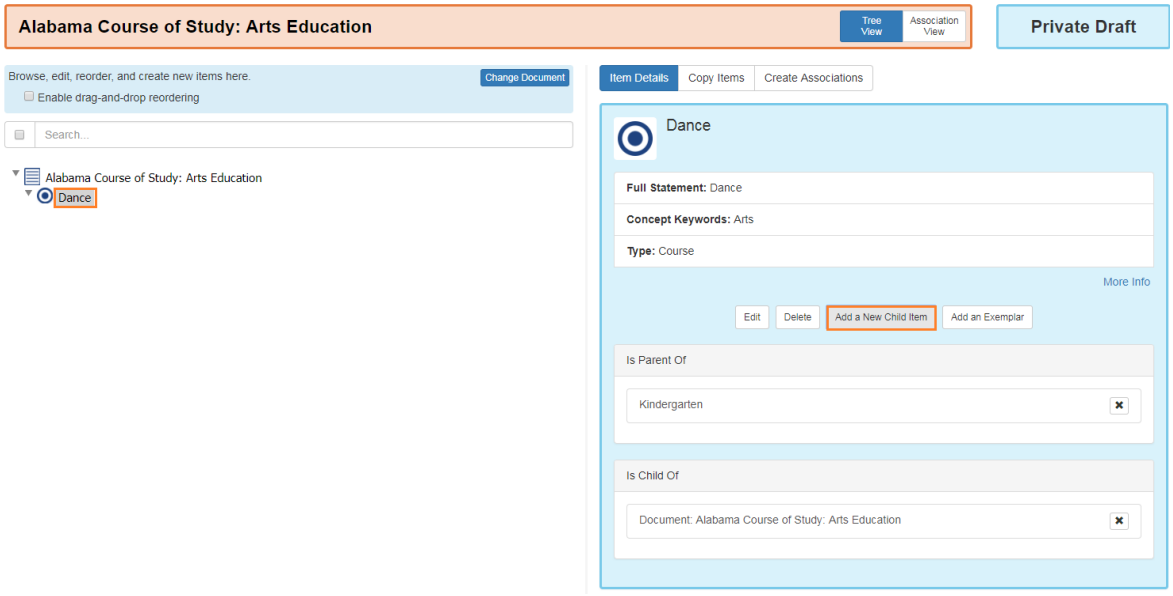
The bullet icon on the left of the item in the **Framework Display** will update to reflect it is now a P**arent item** and the buttons in the **Item Detail Frame** will update to allow the user to either downgrade back to a child with the **Make This Item a Child** button or to add a new child for this parent item with the **Add a New Child** *button.*

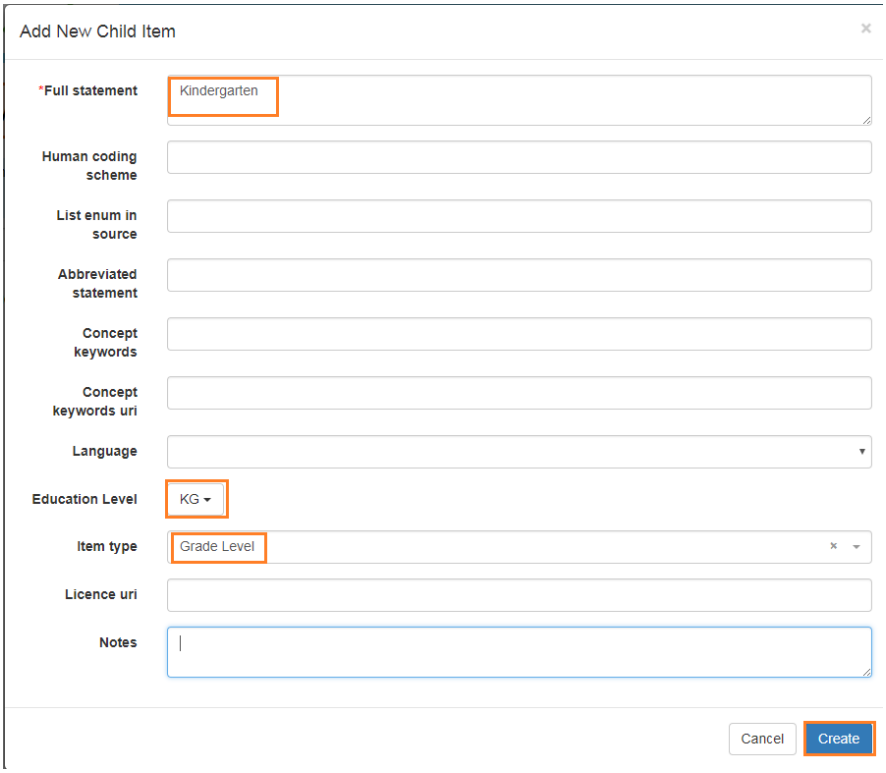
The user can toggle the item between **Parent or Child** as necessary by clicking the **Make This Item a Parent** or **Make this Item a Child** buttons respectively.

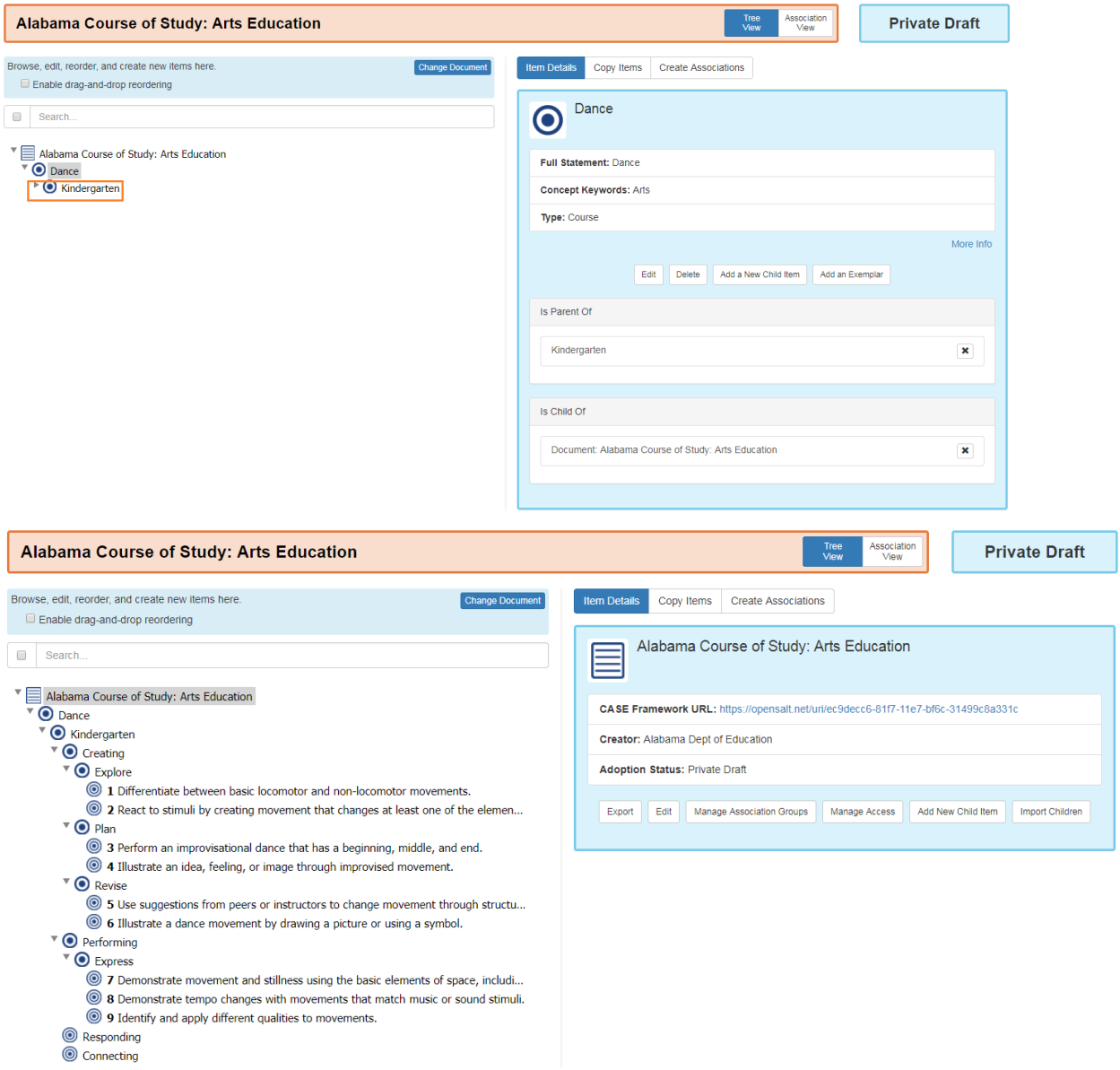
Note an item can be a Child of an item and a Parent to other items, however an item can not be a Child of an item with children below it. This item must be upgraded to a Parent to have children.



The user can repeat this process an unlimited number of times to continue adding all items and marking them as Parents when appropriate to complete the full content for the framework.


Alabama Course of Study: Arts Education
Tree View
Association View
Private Draft
Browse, edit, reorder, and create new items here.
Change Document
Enable drag-and-drop reordering
Search...
Alabama Course of Study: Arts Education
Dance
Item Details
Copy Items
Create Associations
Dance
Full Statement: Dance
Concept Keywords: Arts
Type: Course
More Info
Edit
Delete
Add a New Child Item
Add an Exemplar
Is Parent Of
Kindergarten
Is Child Of
Document: Alabama Course of Study: Arts Education


Add New Child Item
*Full statement
Kindergarten
Human coding scheme
List enum in source
Abbreviated statement
Concept keywords
Concept keywords uri
Language
Education Level
KG
Item type
Grade Level
Licence uri
Notes
Cancel
Create

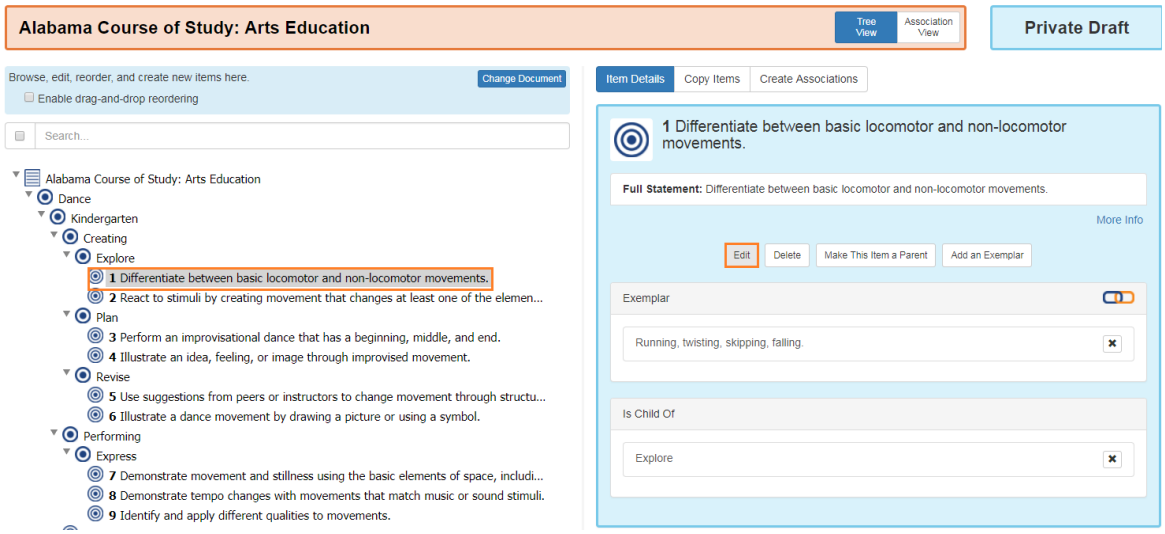


6.3.5 6.3.5 Edit Items

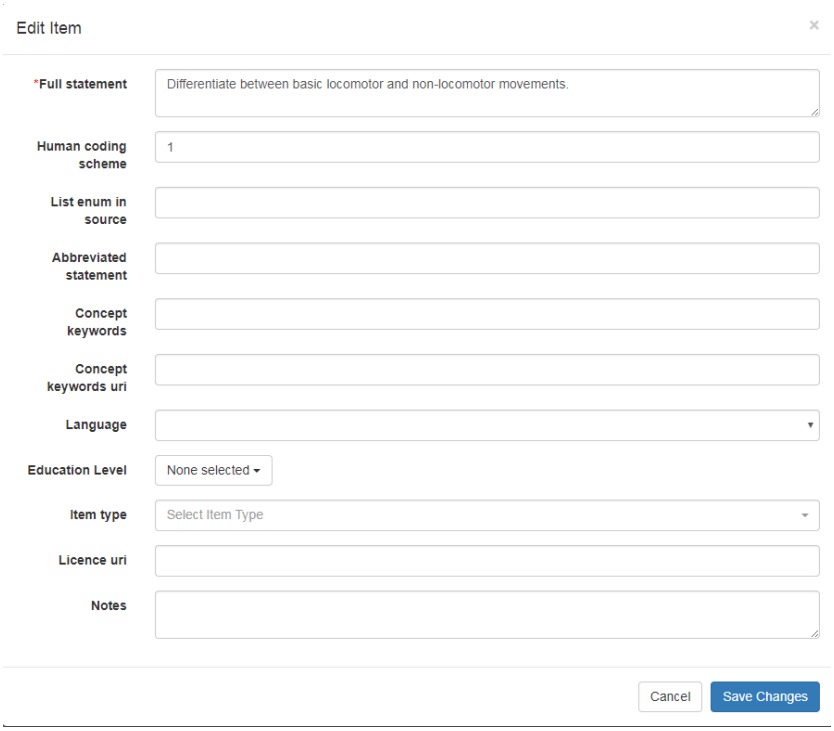
Audience: Super Users, Super Editor, Organization Admins, Organization Editor

6.3.5.1 Edit Item Content

To **edit** an item, the user must click on the item in the **Framework Display** and then click on the **Edit** button in the **Item Detail Frame.**



The **Edit Item** window will display. And the user can alter the data fields as needed.



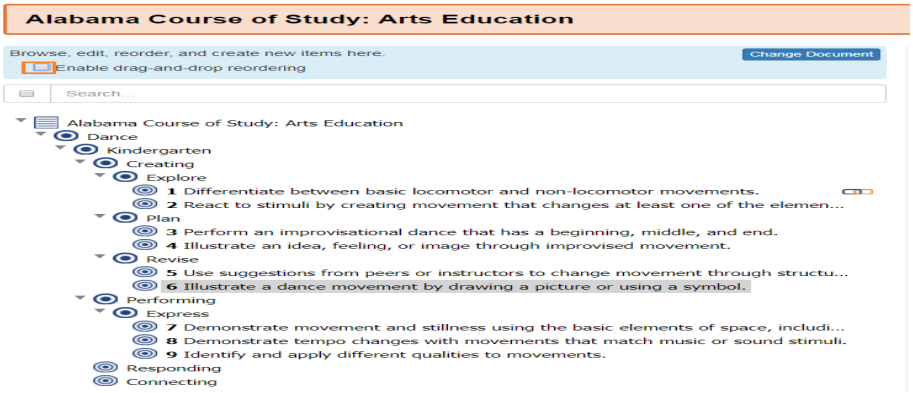
To **save** the changes, click the **Save Changes** button.

To **cancel** and discard the changes , click the **Cancel** button.

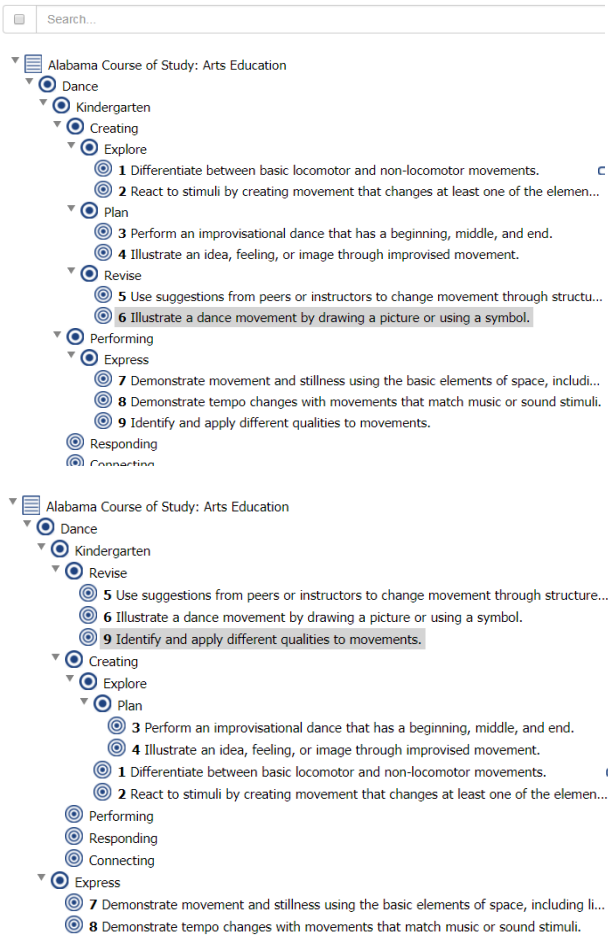
6.3.5.2 Edit Item Positions

Audience: Super Users, Super Editor, Organization Admins, Organization Editor

OpenSALT enables users to move items within frameworks and reorder without the need to edit each item. The user must first enable the functionality but checkin on the box at the top of the Framework Display to Enable drag-drop reordering.



With the box checked and the function enabled, the user can now click on an item and drag and drop it to another location. Note clicking on a child will only move the child, whereas selecting a Parent will move the parent and all of it's children.



Note with the drag and drop feature, items can be indented (moved to new parents) outdented to become parents, and all data can be fully moved around. However once an item is moved, there is no undo button, so the user would need to manually move (drag/drop) the item to the original position if the move is not required or done in error. Because of this, the feature is always defaulted to unchecked so no errors are accidentally made.

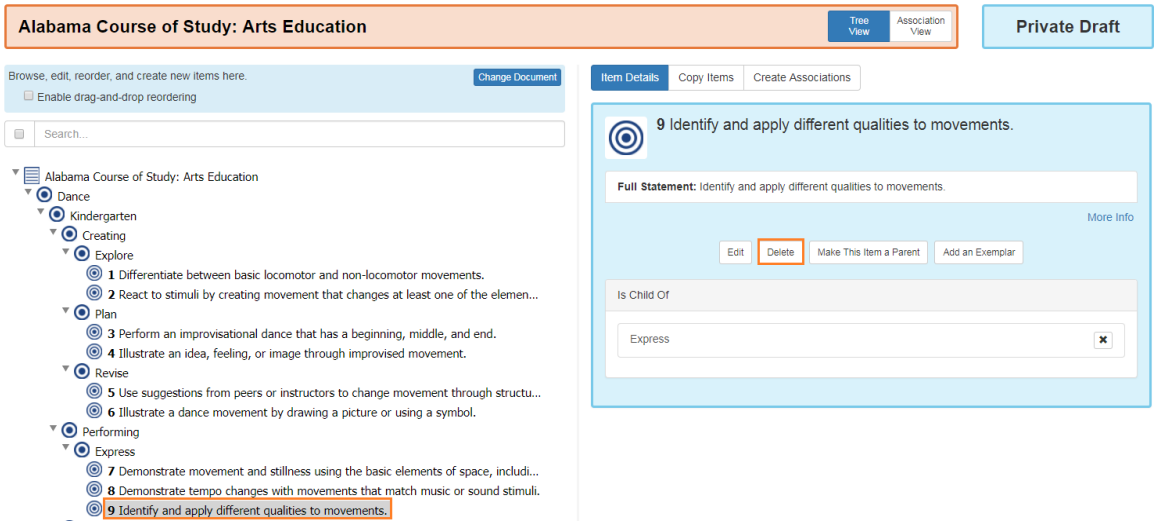
6.3.6 6.3.6 Delete Items

Audience: Super Users, Super Editor, Organization Admins, Organization Editor

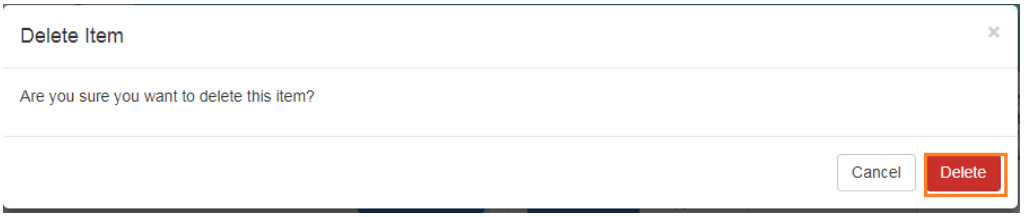
All credentialed users can delete items that they have access to edit.

