madcap software

USER GUIDE

MADCAP CENTRAL

Projects and Builds Guide

MadCap Software
9171 Towne Center Drive, Suite 335
San Diego, California 92122
858-320-0387
www.madcapsoftware.com

THIS PDF WAS CREATED USING MADCAP FLARE.

CONTENTS

CHAPTER 4

CHAPTER 5

CHAPTER 6

CHAPTER 7

CHAPTER 8

CHAPTER 9

CHAPTER 10

CHAPTER 11

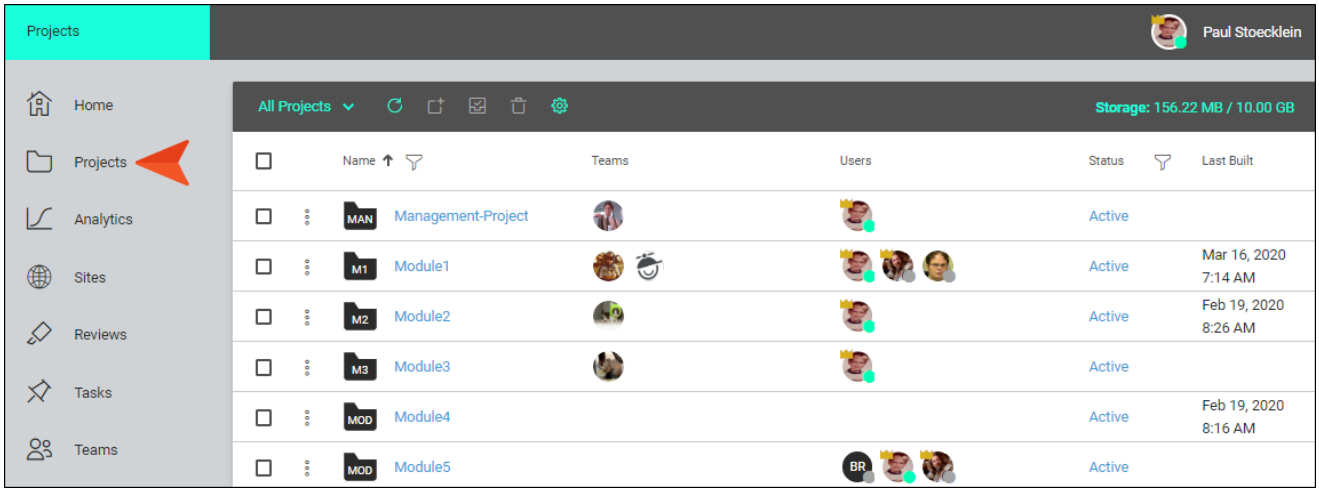
CHAPTER 12

APPENDIX

CHAPTER 1

Introduction

The Projects page displays a grid, which lists all MadCap Flare projects that have been uploaded to your Central license. These are copies of the local projects located on users' desktops. There are many good reasons to upload (i.e., bind) Flare projects to Central, including (but not limited to) the following: (1) You can let Central build and host your output, which means you can publish without having to involve an IT department; (2) the connection between your local Flare project files and the cloned files on Central can be used as a source control solution, with branching integrated in various places in Central; (3) you can use Central as a platform for topic reviews; (4) you can add and edit project files directly in Central, with the ability to tap into the power of ChatGPT; and (5) you can manage progress on topic development by using checklists.



On a smaller screen, the options on the left are opened from a flyout menu.

Main Activities

Other Activities

Page Views

Reviews and Analytics

- "Reviews" on page 214
- "Analytics" on page 295

CHAPTER 2

Main Activities for Projects

Some activities are particularly common and important when it comes to this feature.

This chapter discusses the following:

Uploading (Binding) Projects

You can upload a project to Central from the desktop Flare application. A copy of the project files will therefore reside in the cloud via Central, and you will be able to keep the local and cloud versions of the project synchronized using integrated source control. You will also be able to generate and publish Flare targets using Central.