6.3.6.1 Delete Child Item

To **delete** a child item, the user must click on the item in the **Framework Display** and then click on the **Delete** button in the **Item Detail Frame.**



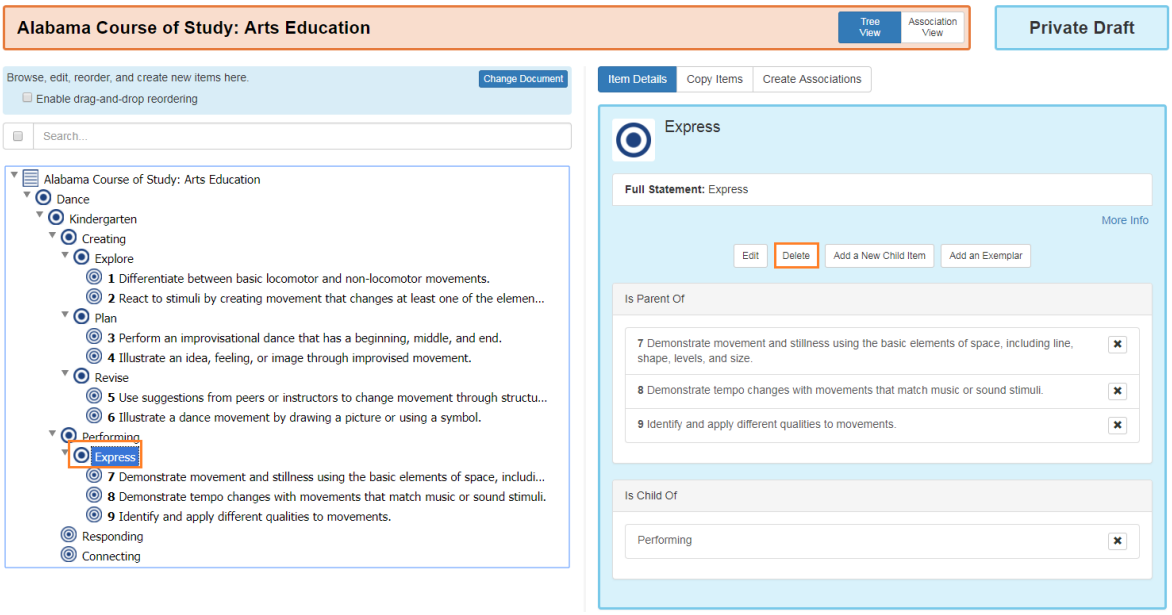
The delete action can not be undone. The user will be prompted with a warning message and must acknowledge the action cannot be reversed by clicking on the **Delete** button to proceed.



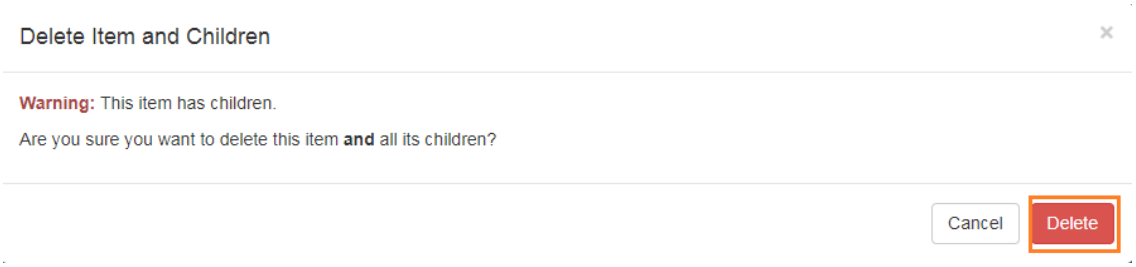
To cancel and leave the item in the Framework, click the **Cancel** button.

6.3.6.2 Delete Parent Item

To **delete** a parent item, the user must click on the item in the **Framework Display** and then click on the **Delete** button in the **Item Detail Frame.**



The delete action can not be undone. The user will be prompted with a warning message that the item and all of it's children will be deleted if the user proceeds. The user and must acknowledge the action cannot be reversed by clicking on the **Delete** button to proceed.



To cancel and leave the item in the Framework, click the **Cancel** button.

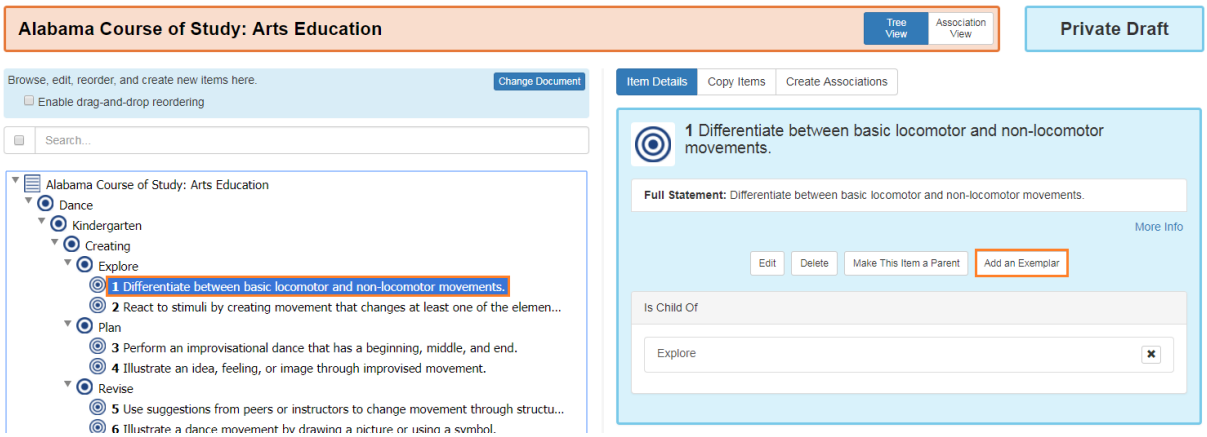
6.4 6.4 Exemplars for Items

Audience: Super Users, Super Editor, Organization Admins, Organization Editor

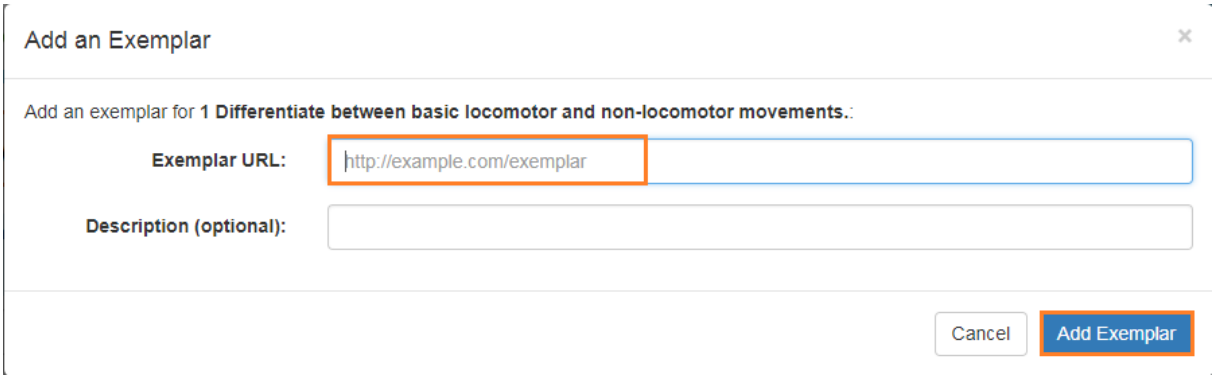
OpenSALT provides the ability for a user to connect an example or **Exemplar** to any item within a Framework. Note the Exemplars can not be associated to the Framework, rather only the items within. However an Exemplar can be associated to a Parent or a Child Item.

6.4.1 6.4.1 Add Exemplar

To connect an example or **Exemplar** to an item. Select the item within the framework on the **Framework Display** page. Then click on the **Add Exemplar** button on in the **Item Details Frame**.

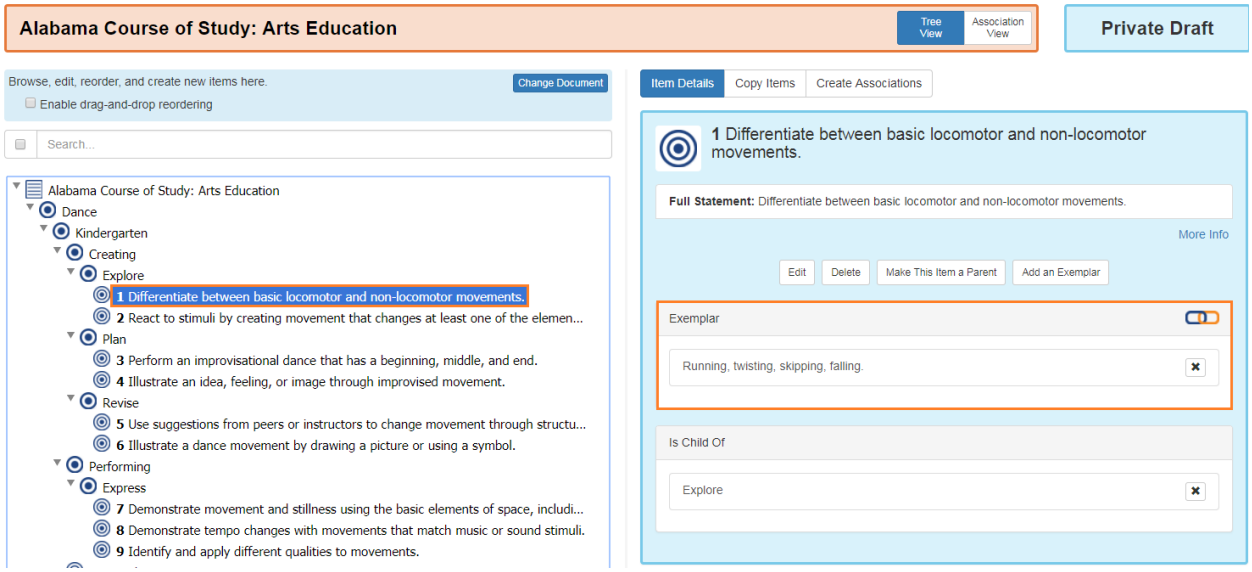


The **Add an Exemplar** window will open.



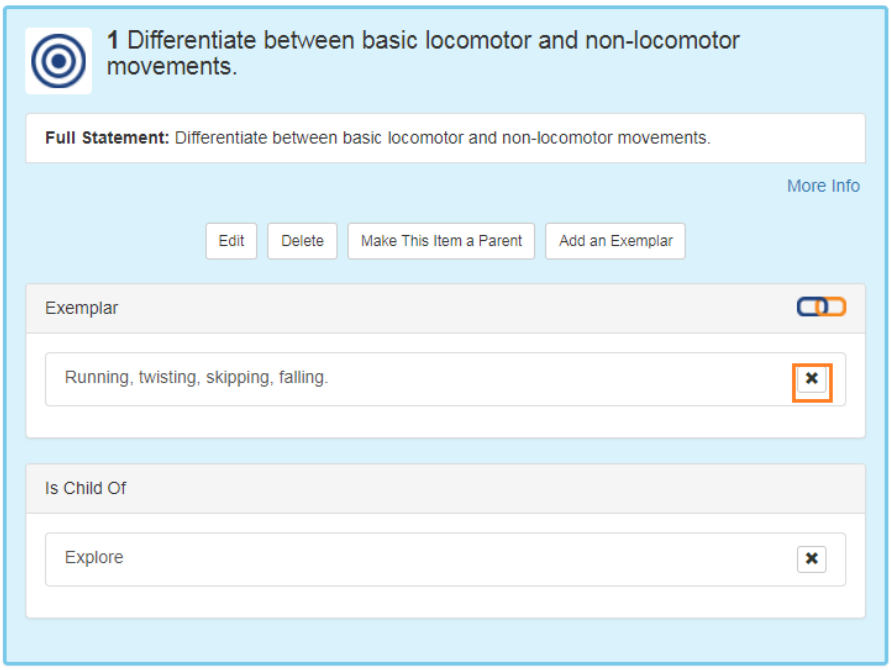
The user can either enter in a URL to an Exemplar or type text into the box. To save the Exemplar, click on the **Add Exemplar** button. To discard changes click on the **Cancel** button.

After clicking the **Add Exemplar** button, the F**ramework Display** page will refresh and the exemplar is linked to the item. This can be verified in the **Item Detail Frame**.

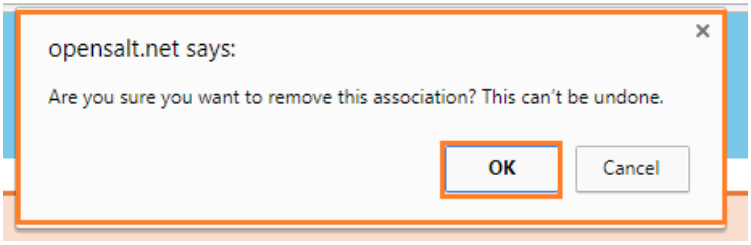


6.4.2 6.4.2 Delete Exemplar

To delete an **Exemplar** from an item, click on the '**x**' icon next to the exemplar detail in the **Exemplar section** of the I**tem Detail Frame** on the **Framework Display** page. Note this action can not be reversed once completed. The user will be promoted with a warning message to proceed.



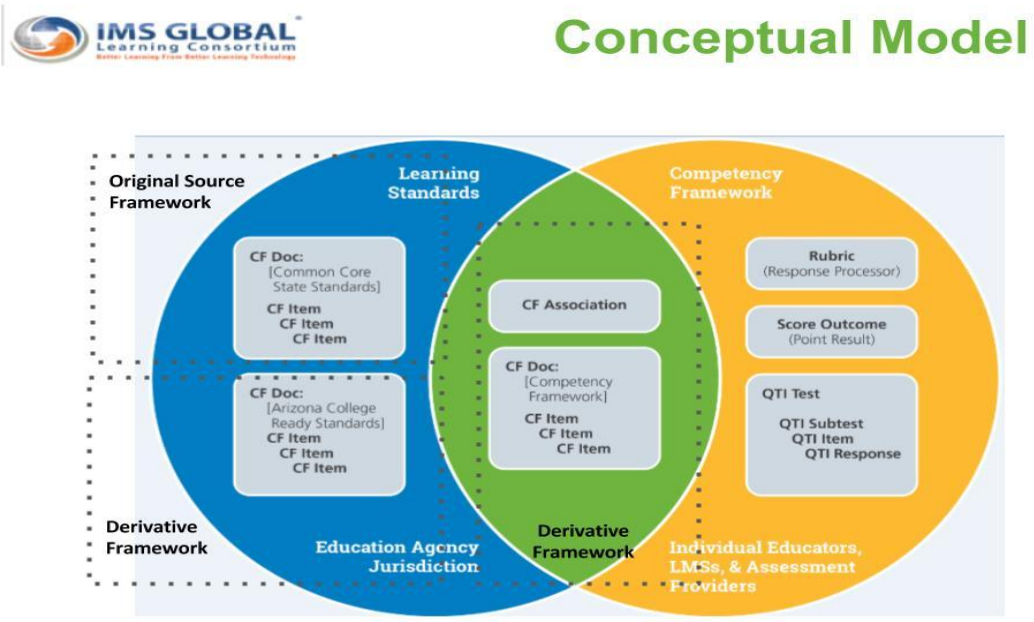
Note this action can not be reversed once completed. The user will be promoted with a warning message and must acknowledge the **OK** button to proceed.



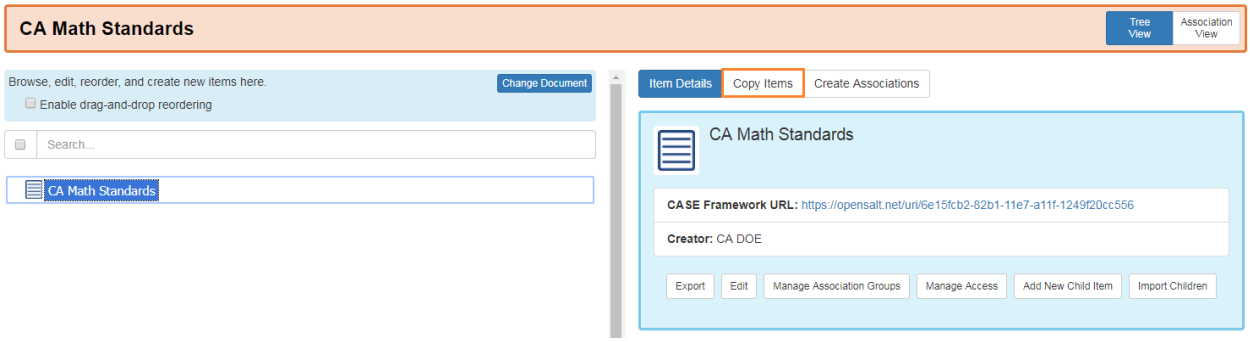
6.5 6.5 Derivative Frameworks - Copy Items

Audience: Super Users, Super Editor, Organization Admins, Organization Editor

OpenSALT allows users to create derivative frameworks from frameworks within an OpenSALT server or instance. Creating a derivative framework allows organizations to take a base original copy of a framework and expand it to better meet its needs. The illustration below shows how the derivative frameworks fit with the scope of frameworks as a whole. Derivative frameworks essentially act as a bridge for organizations to navigate education needs.

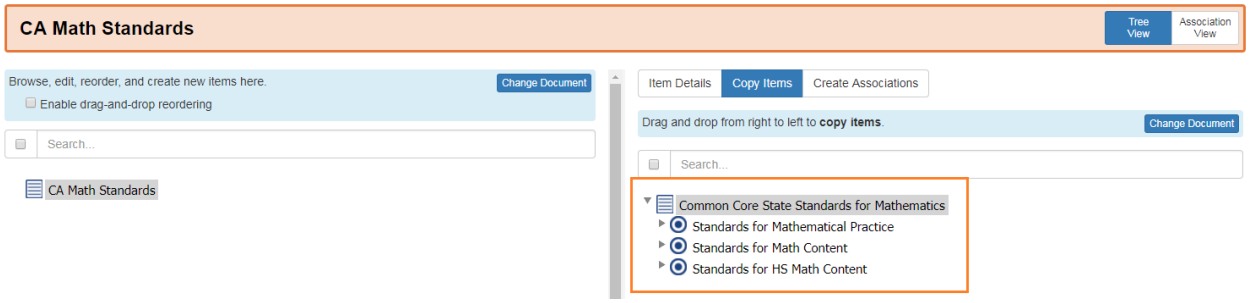


The user will need to be logged into OpenSALT and create or select the framework that will act as the new derivative framework or copy. Please see section if assistance is needed on how to create a framework. On the **Framework Display** page, click on the **Copy Items** button in the **Item Detail** frame.



In the C**opy Items** frame select the document/Framework to copy. Note if the framework to copy is not listed in the document view it is not currently on the OpenSALT server. Please see section for instructions on importing frameworks into OpenSALT.

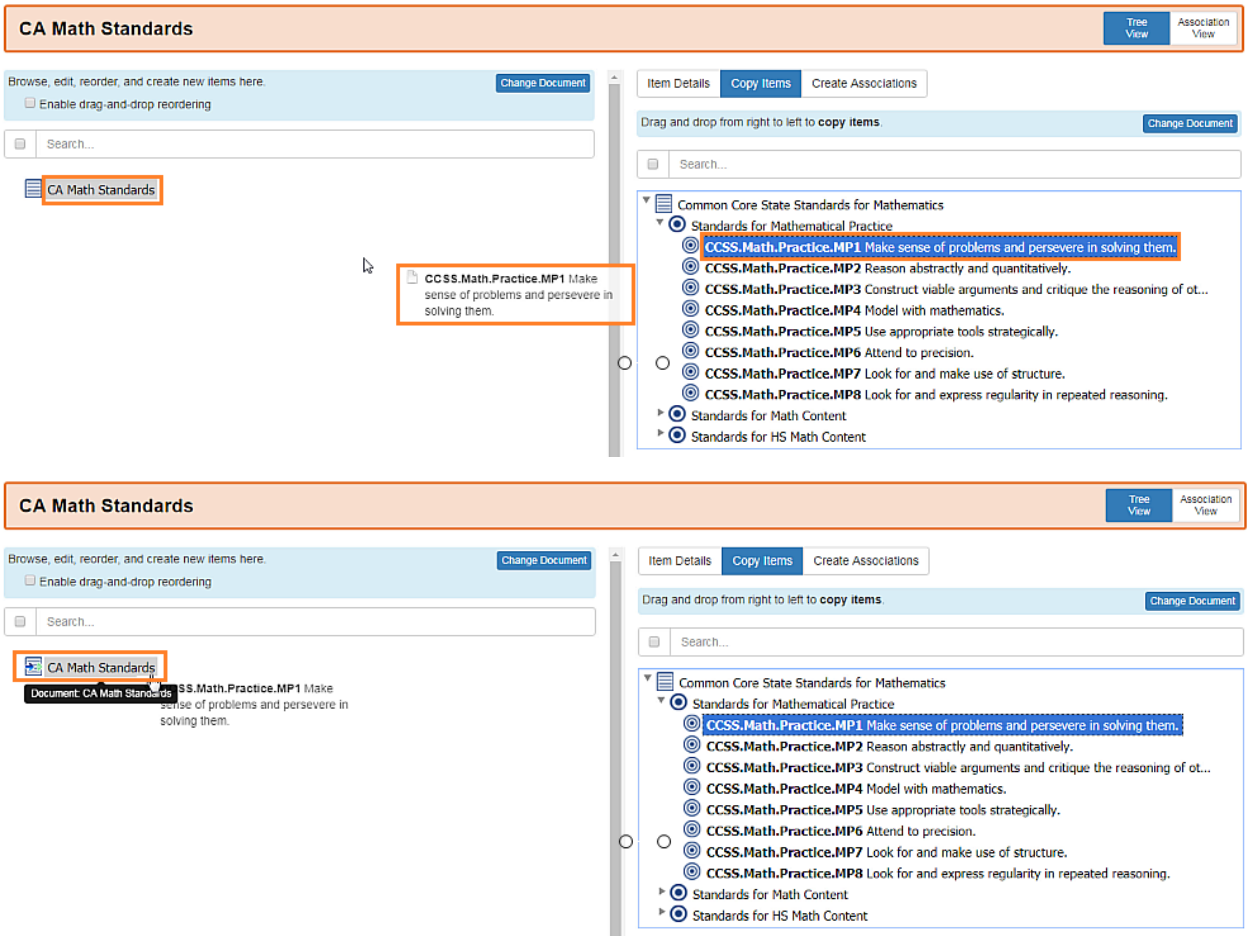
Select the desired framework in the **Document** dropdown. The Copy ITems frame will refresh and load the selected framework so its items can be viewed.



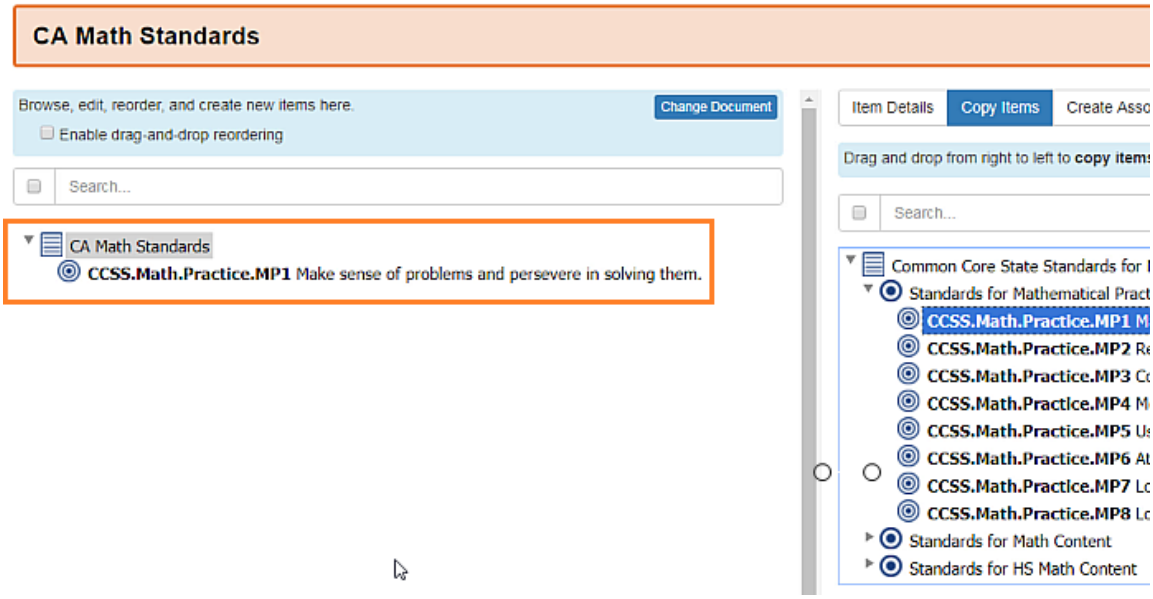
The user can now select individual items or the full set to copy over to the derivative framework on **Framework**

Display on the left.

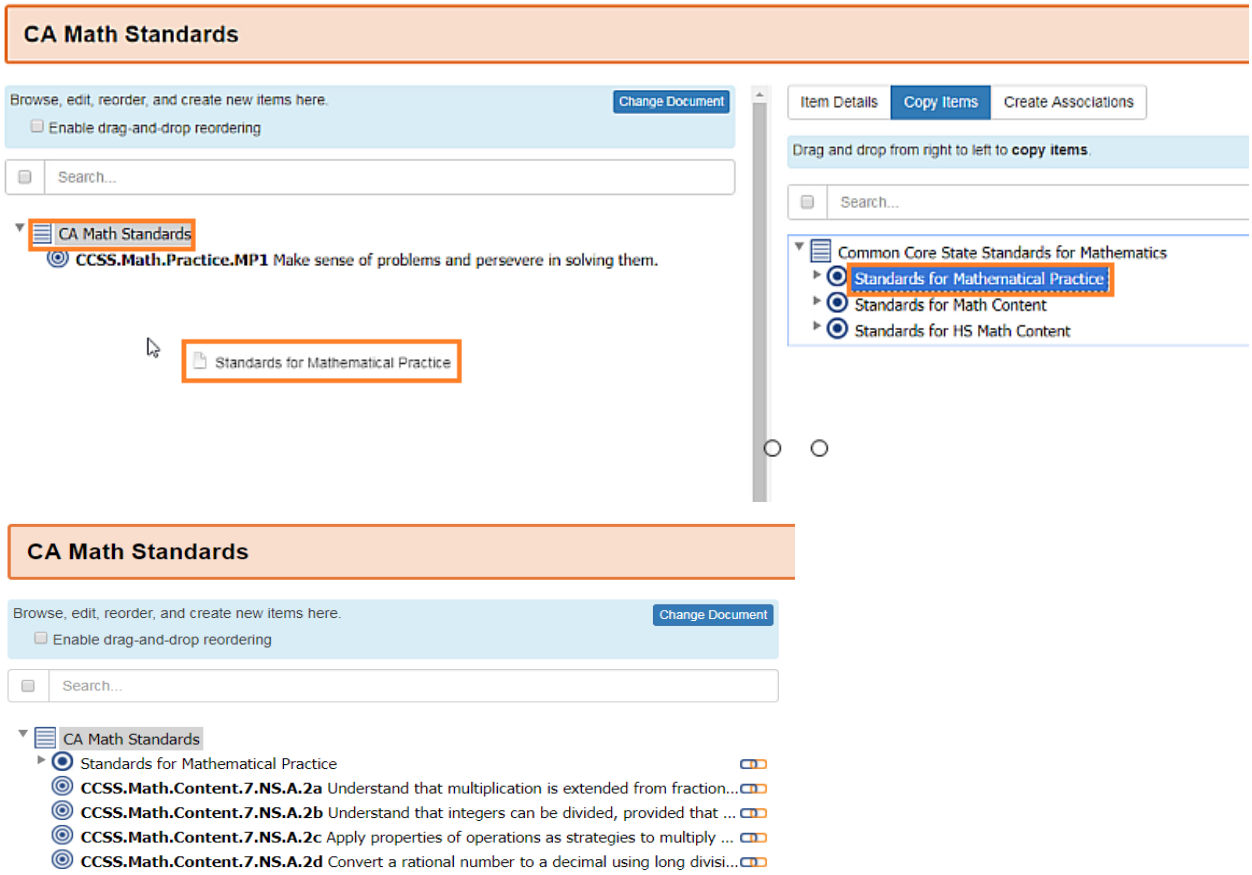
To select and copy a single item, expand the framework to the level/item desired. Click on the item and drag and drop it to the appropriate location on the left.



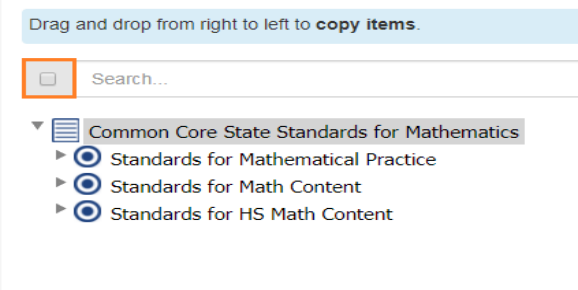
When the user releases the item on the left, the framework will refresh and the item will now display as part of the framework.



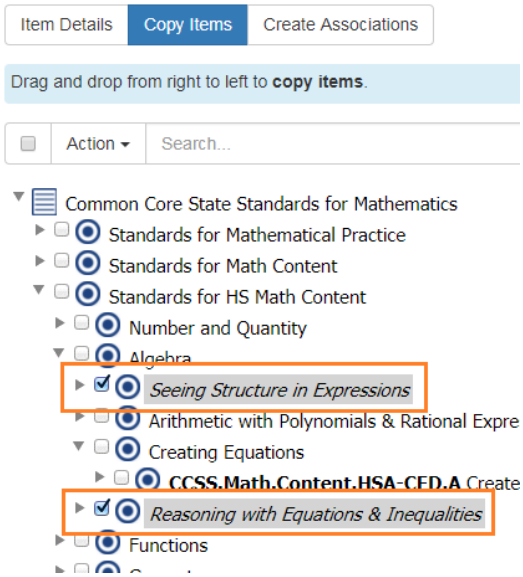
A user can select a parent item and add it to the framework, which will copy over all children under the parent as well.



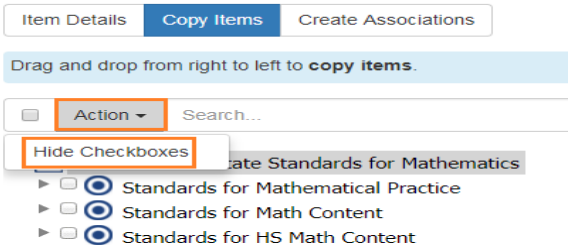
To select multiple items at once the user can expand the multi select function by clicking on the empty checkbox above the framework displayed in the Copy Items frame.

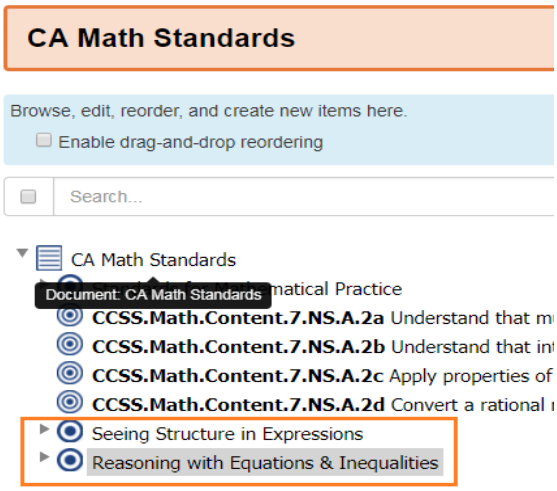


The user can now check one or more boxes and click on an item to drag and drop to the left. All items selected will move. Note if a Parent is selected all the children will also move with it even if they are not individually checked.



Note to close the multi select function, click on **Actions** and select **Hide Checkboxes**.

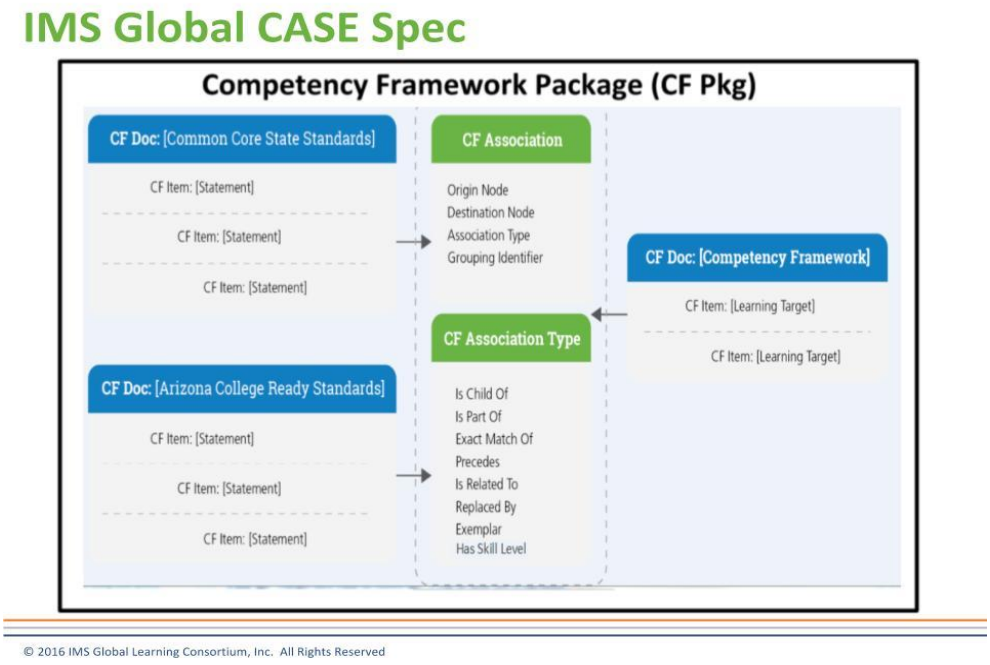




6.6 6.6 Association Management

Audience: Super Users, Super Editor, Organization Admins, Organization Editor

OpenSALT provides users with a robust way to include associations following the IMS GLobal CASE Standards. The application enables users to connect frameworks and framework items together to bring together relationship models that integrate and blend frameworks in a meaningful way. The illustration below provides a visual mapping for how associations can connect frameworks and help to crosswalk standards.



6.6.1 6.6.1 Associations

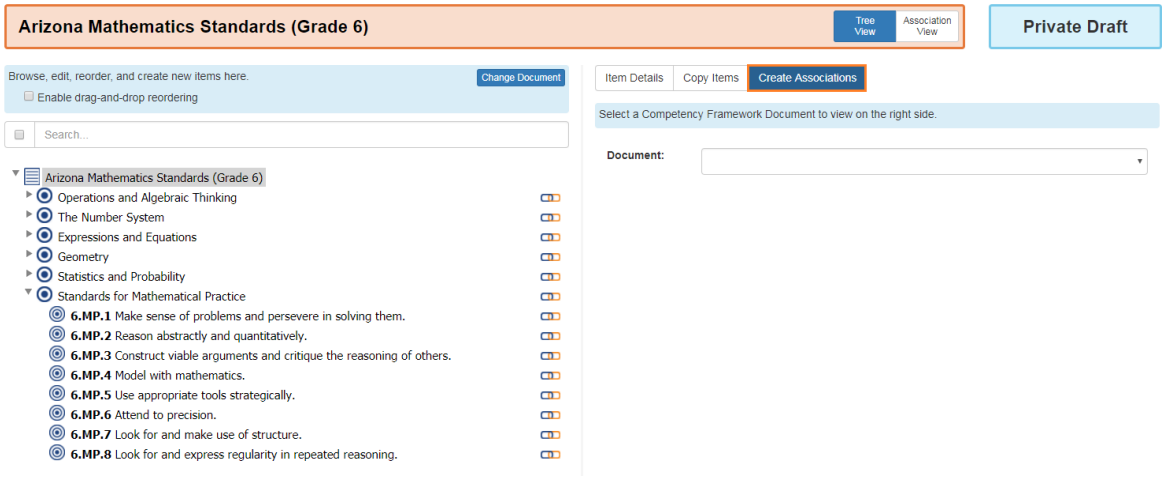
OpenSALT allows for the associations defined by . The permitted associations are:

Association Type	Description	Commentary (Unofficial, not representative of official CASE documentation)
exactMatchOf	Equivalent to. Used to connect derived CFItem to CFItem in original source CFDocument.	Used to connect derivative framework statements.
exemplar	The target/destination node is an example of best practice for the definition of the source/origin.	Point to a learning resource or other public artifacts(Unlimited).
hasSkillLevel	The destination of this association is understood to define a given skill level i.e. Reading Lexile 100, Depth Knowledge 2, or Cognitive Level (Blooms Taxonomy) etc.	Used to capture equivalencies between different types of frameworks.
isChildOf	To represent the structural relationship in a taxonomy between parent and child. The source/origin is a child of the target/destination.	Groups children statements with their parent statement.
isPartOf	The origin of the association is included either physically or logically in the item at the destination of the association. This classifies an item as being logically or semantically contained as a subset of the destination.	As an example, if three skills of a framework are subsets of one skill in another.
isPeerOf	The source/origin is a peer of of the target/destination.	Equivalence across frameworks when no grade level is given.
isRelatedTo	The origin of the association is related to the destination in some way that is not better described by another association type.	Use is acceptable for learning resources but not as specific as assessment results
precedes	The origin of the association comes before the destination of the association in time or order.	For creating learning maps/progressions.
replacedBy	The origin of the association has been supplanted by, displaced by, or superseded by the destination of the association.	For showing equivalence across versions.

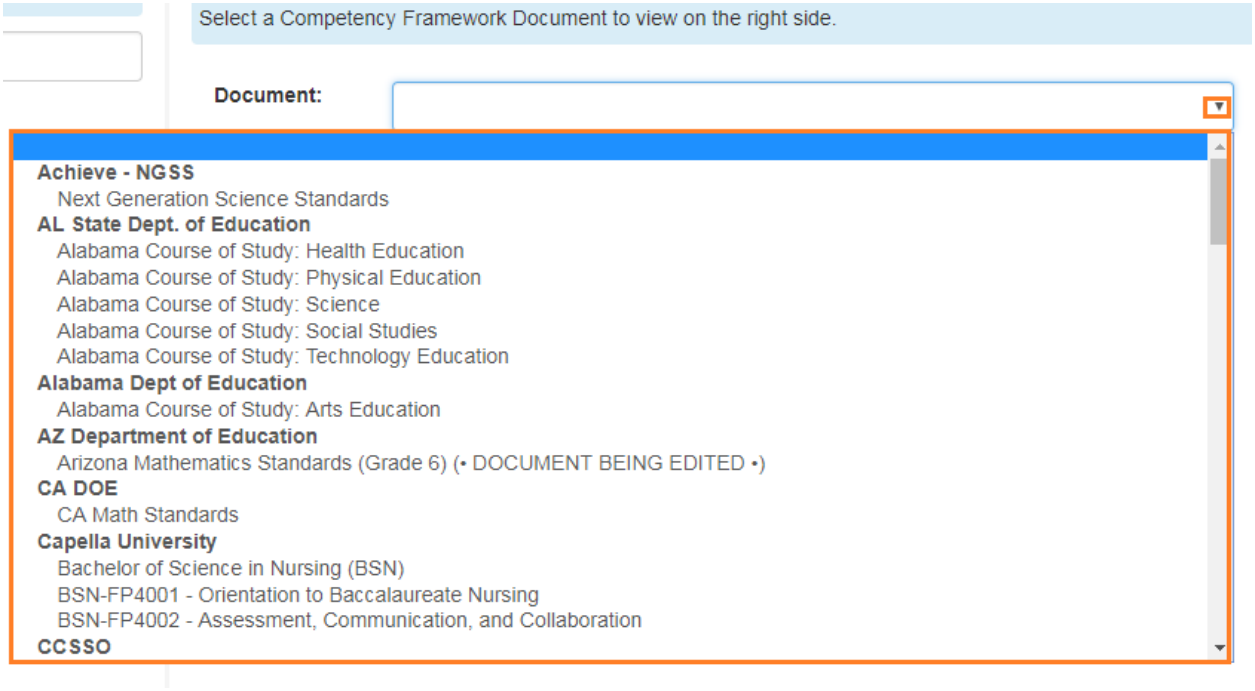
6.6.1.1 Create Associations

OpenSALT allows users to create associations to and between Framework items that either reside in the same OpenSALT instance or to any other outside CASE-compliant system.

To create an association, open the Framework target in the **Framework Dispaly** page.then click **Create Associations** to open the Associations frame.



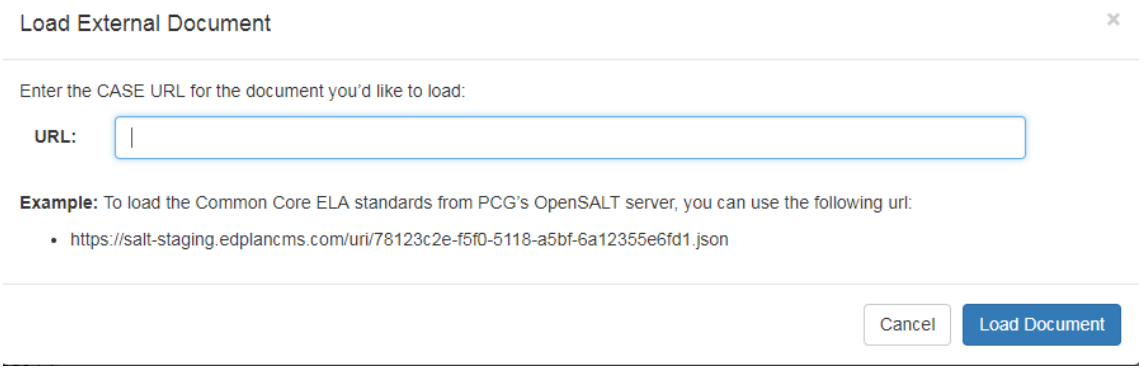
In the Document dropdown, select the framework you want to use with the associations. If the framework is within the OpenSALT server instance, the framework name will display organized by the owning Organization. In OpenSALT 2.3 this can be changed later by super editors.