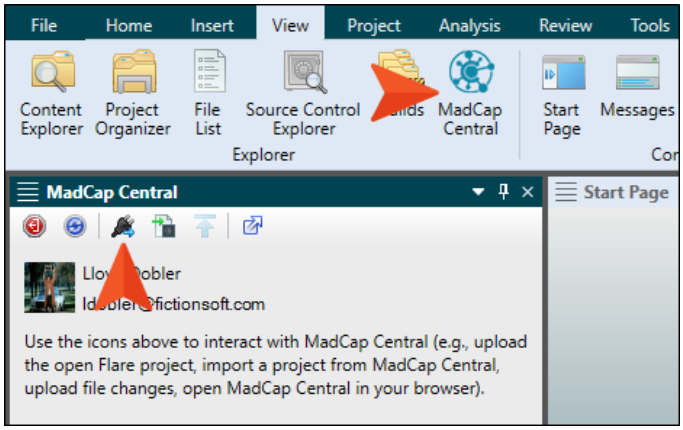
Permission Required?

For this activity, you must have the following permission setting:

For more information about permissions, see the Central online Help.

How to Upload (Bind) a Project to Central

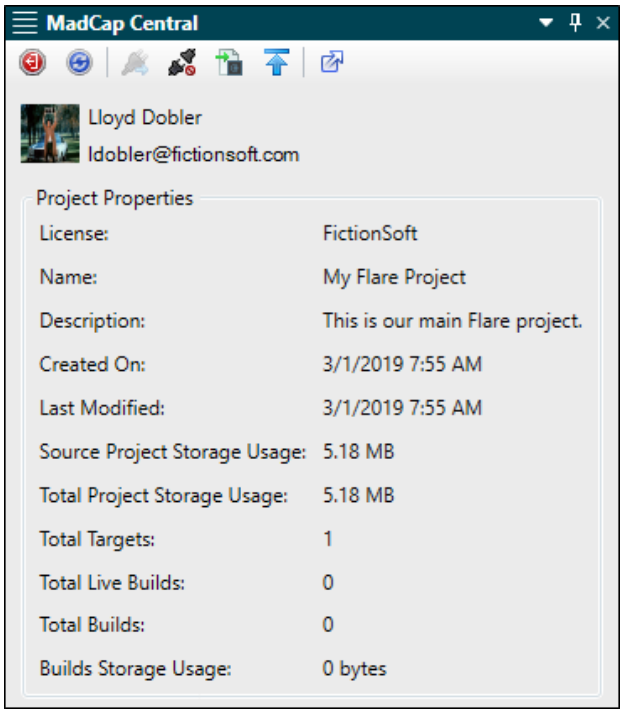
1. Open the project in MadCap Flare.
2. Select **View > MadCap Central**. The MadCap Central window pane opens.
3. Be sure to log in if you aren't already.
4. Click  .



> **TIP** If this button is disabled, try clicking in the window pane first.

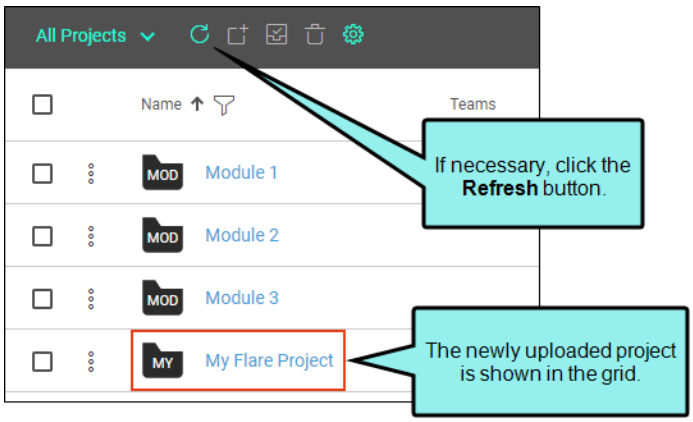
5. In the Bind Project dialog, you can complete these fields, then click **OK**.
   - **Enter project name** This shows the name of the project. You can change the name if you want, but in most cases you will probably want to keep the same name.
   - **Enter project description** You can enter a short description, which can then be seen when anyone views the project profile in Central.

6. In the success message, click **OK**. The MadCap Central window pane adjusts, showing properties for the current project.



7. (Optional) In the local toolbar, click to open Central.

8. (Optional) In Central, you can click **Projects** on the left side to open the Projects page. The uploaded project should be shown in the grid. If it isn't, click to refresh the page.



NOTE If your project is dual-bound, be sure you check in or push the project files to your third-party source control provider after you bind to Central. This will ensure that other users who import the project from source control have the most recent version of the project.

NOTE If you intend to build a target that links to an outside Flare project (e.g., for the purpose of importing files via Global Project Linking, project merging, or multilingual output), make sure that the other project is also uploaded to Central. Otherwise, the automated imports or merging will not work. See "Setting Up Project Linking" on page 68.