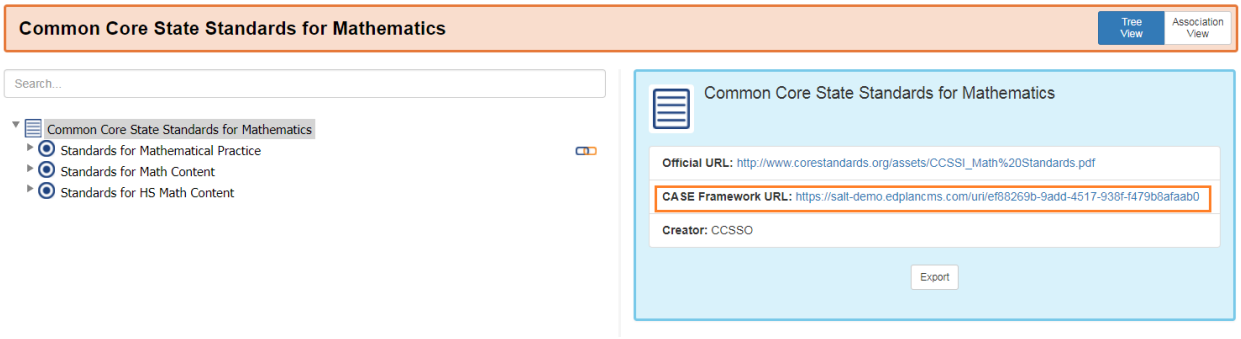
Click on the required Framework to display the framework and items.

Alternatively you can select The final option: **Load an "external" document by url...** This allows the user to select any CASE compliant framework that is stored on an external location.

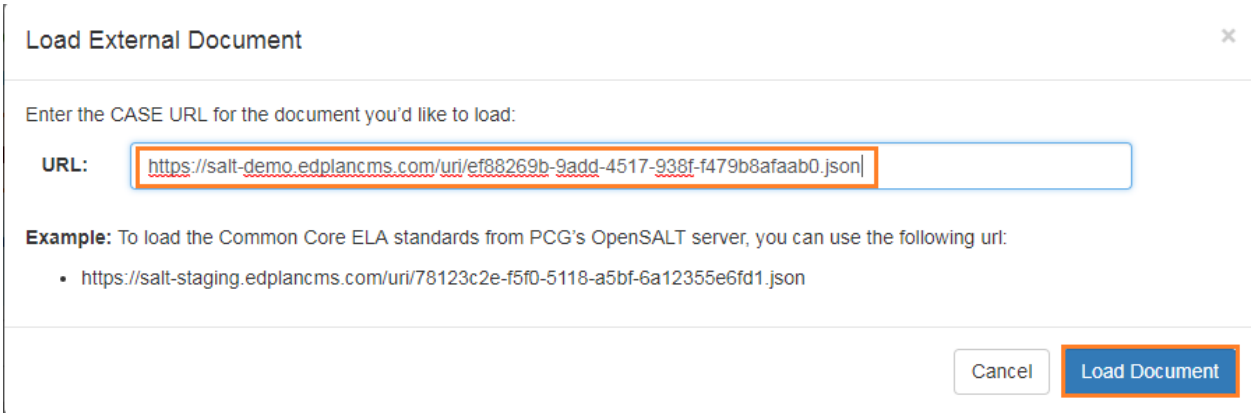
The **Load External Document** window will display. The user will need to copy and paste, or type in the url for the Case-compliant framework.



In instances of OpenSALT, the **Case Framework URL** can be found on the **Framework Display** page in the **Item Detail** frame.



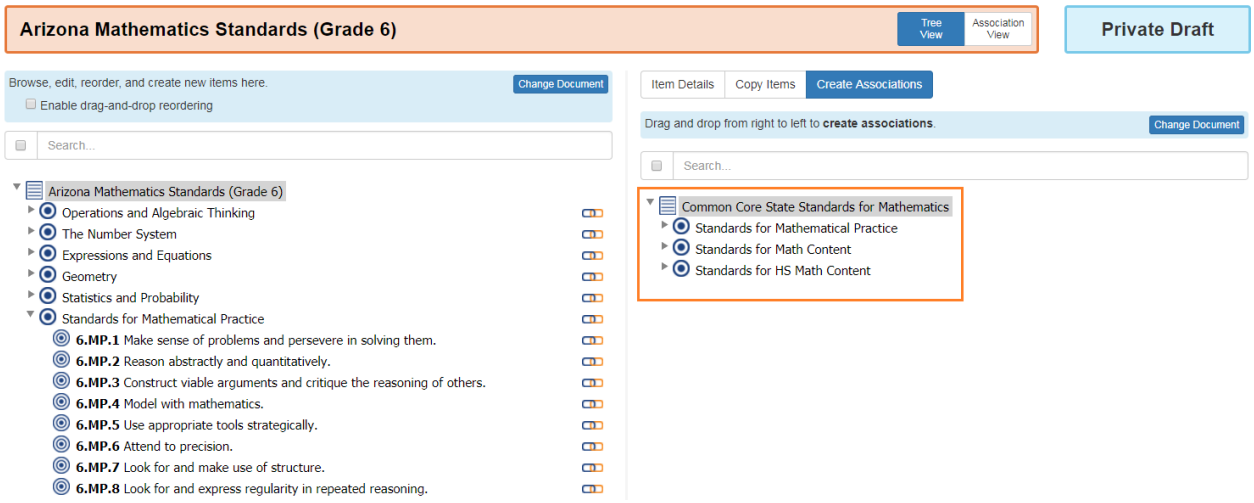
Note if the copied URL does not have the .JSON extension, you will need to add it for the document to load.



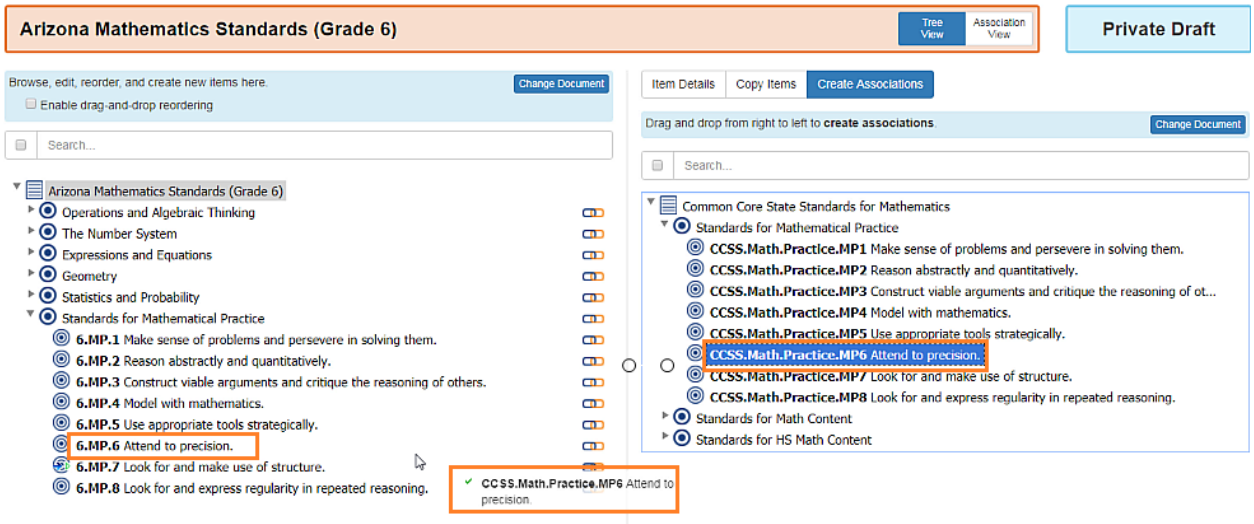
To load the framework, click the **Load Document** button.

To cancel and return to the **Framework Display** page, click the **Cancel** button.

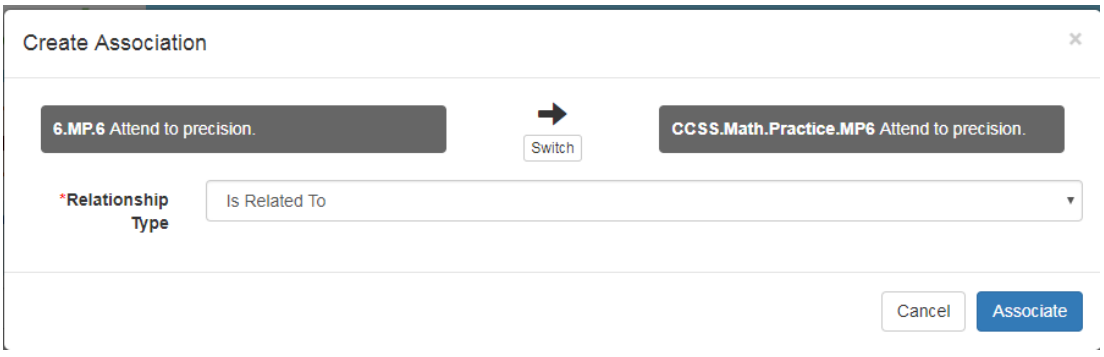
The selected framework will display in the **Create Associations** frame.



Select the item from the right by clicking on it and dragging it over the associated item on the left.



The Create Association window will display showing the item that was both dragged and dropped as well as the item that it was connected to.

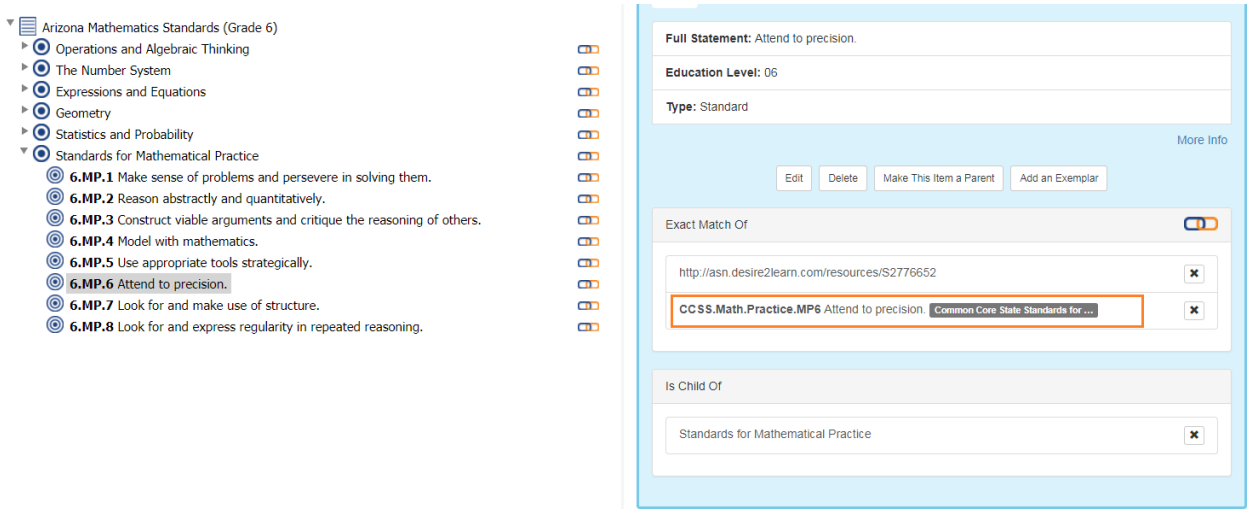


The user can select the **Relationship Type** from the drop down and change the relationship arrow as needed by clicking on the **Switch** button.

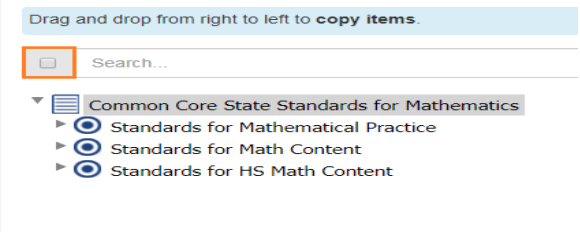
To save the association, click the **Associate** button.

To cancel and abandon the association, click the **Cancel** button.

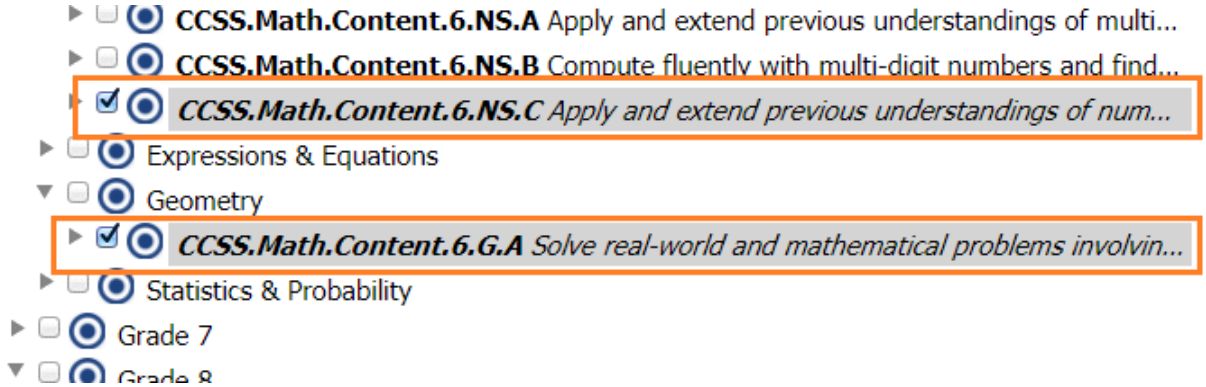
The **Framework Display** page will refresh. If the user clicks on the item in the **Framework Display** and then clicks **Item Details** , the newly created association will be visible.



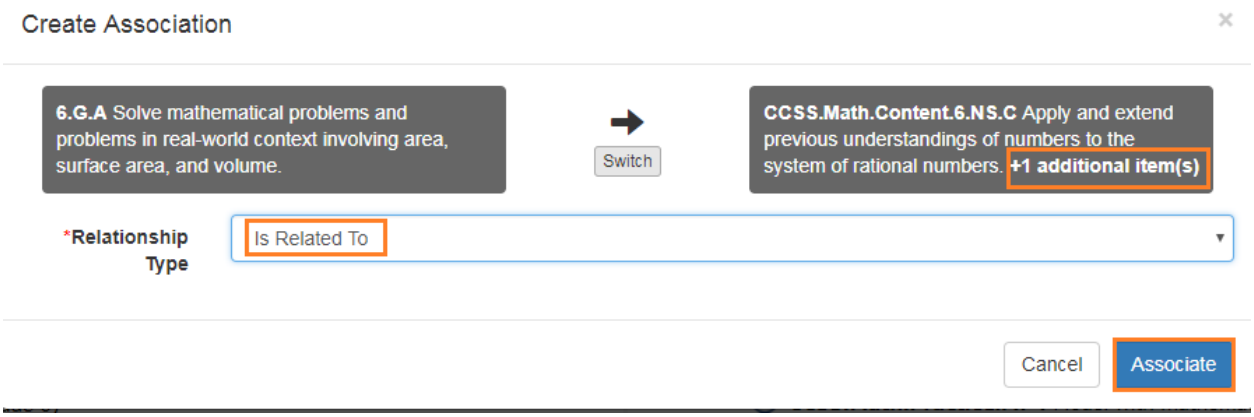
To add several of the same type of association to one item, the user can open the multi select function by clicking on the empty checkbox above the framework displayed in the **Create Associations** frame.



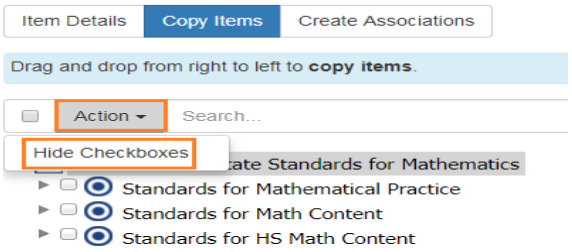
The user can now check one or more boxes and click on an item to drag and drop to the left. All items selected will become part of the association. Note if a Parent is selected all the children will associate even if they are not individually checked.



When the users drags the items to the the left and associates them with an item, the **Create Associations** window will display. The user will note that the list will only display the first of the associated item, and will indicate there are additional items selected. Also note all items will have the same association, as you can only select one association type. Set the associations as required and click the Associate button.



Note to close the multi select function, click on **Actions** and select **Hide Checkboxes**.



6.6.1.2 Edit Associations

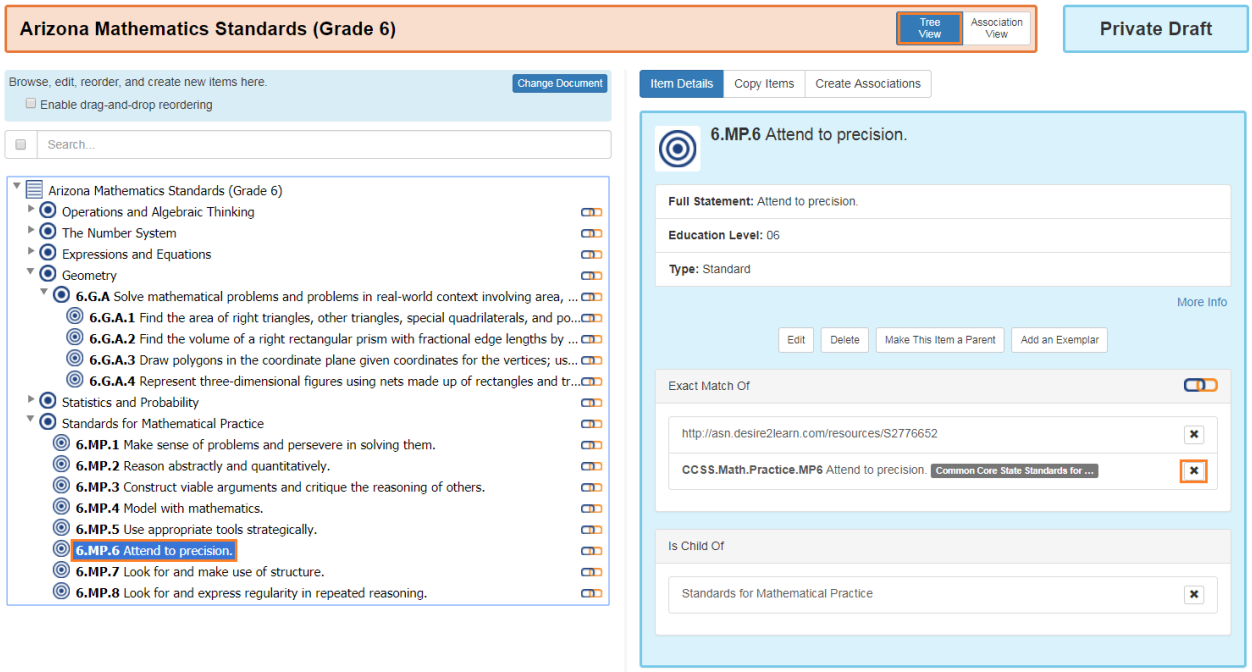
To Edit an association, the user must delete the association and create a new association. PLease see section for how to delete an association and for how to create a new association.

6.6.1.2 Delete Associations

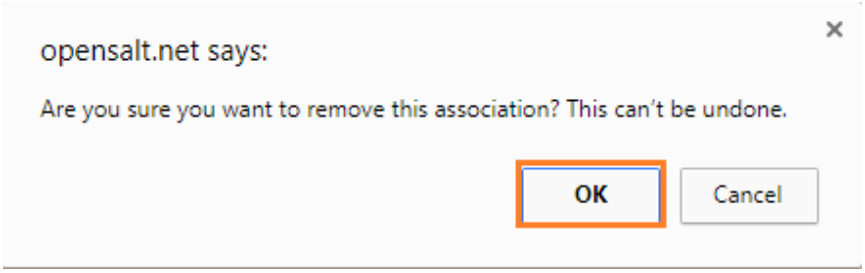
Users can either use the **Tree View** or the **Association View** to delete associations.

6.6.1.2.1 Delete Associations from Tree View

On the **Framework Display** page, click on the **Tree View** button if not already selected. Then locate the association in the I**tem Detail** frame. Click the **X** icon next to the association to remove.

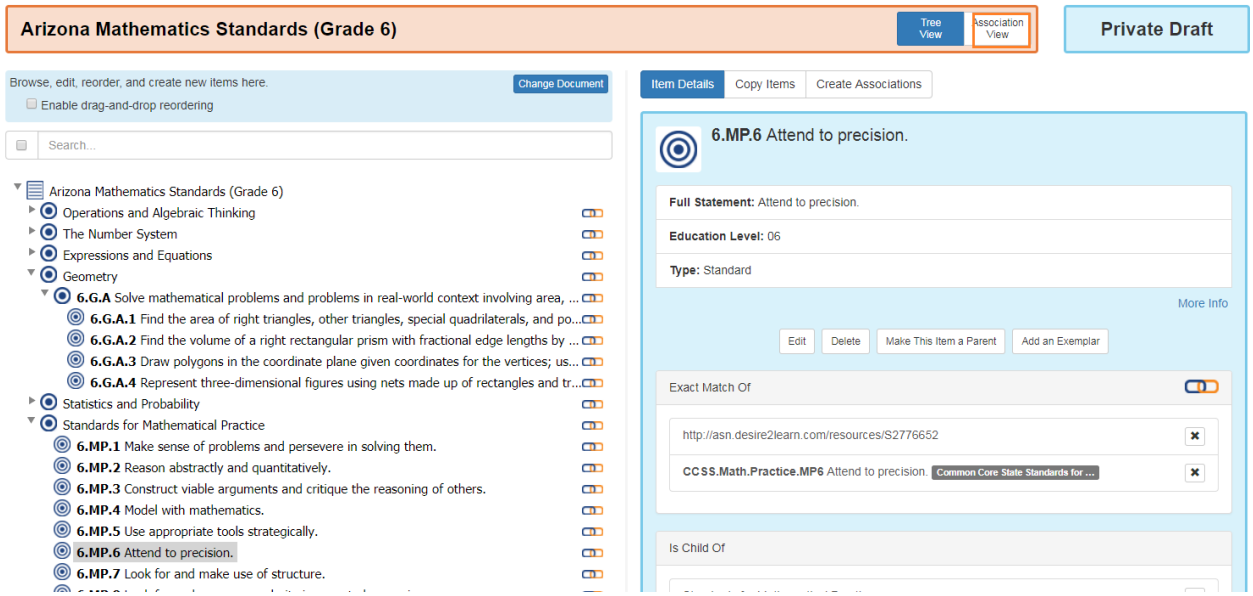


The user will get a popup window warning that he action can not be undone. If the user wants to proceed and delete the association, the user will click the **OK** button. Otherwise the user can cancel by clicking on the **Cancel** button.

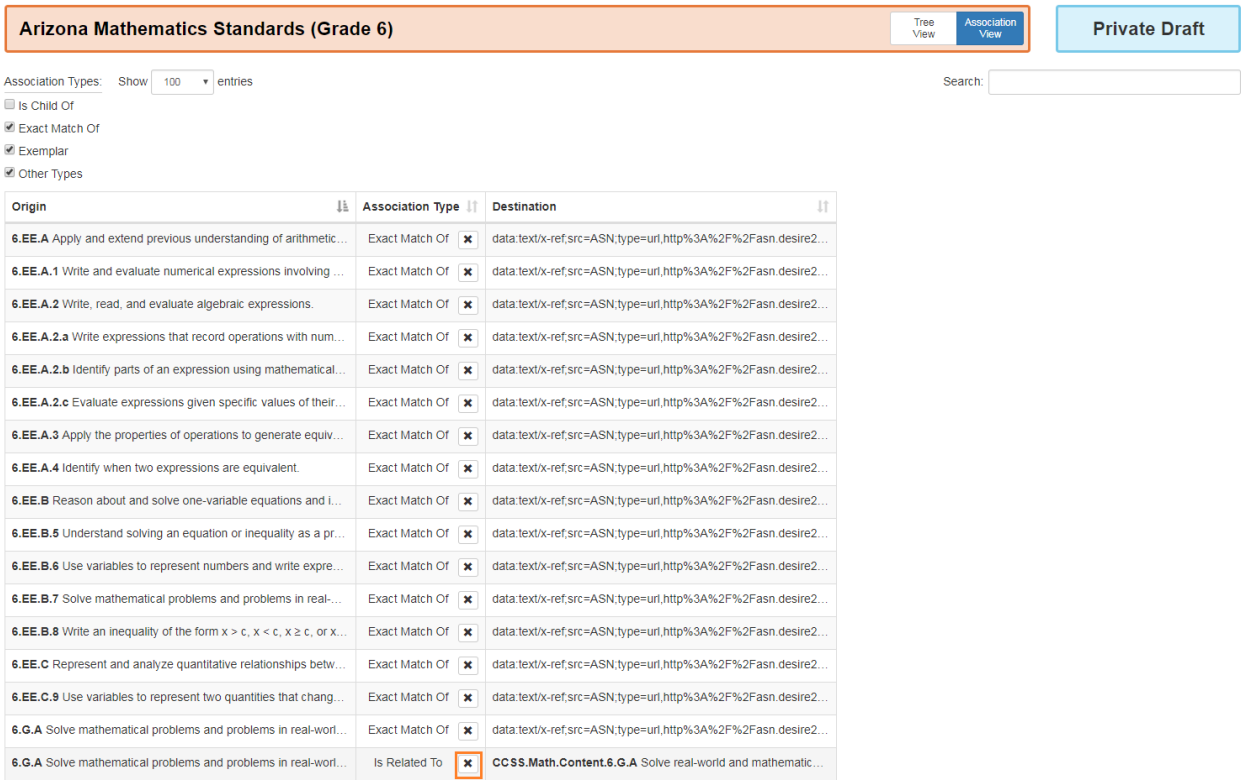


6.6.1.2.2 Delete Associations from Association View

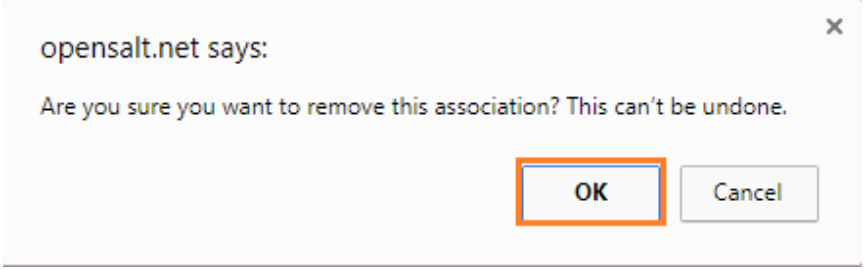
On the **Framework Display** page, click on the **Association View** button if not already selected.



Locate the association in the in the list to be deleted. Click the **X** icon next to the association to remove.



The user will get a popup window warning that he action can not be undone. If the user wants to proceed and delete the association, the user will click the **OK** button. Otherwise the user can cancel by clicking on the **Cancel** button.

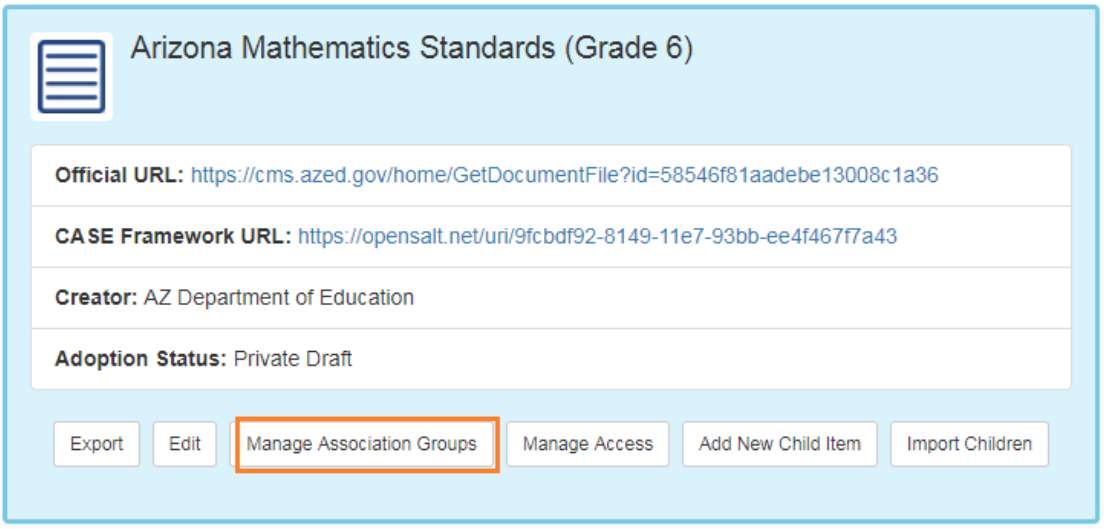


6.6.2 6.6.2 Association Groups

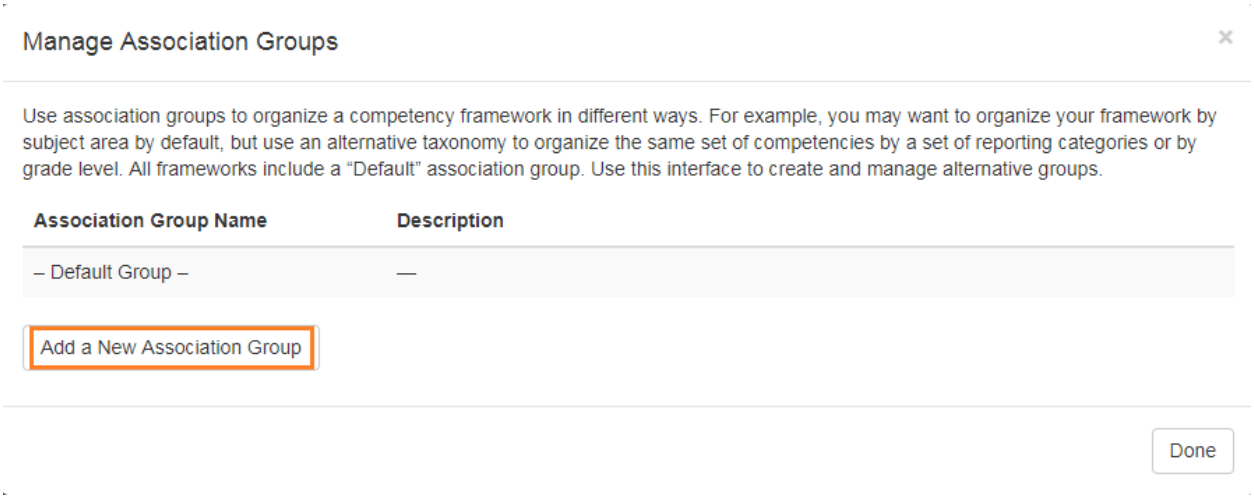
Audience: Super Users, Super Editor, Organization Admins, Organization Editor

Associations can belong to an association group. There is a selector to filter the view for only those items and associations for a specific group, if there are any. If there are no group, all associations belong to the default "Null" group. If there are any association group, the default view is "All" association groups. The Association group filter drop list allows you to choose the "null" group, the "All" group or any specific group. In the view, an association that belongs to an association group is tagged in the display as such:

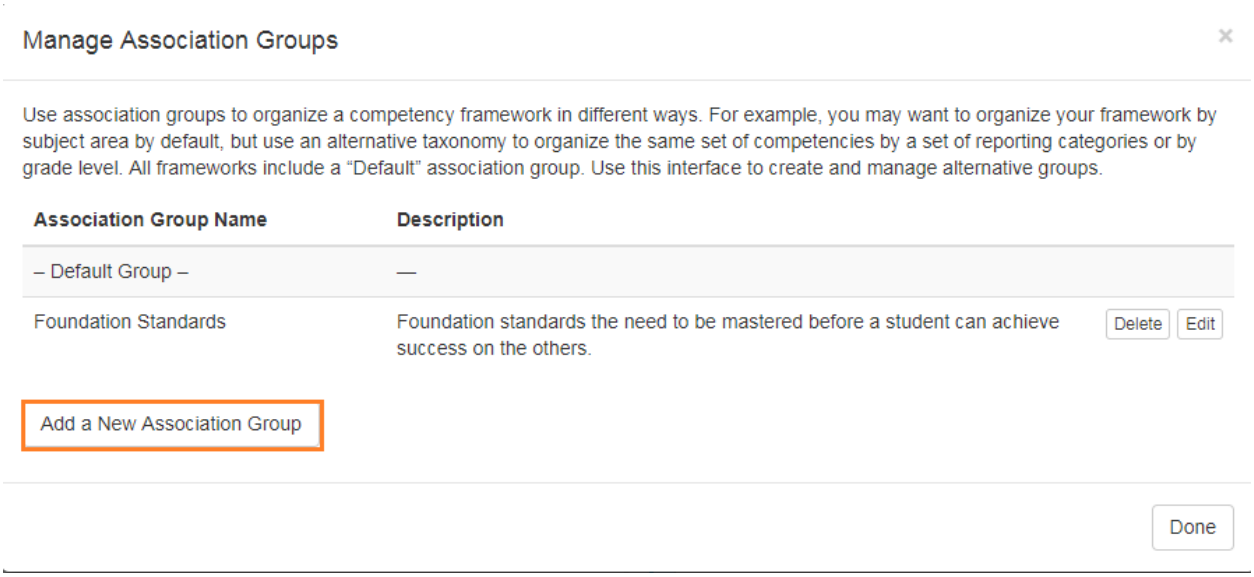
On the **Framework Display** page, click the **Manage Association Groups** button in the **Item Detail** frame.



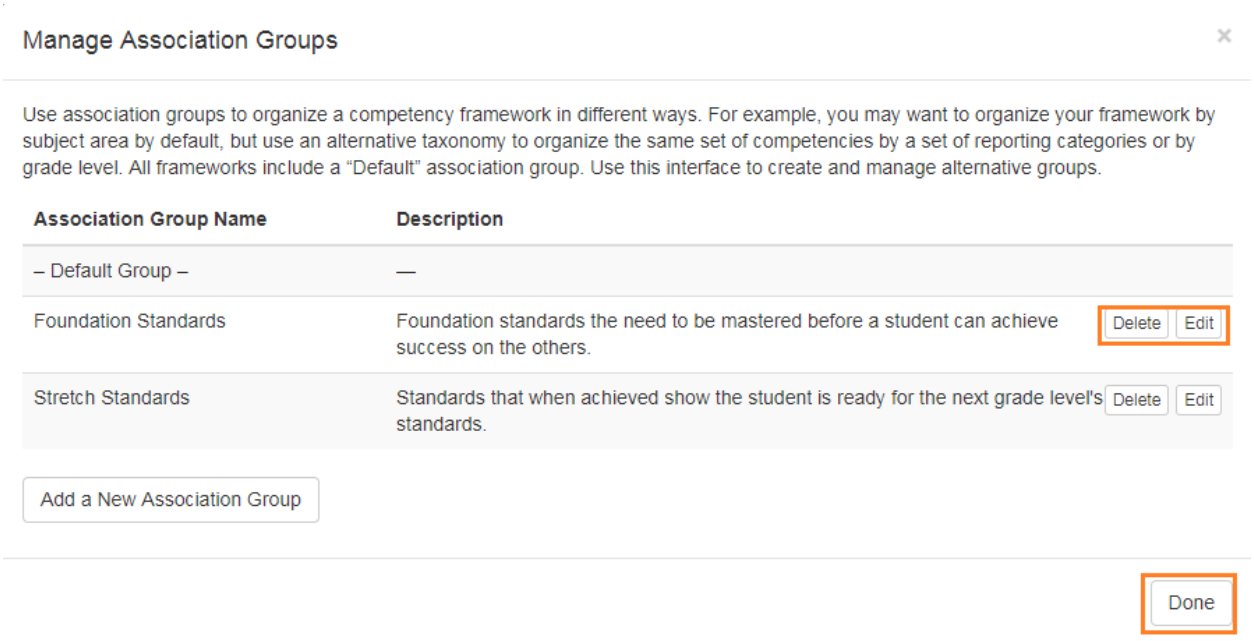
The **Manage Association Groups** window will display. Click on the **Add a New Association Group** button.



The **Add New Association Group** window will display. The user will need to enter a **Title** and the optional **Description**. Next click on the **Create** button.

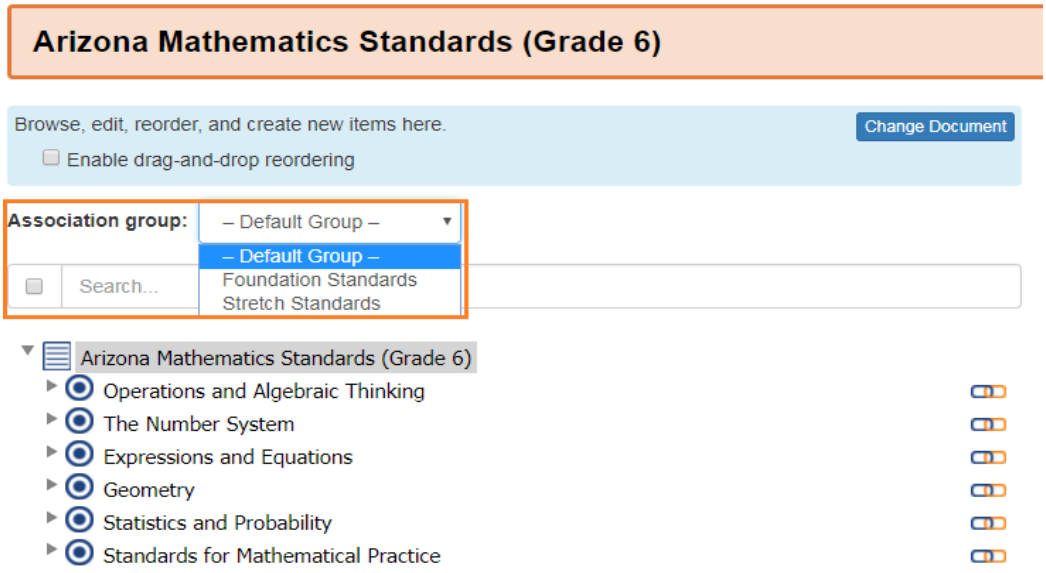


The user will repeat the process to add any needed **Association Groups**.



After creating the required Association Groups, the user can **Edit** a Group, **Delete** a Group or click **Done**.

The filter option for the **Association Groups** will now display on the **Display Frameworks** page.



The user will now need to associate items with the appropriate groups.

Note that association groups working is new in OpenSALT 2.3.

6.6.3 6.6.3 Crosswalk Associations

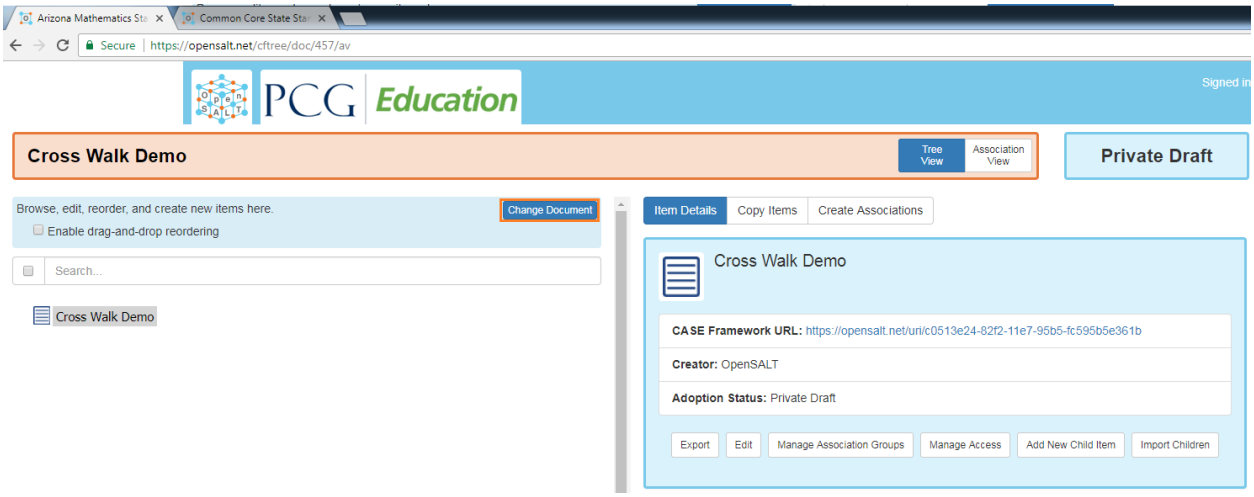
Audience: Super Users, Super Editor, Organization Admins, Organization Editor

OpenSALT allows for users to create crosswalks between frameworks. To create a crosswalk the user will need to start with an empty framework. For instructions on creating a framework see section .

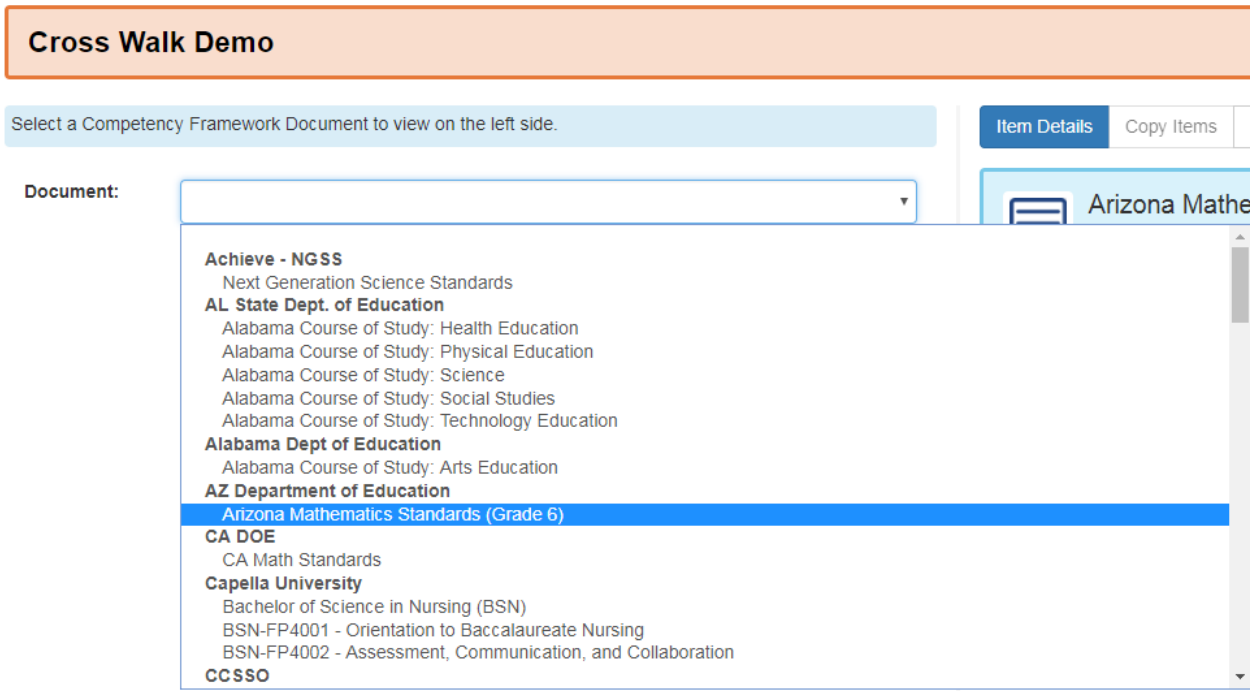
How-To Video

6.6.3.1 Create Crosswalk Associations

Open the framework to the **Display Framework** page. In the click on **Change Document** to select the first framework to use in the crosswalk.