NOTE Due to security issues, MadCap Software does not support the use of the Secure Shell (SSH) protocol in MadCap Flare when connecting to MadCap Central. The Hypertext Transfer Protocol Secure (HTTPS) protocol is supported.

Importing Projects

If another user uploads a Flare project to your MadCap Central license and you do not yet have that project on your computer, you can import it. You can do this from the MadCap Central window pane in Flare.

In order to import a project, you must be associated with that project in Central. For more information, see the online Help.

NOTE If your project is already bound to a third-party source control provider other than Git—i.e., you are working in a dual-bound model as opposed to a single-bound model—the Import option in the MadCap Central window pane will be disabled. You will have to import your project directly from source control using your non-Central binding.

If you already have the project on your machine, you do not need to reimport the project from source control. Instead, you can get the latest version from source control by using the Flare Source Control ribbon (see the Flare online Help).

Be sure that you have the latest version of the project.

IMPORTANT If you are using a Git-Central dual-binding, you can import projects from Git (using the Import Project From Source Control Wizard) or from Central. The project will be the same. However, if you import from Central, you will only be able to push changes to and pull changes from Central. If you import from Git(see the Flare online Help), you will be able to push changes to and pull changes from Central *as well as* your main Git repository. This is because the project in Central has no connection to the original Git repository, and if you import from Central you will not have those source control bindings.

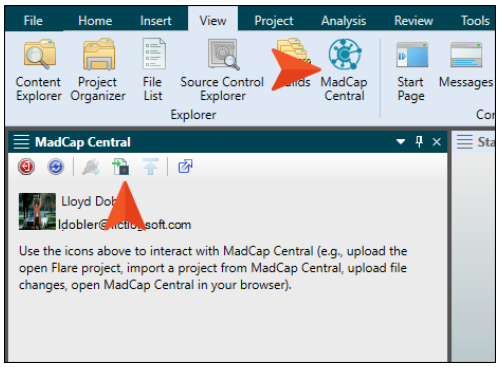
Permission Required?

For this activity, you must have the following permission setting:

For more information about permissions, see the Central online Help.

How to Import a Project From Central

1. Select **View > MadCap Central**. The MadCap Central window pane opens.
2. Be sure to log in if you aren't already.
3. Click  .



> **TIP** If this button is disabled, try clicking in the window pane first.

The Import Project dialog opens.

4. From the **My Projects** field, select the project you want to import.
5. Next to the **Destination folder** field, click ... and choose the folder where you want the imported project to be stored on your computer.
6. Click **OK**.
7. When the import is finished, you can click **Open** to launch the project in Flare. Otherwise, you can click **Cancel**.

Opening a Project

If you are associated with a project, you can open it to see its dashboard and navigate to other views.

How to Open a Project

1. On the left side of the Central interface, click **Projects**.
2. In the grid, click the name of the project. The project opens in the Dashboard view.
3. By clicking the tabs at the top of the screen, you can move between the other project views—Builds, Checklists, Files, Commits, and Reports.

> NOTE A user needs to be associated with a project in order to open it. See the Central online Help.

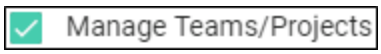
Associating Projects With Users and Teams

Administrators can associate users with teams and projects in the system.

- Associating users with teams is a convenient way to set the same permissions for a group of people. It also makes it easy to send messages to those who are designated as authors on the license. Also, if you intend to have private output (i.e., output where users must log in), it is necessary to associate those users (authors, SMEs, or viewers) with a team, and then associate that team with the appropriate site.
- Associating users with projects gives those users access to specific projects. Without access to a project, a user—even an administrator—cannot open that project to view it or work in it. You can only associate authors with projects; you cannot associate SMEs or viewers with projects.

Permission Required?

For this activity, you must have the following permission setting:



For more information about permissions, see the Central online Help.

How to Associate Projects With Users and Teams

The following steps show how to make associations in the project profile. However, you can also open the user profile or team profile and do the same.

1. Make sure you have already created the team(s) and invited the user(s).
2. On the left side of the interface, click **Projects**.
3. Click the project's avatar.
4. On the left, click **Access**.
5. Click **Edit**.
6. In the **Teams** section, you can associate teams with the project.
7. In the **Users** section, you can associate authors with the project.
8. Click **Save**.