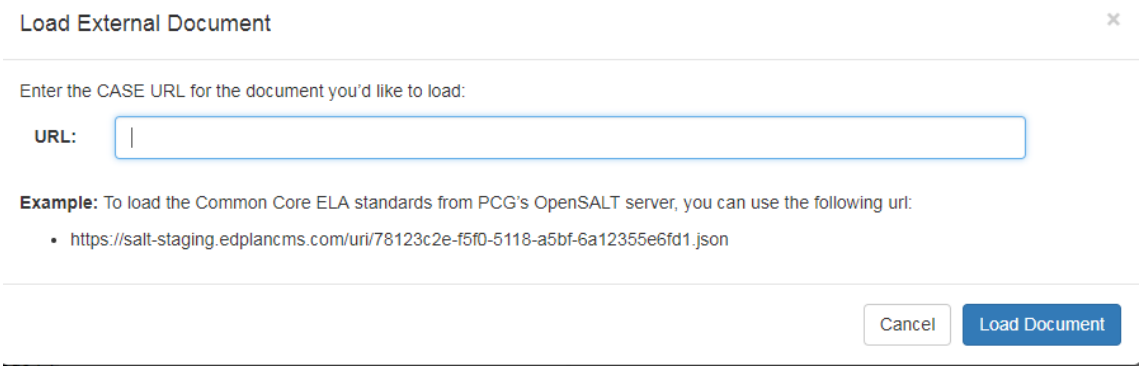


This will open a Document dropdown. The user can select a framework that is in their current OpenSALT server from the list by clicking on the desired framework.

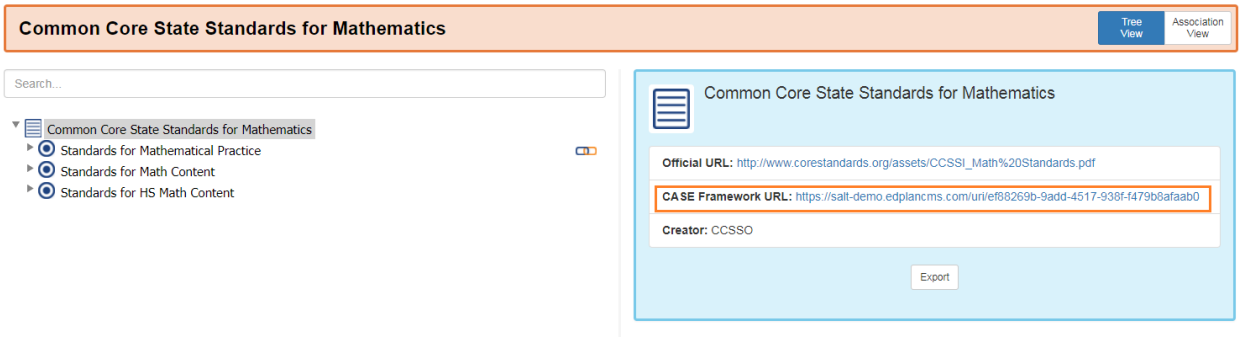


Alternatively the user can connect to a framework that is outside their system by selecting the final option: **Load an "external" document by url...** This allows the user to select any CASE compliant framework that is stored on an external location.

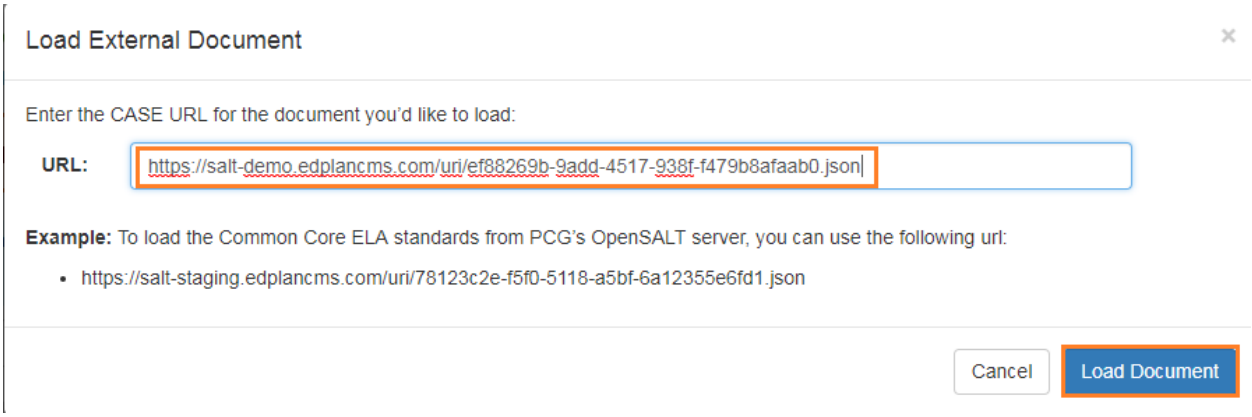
The **Load External Document** window will display. The user will need to copy and paste, or type in the url for the Case-compliant framework.



In instances of OpenSALT, the **Case Framework URL** can be found on the **Framework Display** page in the **Item Detail** frame.

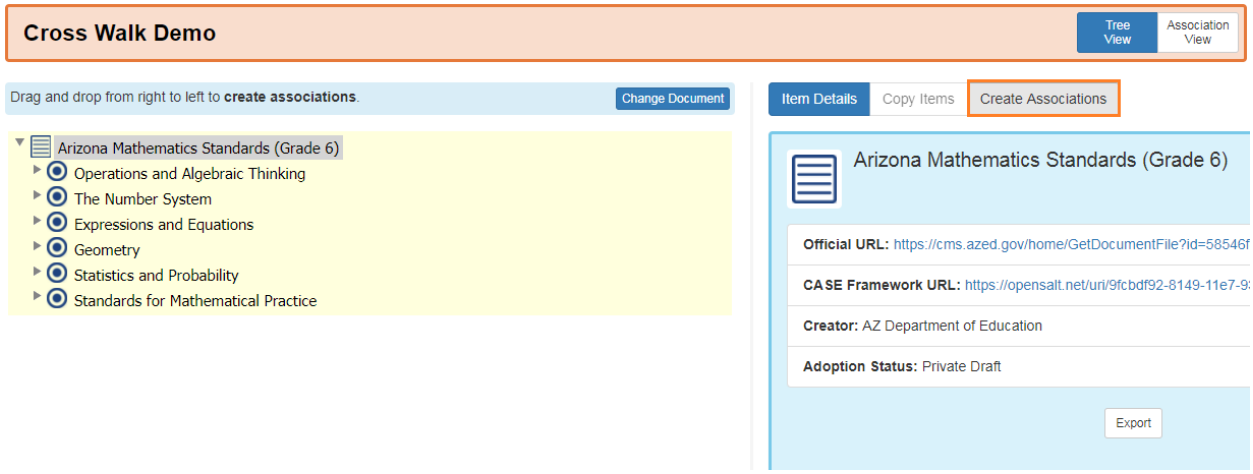


Note if the copied URL does not have the .JSON extension, you will need to add it for the document to load.

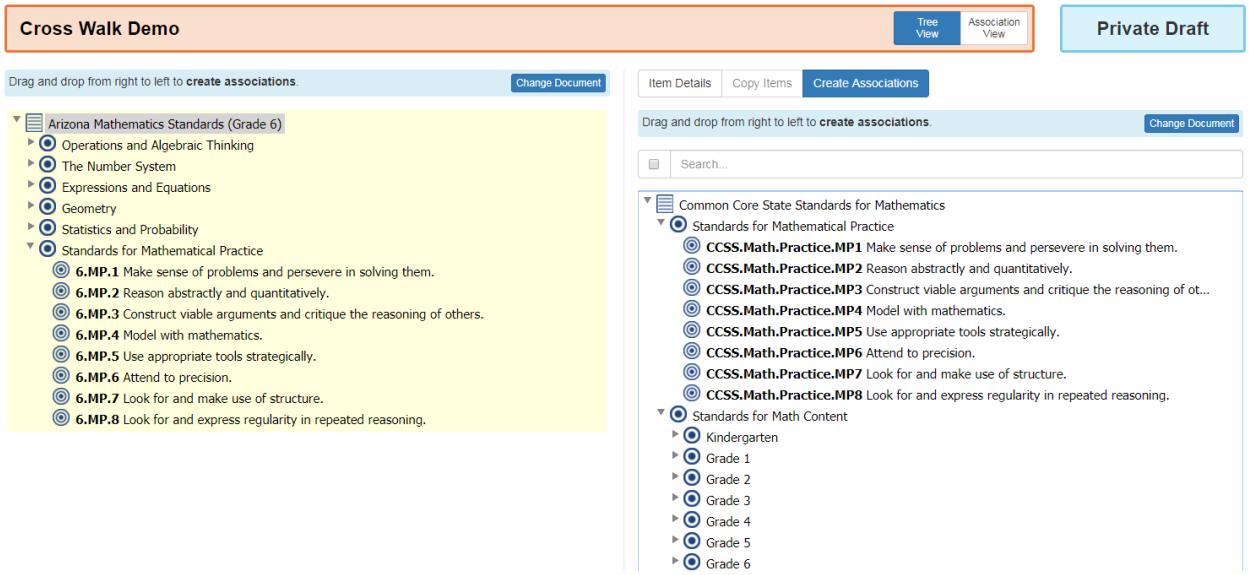


To load the framework, click the **Load Document** button.

Next the user will click on the **Create Association** option above the **Item Detail** frame.



A Document dropdown will display to allow the user to select the second framework to be selected. Again the user can select an existing framework from their OpenSALT server or can select an outside CASE-compliant framework.



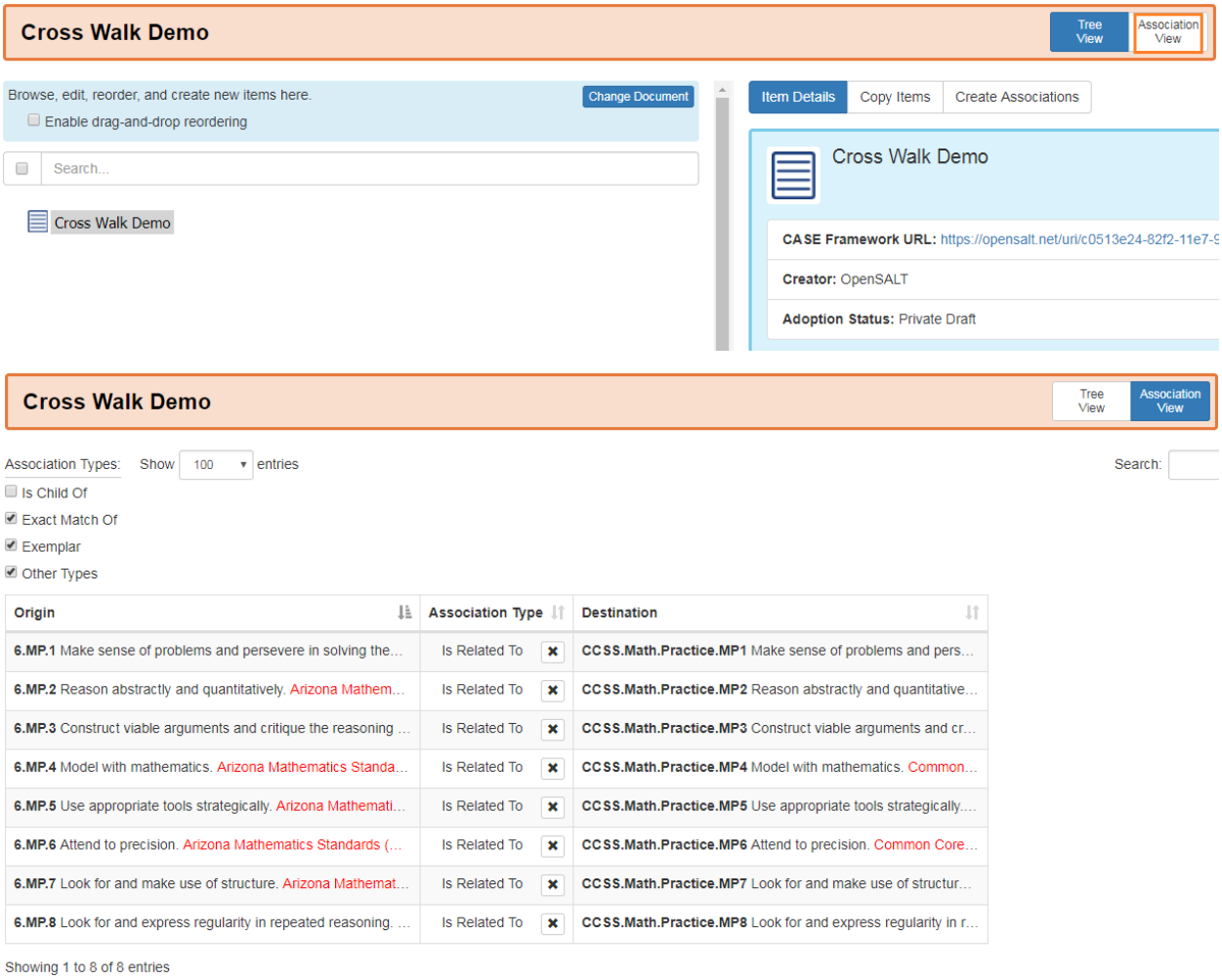
The user can now use drag and drop to select items from the right and drag to connect to items on the left to establish the cross walk. After an association is connected with the drag and drop, the Create Association window will display and the user can define the association.

When the user is satisfied with the association definition, click on the **Associate** button.

To cancel, click on the **Cancel** button.

The user should repeat this process to connect all required framework items for the crosswalk.

To view the cross walk, the user will click on the **Association View** at the top of the screen.

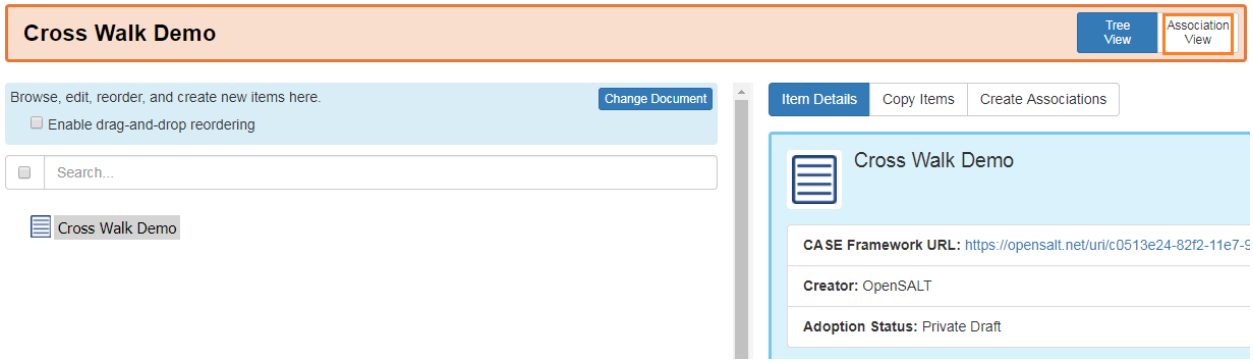


6.6.3.2 Edit Crosswalk Associations

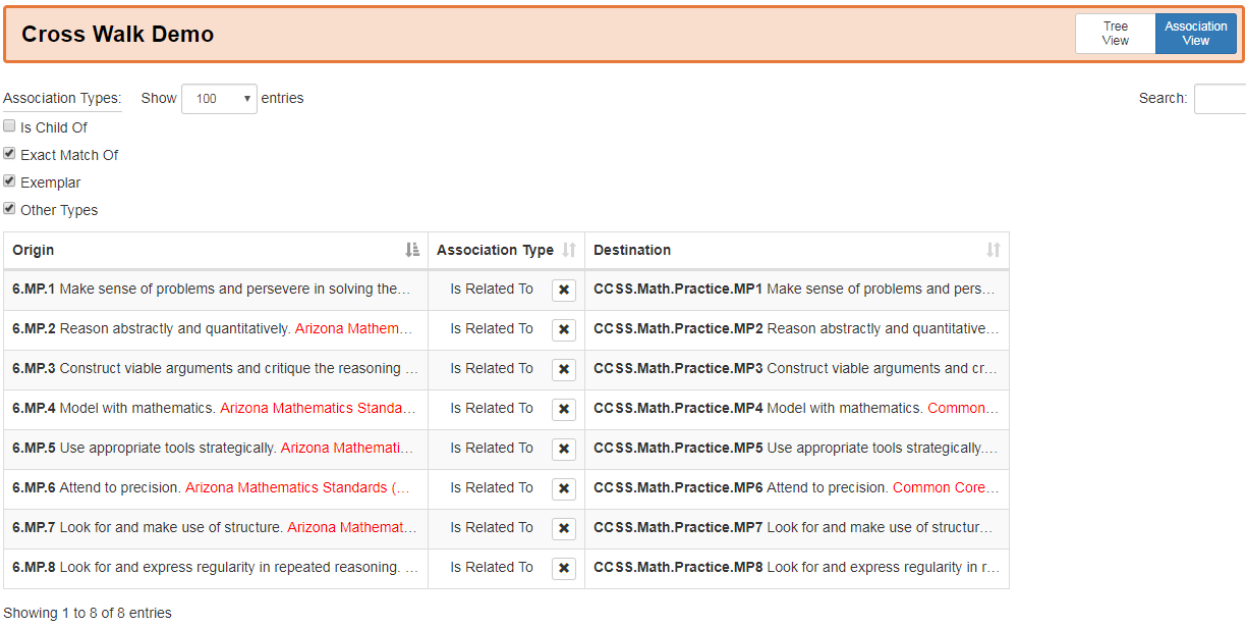
To Edit a crosswalk association, the user must delete the associations and create new associations. Please see section for how to delete an association and for how to create a new association.

6.6.3.3 Delete Crosswalk Associations

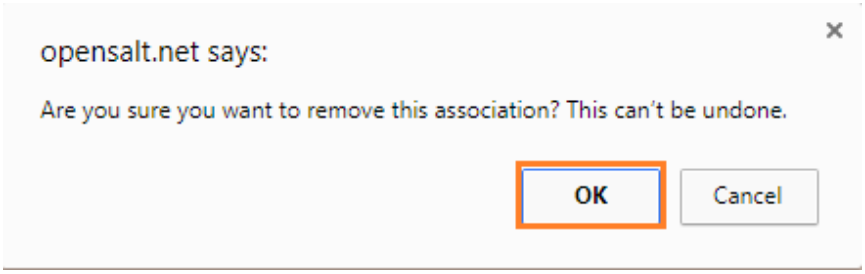
On the **Framework Display** page, click on the **Association View** button if not already selected.



Locate the association in the in the list to be deleted. Click the **X** icon next to the association to remove.



The user will get a popup window warning that he action can not be undone. If the user wants to proceed and delete the association, the user will click the **OK** button. Otherwise the user can cancel by clicking on the **Cancel** button.



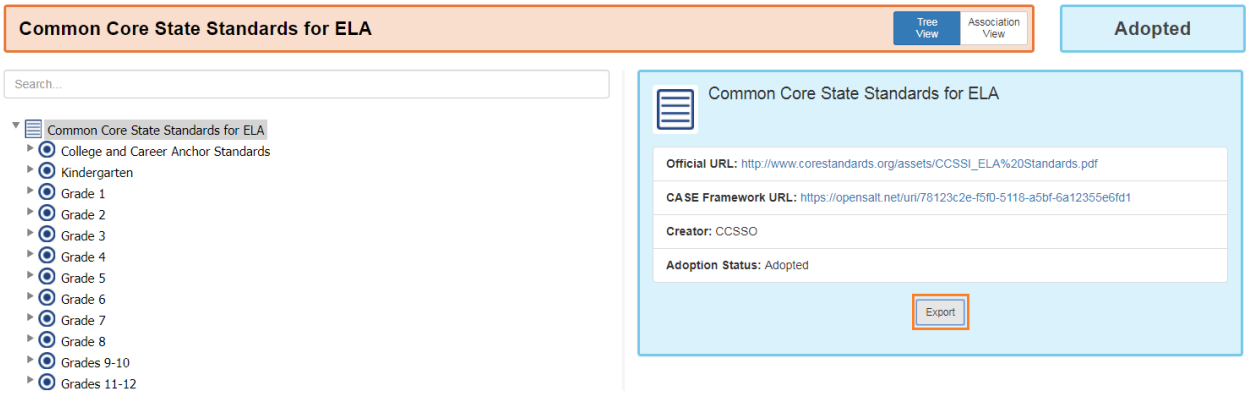
Alternatively if the entire crosswalk needs to be deleted and all associations, the user can delete the framework instead of removing all associations. Please see section for instructions on deleting a framework.

6.7 6.7 Export Frameworks Packages

Audience: Super Users, Super Editor, Organization Admins, Organization Editor, Public

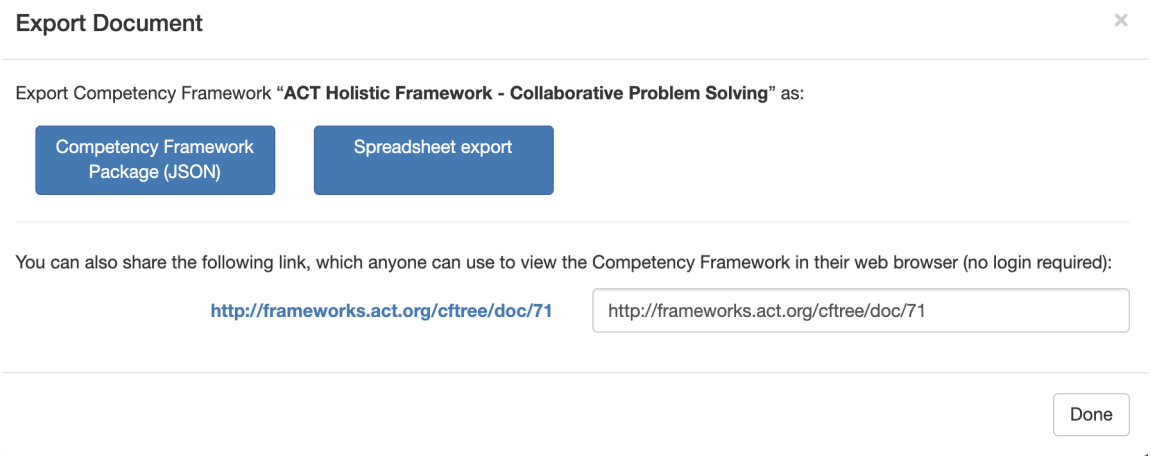
OpenSALT is designed to enable all users roles to **Export** complete framework packages or documents. While **Public** users can only export framework packages that are in an **Adopted**, **Draft**, or **Depreciated** release status, credentialed users will have the ability to export **Draft** and **Private Draft** framework packages as well.

To **Export** a framework package the user needs to be on the **Framework Display** page for the selected framework and then click on the **Export** button.



An Export window will display to provide the user with several **Export** options:

- Competency Framework Package (JSON)
- Spreadsheet Export

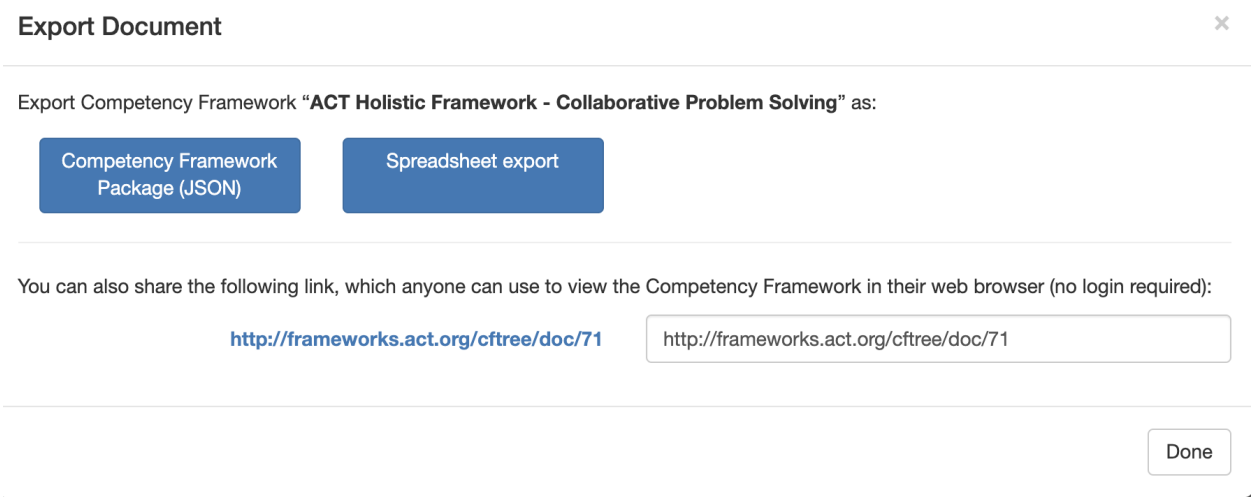


The user can select an **Export** option or can click on the **Done** button or the '**X**' at the top of the window to close the **Export** window. Each **Export** type will be discussed in the following subsections.

6.7.1 6.7.1 Export Competency Framework Package (JSON) File

The first **Export** option is the **Competency Framework Package (JSON)**. This option allows users to extract the data for the framework packaging in the JavaScript Object Notation (JSON) file format. This is a lightweight, text-based, language-independent data interchange format that allows for easy ingestion into other system and applications using a common standard programming language.

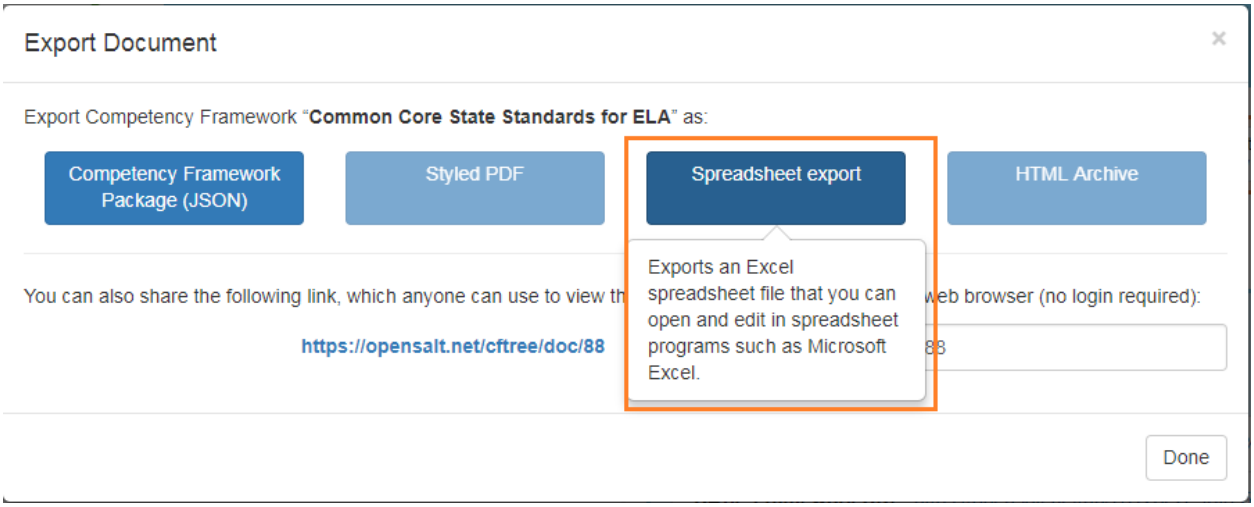
When a user mouses over the **Competency Framework Package (JSON)** button, a soft explanation is displayed that advises the user of the button's intended purpose: 'Exports a JSON file using the IMS-standard format. This is the best format to user for Archiving Frameworks.'



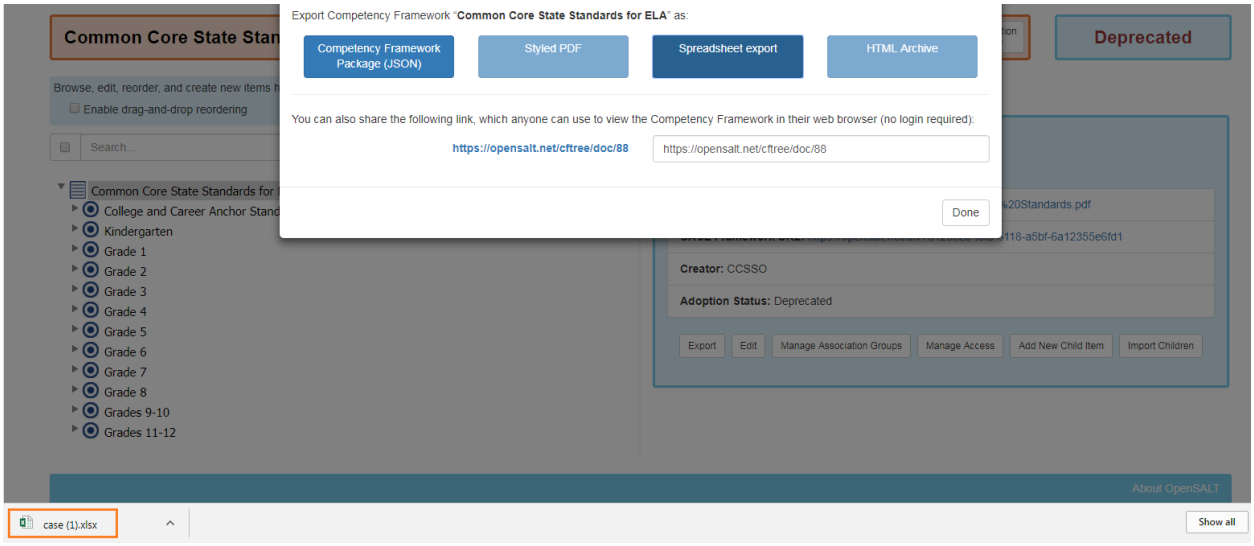
6.7.2 6.7.2 Export Spreadsheet

The second **Export** option is the **Spreadsheet Export.** This option allows users to extract the data for the framework packaging in a spreadsheet file that can be opened/viewed/edited in a program like Microsoft's Excel or Google Sheets.

When a user mouses over the **Spreadsheet Export** button, a soft explanation is displayed that advises the user of the button's intended purpose: 'Exports as an Excel spreadsheet file that you can open and edit in spreadsheet programs such as Microsoft Excel. '



To generate a spreadsheet file for the Framework Package, the user will click on the **Spreadsheet Export** *button.* The browser will automatically create and download a case xlsx file for the Framework Package. The user will need to retrieve the file following the browser's specific protocol and path. The image below shows the download in Chrome. Note the file is in the download frame at the bottom of the page. It has also been added to the user's Download folder within the user's My Documents.

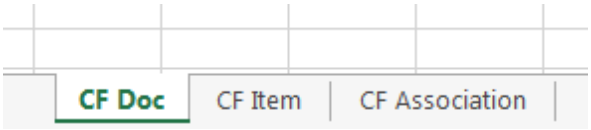


To open the spreadsheet file the user will need to identify a compatible application on their computer. The most common applications used are Microsoft Excel or Google Sheets.

Users credentialed users who are experienced with importing Frameworks into OpenSALT with the CASE template will already be familiar with this spreadsheet format. For all other users, the following details the exported spreadsheet file.

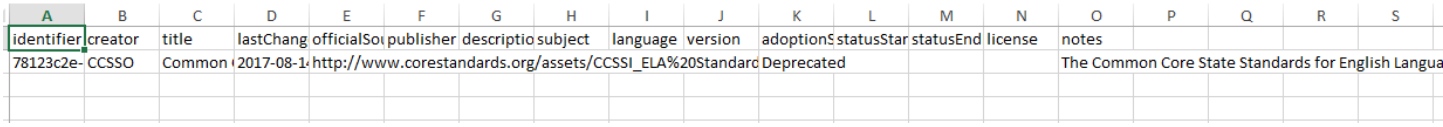
When opened the spreadsheet file will have three tabs:

1. CF Doc
2. CF Item
3. CF Association



6.7.2.1 CF Doc Tab

The purpose of the first tab, CF Doc, is to provide the user with basic background information on the Framework Package.



The columns on the spreadsheet include the following:

- *Identifier:* The identifier is intended to be used as the primary key global identifier within or external to the system. This is a mandatory field in OpenSALT and the export will contain data.
- *Creator:* The entity that authorized or created the competency framework. It could be an education agency, higher education institution, professional body. It is the owner of the competency framework (e.g CCSSO, TEA, NGSS). This is a mandatory field in OpenSALT and the export will contain data.
- *Title*: The title as it appears on the cover of the Official Source artifact, although it may be a title created by the Publisher. This is a mandatory field in OpenSALT and the export will contain data.

- *LastChangeDateTime:* The field is used to establish any change, not just major version revisions. This is a mandatory field in OpenSALT and the export will contain data.
- *OfficialsourceURL*: The URL of the artifact adopted by the Standard Setting Entity. Often this document is published in html and/or as pdf and is used by the standard setting entity as part of its approval process. Since it is not the intent of this specification to fully reproduce the human-facing content and formatting of the source document, it is recommended that this document be transmitted as part of the competency framework package. This is an optional field in OpenSALT. If best practices are not followed, this field may be blank.
- *Publisher:* The entity that loads and publishes the Framework. Note that in many cases, the Standard Setting Entity may lack technical capabilities to publish the Competency Framework in a standard format so a third party may be displayed. This is an optional field in OpenSALT. If best practices are not followed, this field may be blank.
- *Description:* The description is typically created by the the Publisher as a standard description of the Competency Framework.This is an optional field in OpenSALT. If best practices are not followed, this field may be blank.
- *Subject:* This is a string expressing the general subject area of the Competency Framework (e.g. Mathematics). This is an optional field in OpenSALT. If best practices are not followed, this field may be blank.
- *Language:* HTML Language Country Code VIA- country code from . This is an optional field in OpenSALT. If best practices are not followed, this field may be blank. However OpenSALT assumes English if not other language value is entered.
- *Version:* This is used to separate any version information expressed by the Official Source artifact. Once and CF Pkg has been approved and published, any changes to an CF Item will constitute a new version of the CF Doc. This is an optional field in OpenSALT. If best practices are not followed, this field may be blank.
- *AdoptionStatus:* Adoption status displays the Framework's current status as Draft, Private, Draft, Adopted, or Deprecated. OpenSALT assumes Adopted as the default if no status is specifically selected for the framework. This is an optional field in OpenSALT. If best practices are not followed, this field may be blank. OpenSALT assumes Adopted as the default if no status is specifically selected for the framework.
- *StatusStartDate:* The date that the CF Doc status started. This is an optional field in OpenSALT. If best practices are not followed, this field may be blank.
- *StatusEndDate:* This date is often only known when a new status is started. This is an optional field in OpenSALT. If best practices are not followed, this field may be blank.
- *License:* Systems may filter for content with particular licences to support discovery. This is an optional field in OpenSALT. If best practices are not followed, this field may be blank.
- *Notes:* Notes or comments generated by the Framework Publisher about the context of the Framework. This is an optional field in OpenSALT. If best practices are not followed, this field may be blank.

6.7.2.2 CF Item Tab

The purpose of the second tab, CF Item, is to provide the user with items and content within the Framework Package.

A	B	C	D	E	F	G	H	I	J	K	L	M	N
identifier	fullStatem	humanCo	smartLeve	listEnume	abbreviat	conceptKe	notes	language	education	CFItemTy	license	lastChangeDateTime	
6fe78131-	Demonstr	CCSS.ELA-	3.6.1.1	1				en	01	Standard		2017-03-30 16:45:29	
5f9029a8-(	(Begins in	CCSS.ELA-	3.6.2.1	1				en	01			2017-03-31 23:51:30	
b2be2352-	Determin	CCSS.ELA-	3.6.3.1	1				en	01	Standard		2017-03-30 16:46:46	
6afe5972-	Demonstr	CCSS.ELA-	4.6.1.1	1				en	02	Standard		2017-03-30 18:03:30	
7638fbcc-(	Use know	CCSS.ELA-	4.6.2.1	1				en	02	Standard		2017-03-30 18:03:41	
7a4ec01a-	Determin	CCSS.ELA-	4.6.3.1	1				en	02	Standard		2017-03-30 18:03:58	
f04ef59f-c	Demonstr	CCSS.ELA-	5.6.1.1	1				en	03	Standard		2017-03-30 18:13:09	
8237b914-	Use know	CCSS.ELA-	5.6.2.1	1				en	03	Standard		2017-03-30 18:13:21	
d053c224-	Determin	CCSS.ELA-	5.6.3.1	1				en	03	Standard		2017-03-30 18:13:38	
5d70d541-	Demonstr	CCSS.ELA-	6.6.1.1	1				en	04	Standard		2017-03-30 22:28:21	
9b3348ca-	Use know	CCSS.ELA-	6.6.2.1	1				en	04	Standard		2017-03-30 22:28:38	
aa9e2dab-	Cite speci	CCSS.ELA-	9.6.1.1	1				en	06,07,08	Standard		2017-03-31 23:05:40	
7cfce4a3-f	Acquire a	CCSS.ELA-	6.6.3.1	1				en	04	Standard		2017-03-30 22:28:55	
80d9d9fe-	Demonstr	CCSS.ELA-	7.6.1.1	1				en	05	Standard		2017-03-31 17:31:22	
b5c39b97-	Use know	CCSS.ELA-	7.6.2.1	1				en	05	Standard		2017-03-31 17:31:33	
82d5b74a-	Determin	CCSS.ELA-	7.6.3.1	1				en	05	Standard		2017-03-31 17:31:51	
212179c1-	Demonstr	CCSS.ELA-	8.5.1.1	1				en	06	Standard		2017-03-31 18:04:05	
a9f295b5-	Use know	CCSS.ELA-	8.5.2.1	1				en	06	Standard		2017-03-31 18:04:15	
c7e53de3-	Determin	CCSS.ELA-	8.5.3.1	1				en	06	Standard		2017-03-31 18:04:28	
c41f1536-(	Use know	CCSS.ELA-	9.5.2.1	1				en	07	Standard		2017-03-31 18:44:48	
73e3de11-	Determin	CCSS.ELA-	9.5.3.1	1				en	07	Standard		2017-03-31 18:45:06	
88d9729c-	Demonstr	CCSS.ELA-	10.5.1.1	1				en	08	Standard		2017-03-31 18:52:46	

The columns on the spreadsheet include the following:

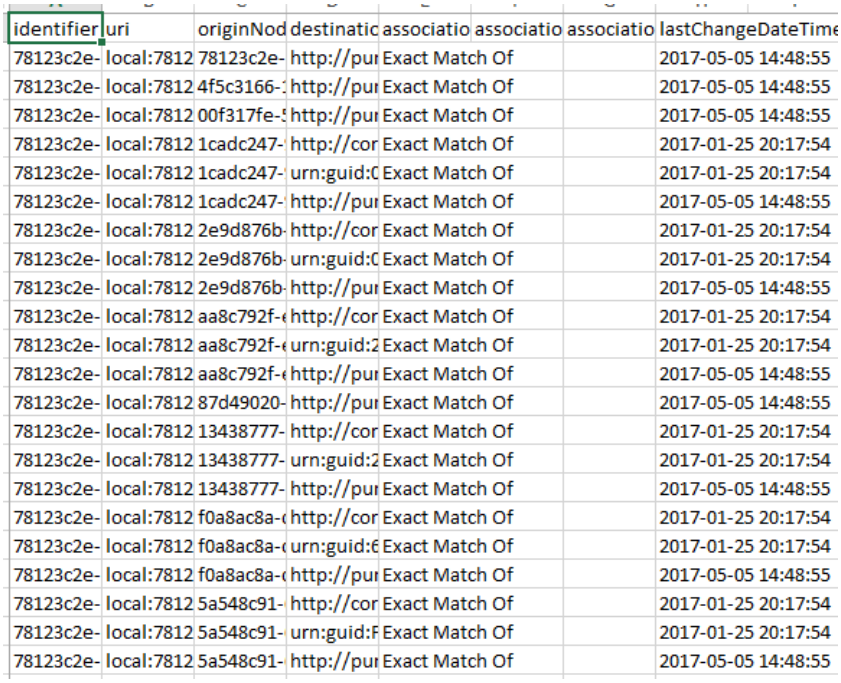
- *Identifier:* The identifier is intended to be used as the primary key global identifier within or external to the system. This is a mandatory field in OpenSALT and the export will contain data.
- *FullStatement*: The the main content of the CF Item. It is used to express both nodes and granular statements. If the statement is part of a list, the list enumeration should not be included in the statement and should instead be contained in the List Enumeration in Source Document. This is a mandatory field in OpenSALT and the export will contain data.
- *HumanCodingScheme*: The ID sometimes used by humans to identify a CF Item. It often will use concatenated codes expressing its position in the taxonomy and abbreviations to convey other classification information (e.g. K.CC.1.1). This is an optional field in OpenSALT. If best practices are not followed, this field may be blank.
- *SmartLevel:* This is an optional field in OpenSALT. If best practices are not followed, this field may be blank.
- *ListEnumeration:* Used to parse out enumerations or bullets that precede CF Item statements. This is an optional field in OpenSALT. If best practices are not followed, this field may be blank.
- *AbbreviatedStatement:* Abbreviated or summary statement provided by the Publisher. This is an optional field in OpenSALT and may be blank.
- *ConceptKeywords:* Upper level CF Item node statements may be used to populate Concept Keywords of lower level nodes. upper The concepts data structure allows a master list of keywords to be defined which can then be parsed down specific to a node. This works as usually concepts will be a less granular hierarchy representation of of the more detailed nodes in CF items. A node could be 'Geometry' and the lower node is 'Tangents' but the keywords for 'tangents' could include the word geometry. This is an optional field in OpenSALT and may be blank.
- *Notes:* In some cases, this can be used to contain additional information found in the original source document. This is an optional field in OpenSALT and may be blank.
- *Language:* HTML Language Country Code VIA- country code from . This is an optional field in OpenSALT. If best practices are not followed, this field may be blank. However OpenSALT assumes English if not other language value is entered.
- *EducationLevel:* The current US K12 defined vocabulary is to use CEDS https://ceds.ed.gov/

cedselementdetaiCF.aspx?termid=8267. Multiple values are allowed via comma delimitation and should be used to express grade spans. This is an optional field in OpenSALT and may be blank.

- *CFItemType:* e.g., "Standard," "Benchmark," "Strand," or "Topic." or "Level 1, Level 2,..." This is an optional field in OpenSALT and may be blank.
- *License:* Systems may filter for content with particular licences to support discovery. This is an optional field in OpenSALT and may be blank.
- *LastChangeDateTime:* This is used for versioning. This is a mandatory field in OpenSALT and the export will contain data.

6.7.2.3 CF Association Tab

The purpose of the third tab, CF Association, is to provide the user with items and content within the Framework Package.



identifier	uri	originNod	destinatic	associatio	associatio	associatio	lastChangeDateTime
78123c2e-	local:7812	78123c2e-	http://pur	Exact Match Of			2017-05-05 14:48:55
78123c2e-	local:7812	4f5c3166-	http://pur	Exact Match Of			2017-05-05 14:48:55
78123c2e-	local:7812	00f317fe-	http://pur	Exact Match Of			2017-05-05 14:48:55
78123c2e-	local:7812	1cadc247-	http://cor	Exact Match Of			2017-01-25 20:17:54
78123c2e-	local:7812	1cadc247-	urn:guid:0	Exact Match Of			2017-01-25 20:17:54
78123c2e-	local:7812	1cadc247-	http://pur	Exact Match Of			2017-05-05 14:48:55
78123c2e-	local:7812	2e9d876b-	http://cor	Exact Match Of			2017-01-25 20:17:54
78123c2e-	local:7812	2e9d876b-	urn:guid:0	Exact Match Of			2017-01-25 20:17:54
78123c2e-	local:7812	2e9d876b-	http://pur	Exact Match Of			2017-05-05 14:48:55
78123c2e-	local:7812	aa8c792f-	http://cor	Exact Match Of			2017-01-25 20:17:54
78123c2e-	local:7812	aa8c792f-	urn:guid:2	Exact Match Of			2017-01-25 20:17:54
78123c2e-	local:7812	aa8c792f-	http://pur	Exact Match Of			2017-05-05 14:48:55
78123c2e-	local:7812	87d49020-	http://pur	Exact Match Of			2017-05-05 14:48:55
78123c2e-	local:7812	13438777-	http://cor	Exact Match Of			2017-01-25 20:17:54
78123c2e-	local:7812	13438777-	urn:guid:2	Exact Match Of			2017-01-25 20:17:54
78123c2e-	local:7812	13438777-	http://pur	Exact Match Of			2017-05-05 14:48:55
78123c2e-	local:7812	f0a8ac8a-	http://cor	Exact Match Of			2017-01-25 20:17:54
78123c2e-	local:7812	f0a8ac8a-	urn:guid:6	Exact Match Of			2017-01-25 20:17:54
78123c2e-	local:7812	f0a8ac8a-	http://pur	Exact Match Of			2017-05-05 14:48:55
78123c2e-	local:7812	5a548c91-	http://cor	Exact Match Of			2017-01-25 20:17:54
78123c2e-	local:7812	5a548c91-	urn:guid:F	Exact Match Of			2017-01-25 20:17:54
78123c2e-	local:7812	5a548c91-	http://pur	Exact Match Of			2017-05-05 14:48:55

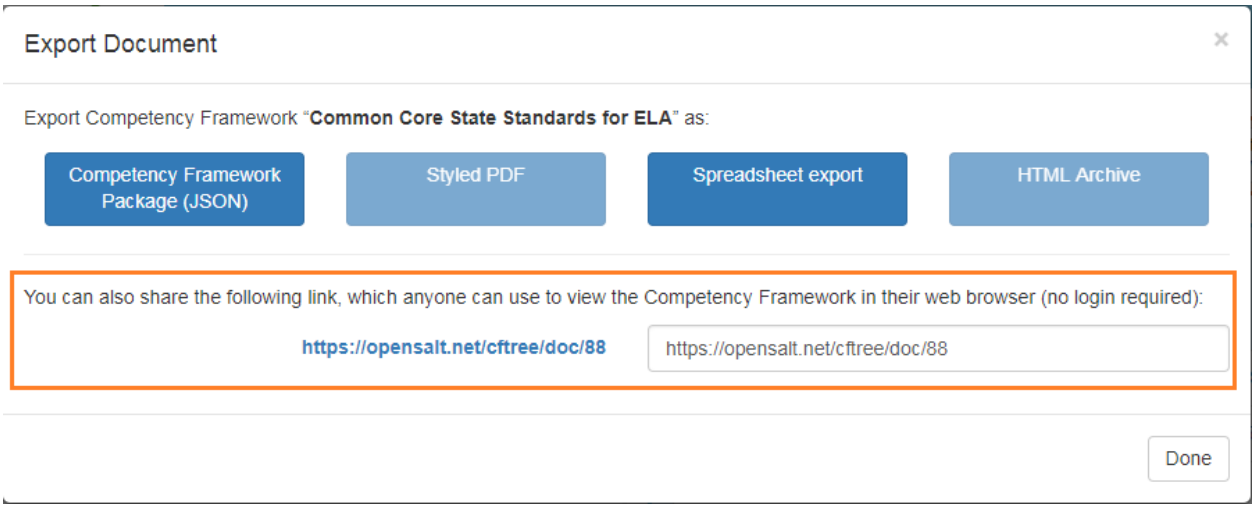
The columns on the spreadsheet include the following:

- *Identifier:* The identifier is an unambiguous, synthetic, unique reference to the association. This is a mandatory field in OpenSALT and the export will contain data.
- *URI:* Establishes uniqueness of an association between a learning standard and another learning standard or other objects such as learning resources. [CEDS Element: Learning Standard Item Association Identifier URI, ID:000871]. This is a mandatory field in OpenSALT and the export will contain data.
- *OriginNodeIdentifier:* Identifier of the origin node when the Learning Standard Item Association is used as a connector in a learning map. [CEDS Element: Learning Standard Item Association Origin Node URI, ID: 001406]. This is a mandatory field in OpenSALT and the export will contain data.
- *DestinationNodeIdentifier:* Identifier of the destination node when the CF Association is used as a connector in a learning map. [CEDS Element: Learning Standard Item Association Destination Node URI, ID: 001404]. This is a mandatory field in OpenSALT and the export will contain data.

- *AssociationType:* A controlled vocabulary used to express the types of associations used to describe the relationship between CF Docs and between CF Items. This is a mandatory field in OpenSALT and the export will contain data.
- *AssociationGroupIdentifier:* An identifier to allow associations to be grouped together. Different values only have to be unique within the document. This is an optional field in OpenSALT and may be blank.
- *AssociationGroupName:* A common group name to allow associations to be grouped together. Different values only have to be unique within the document. This is an optional field in OpenSALT and may be blank.
- *LastChangeDateTime:* A system generated log of the most recent change to this record. This is a mandatory field in OpenSALT and the export will contain data.

6.7.3 6.7.3 Link for Browser View

The final share option to connect to the Framework Package outside of OpenSALT is the **Browser Link**.



OpenSALT provides the users with the direct **URL** for the selected Framework Package as both a **hyperlink** and display the text of the unique **URL**. The sure can copy the **URL** to us in any document, application or website. The **URL** will return the audience to the selected **Framework's Display** page within OpenSALT when followed. Any user will be able to access the selected Framework package as no credentials or login are required when following the URL link.

CHAPTER 7

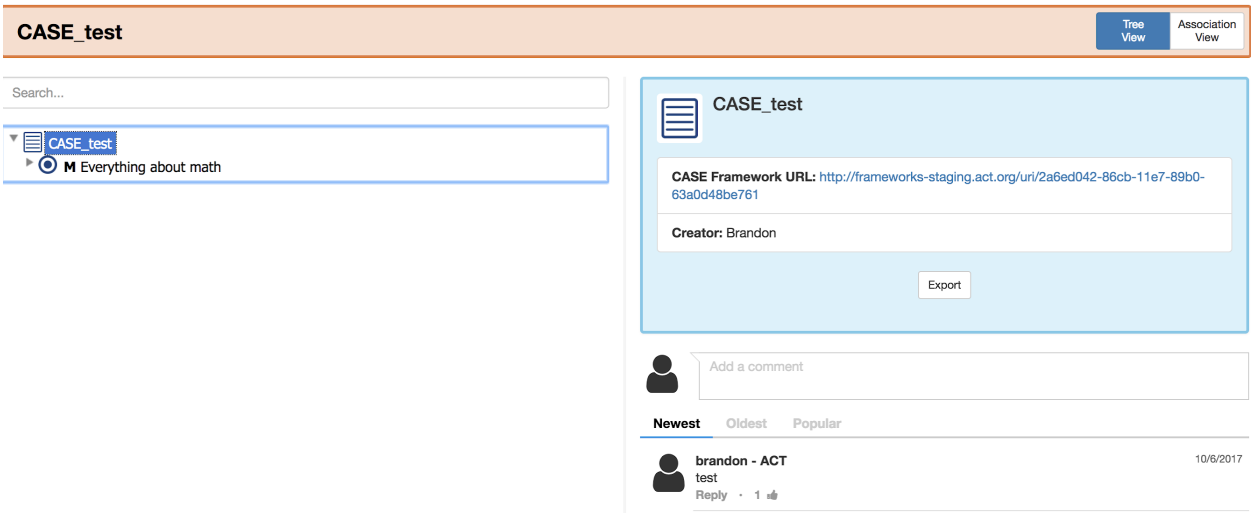
7.0 Options

7.1 7.1 Commenting Module

Published frameworks are able to be commented on by authenticated users for the purpose of collecting feedback on framework item metadata. Commenting must be turned on by a system administrator.

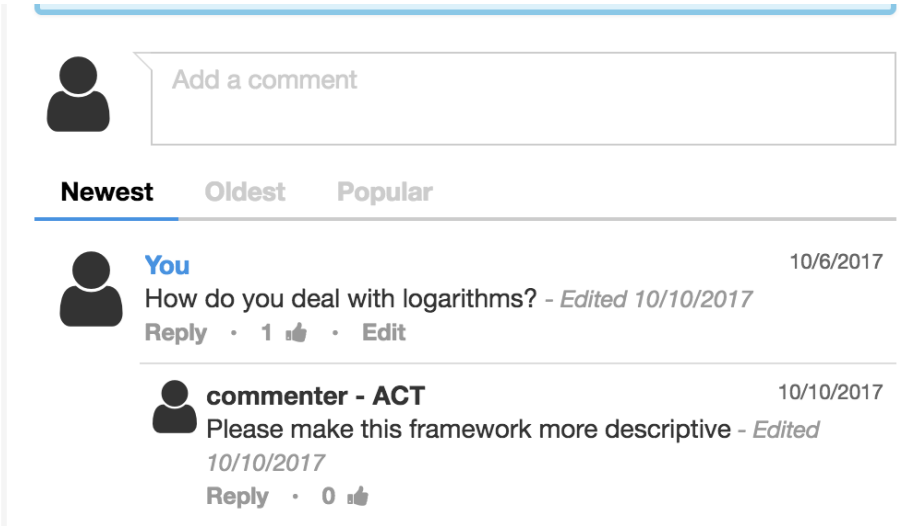
7.1.1 7.1.1 Features

Provisioned users are able to upvote and reply to other users comments. You are able to delete comments that you have written as well. Simply navigate to any framework item and make your comment, or upvote/respond to others previous comments. The comment panel may be hidden - it will be on the bottom of the ride side panel.



7.1.2 7.1.2 Viewing Comments

You are able to sort comments by the three tabs - newest, Oldest, and Most Popular (see graphic above). You may use these tabs to sort comments in the way that suits your interest and purpose best. Comments are not a part of the framework itself but independently stored. Due to a feature that truncates the email address of a participant, users cannot start their usernames with the @ symbol.



7.1.3 7.1.3 Commenting Configuration

Configuration for commenting can be found via the Github Config files for administrators. Administrators can configure commenting to include the ability for attachments as well as upvoting, replies etc.

7.1.4 7.1.4 License

OpenSALT has integrated this third party code (http://viima.github.io/jquery-comments/)with a congruent MIT license in order for reviewing frameworks.

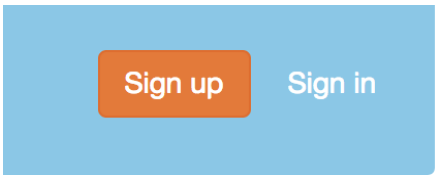
7.2 7.2 User Account Self-Creation

7.2.1 7.2.1 Features

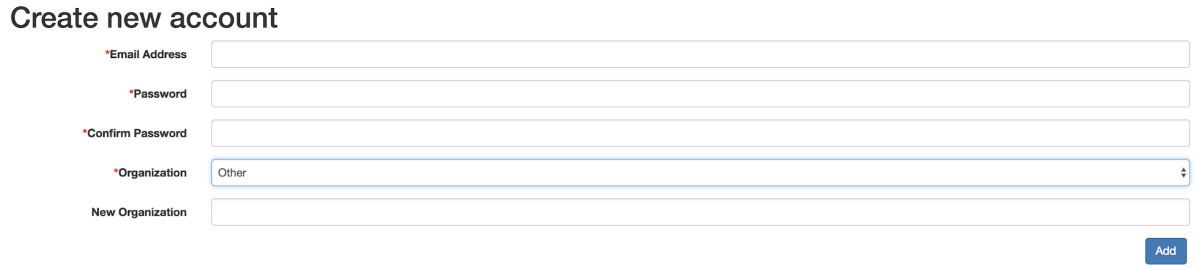
Users have the ability to create their own accounts and organizations on signup.

7.2.2 7.2.2 Process

1. Select “Sign up” from the top right page or Log In page.



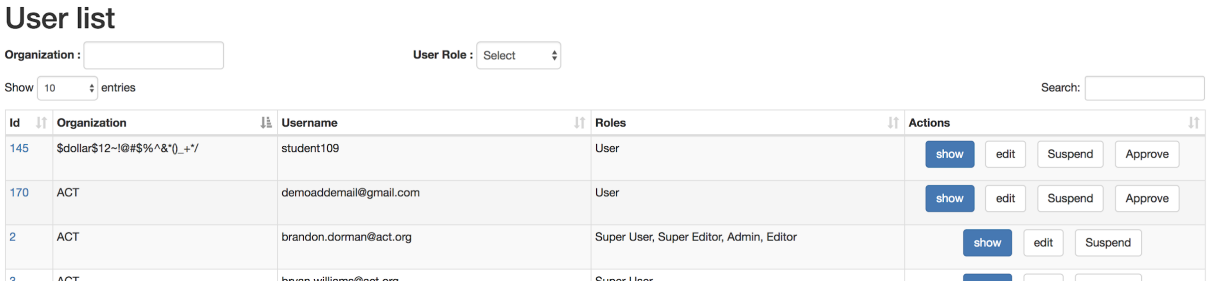
1. User enters their desired email address and password (twice). They cannot enter an email address already in use and their passwords must match.



1. They also can select their Organization and if no Organization exists, create one.

Upon submitting this information a short email is sent notifying the user of a pending account status.

4. Once the Super Admin receives an email notification, they can navigate to the User List page and Approve or Suspend/Reject the user.



Rejecting the users will simply suspend the accounts. Accepting the user will generate another confirmation email to the users notifying them of successful account creation,

7.2.3 7.2.3 Configuration

$. System Administrators can turn this feature on or off as well as email addresses/text in their parameters.yml file to “always-on” then restart docker. You must also enable the email server. This change triggers the Create Account

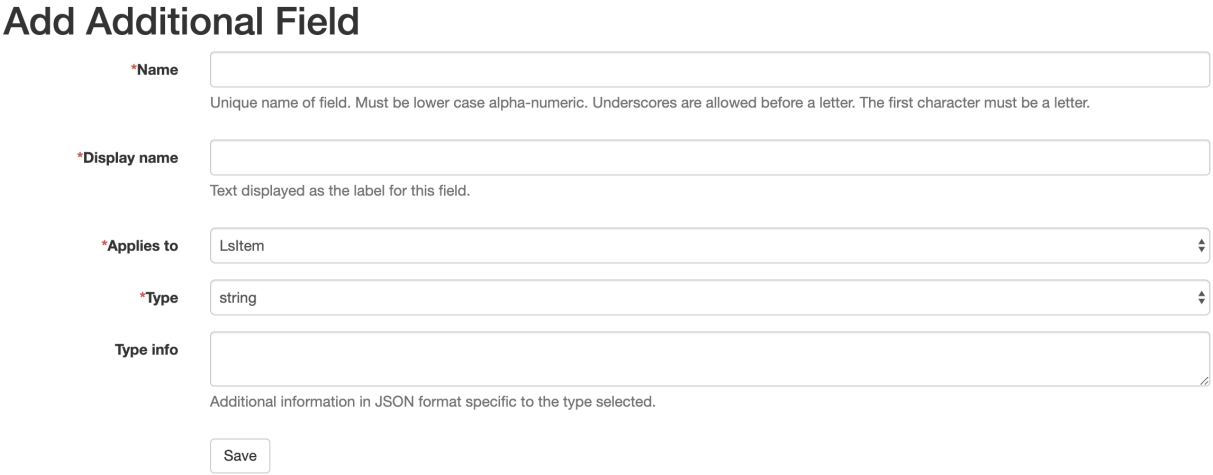
button as well as allows user access to the Create account page and shows Pending users on User List page. More information:

7.2.4 7.3 Additional Fields

OpenSALT 2.2 has the ability to add additional fields to the CfItem entity. These fields will NOT be exported with a standard CASE export or visible via the API. They will however be viewable when editing the Cf Item of any framework on that OpenSALT instance and be exported/able to be updated via the Spreadsheet Updater.

To view the additional fields, login and a super user and go to <server name>/additional_field eg frameworks.act.org/additional_field

To create a new field, select New and create your field.



The video below shows this as well as offering best practices guidance. **|LINK29|**

Additional Fields Feature Video

7.2.5 7.4 File Storage and Image Attachment

Often users want to attach an image to the notes section for clarity, or perhaps fullStatement can point to a learning object etc. The ability to add uploads and images was created for this type of functionality.

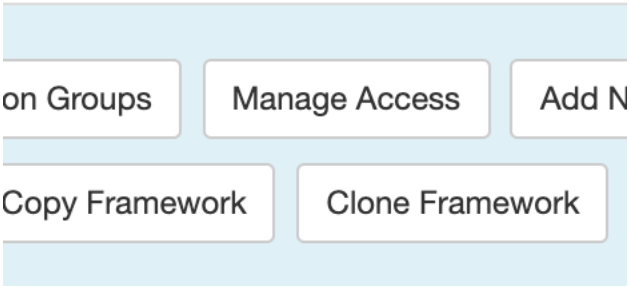
7.4.1 Configuration

While files cannot be stored natively in the CASE JSON, OpenSALT allows administrators setting up OpenSALT to include an AWS bucket as well as a prefix in the docker-compose.yml file - then the images are inserted with the native markdown/attachments are linked. Make sure your bucket is public if you intend for the CASE framework to be shared and disseminated publicly.

AWS_ACCESS_KEY_ID: AWS_SECRET_ACCESS_KEY: AWS_S3_BUCKET: AWS_S3_REGION: AWS_S3_PREFIX: ATTACHMENT_URL_PREFIX: BUCKET_PROVIDER:

7.4.2 Usage

Users may drag and drop and image into the fullStatement and Notes field when this functionality is enabled. Future plans would add it to Exemplar as well but that is not on the OpenSALT roadmap at this time.

7.2.6 7.5 Clone Framework

New in OpenSALT 2.3 is the ability to clone a framework directly. This is used in best practice when template frameworks are needed. The clone button shows up for editors with proper permissions