NOTE Associations between users, projects, and teams are treated separately. Therefore, if one connection is severed, another one remains.

For example, let's say you open a team and associate it with a user and a project. This means that both the user and project are associated with that team, but the user is not automatically associated with the project. You need to manually associate the user with the project. This is an extra step, but if you then remove the user from the team, that person is still connected with the project.

CHAPTER 3

Other Activities for Projects

In addition to the main activities, there are some other tasks you might perform regarding this feature.

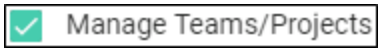
This chapter discusses the following:

Editing Project Profiles

You can edit the profile for a project. This includes the ability to change the project's name, initials, description, and associated color. You can also change which teams and/or users are associated with the project.

Permission Required?

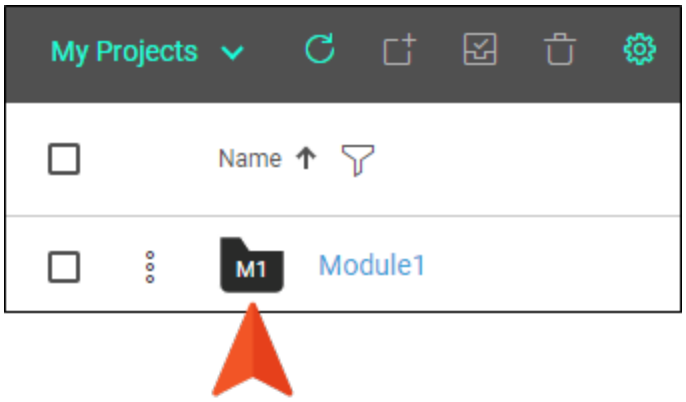
For this activity, you must have the following permission setting:



For more information about permissions, see the Central online Help.

How to Edit a Project Profile

1. On the left side of the Central interface, click **Projects**.
2. Click the project avatar.



3. The Overview page provides an easy-to-read summary of the project's details. You can select options on the left side to make changes.
   - **Settings** Change the project's icon color, description, initials for the avatar, and name.
   - **Access** View and change the teams and users associated with the project.

     > **NOTE** Associations between users, projects, and teams are treated separately. Therefore, if one connection is severed, another one remains.
     >
     > For example, let's say you open a team and associate it with a user and a project. This means that both the user and project are associated with that team, but the user is not automatically associated with the project. You need to manually associate the user with the project. This is an extra step, but if you then remove the user from the team, that person is still connected with the project.

   - **Activity** View the most recent actions that have taken place for the project.
   - **Delete** Remove the project from Central.
   - **Dashboard** Open the Dashboard view for the project.
   - **Builds** Open the Builds view for the project.

- **Checklists** Open the Checklists view for the project.
- **Files** Open the Files view for the project.
- **Commits** Open the Commits view for the project.
- **Reports** Open the Reports view for the project.

4. Click **Save**.

Viewing Project Profiles

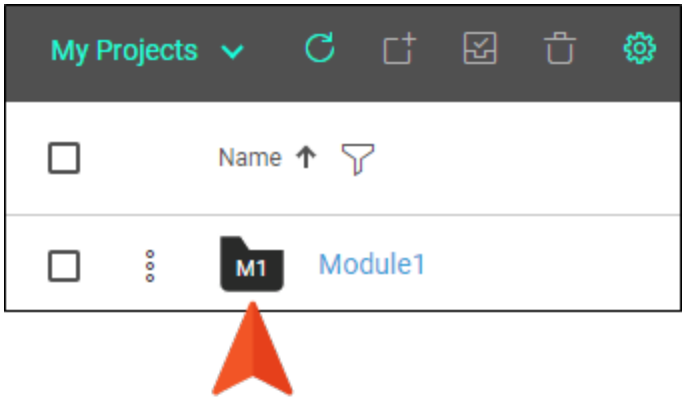
You can view the profile for a project to see its description, the teams and/or users associated with it, and recent activity. Depending on your permission settings, you can also edit it, delete it, or open one of the following views: Dashboard, Builds, Checklists, Files, Commits, Reports.

Permission Required?

No special permission is required for this activity. All authors are allowed.

How to View a Project Profile

1. On the left side of the Central interface, click **Projects**.
2. Click the project icon.



3. The Overview page provides an easy-to-read summary of the project's details. You can select options on the left side to see more information or make changes. See "Editing Project Profiles" on page 21.

Viewing Project Activity

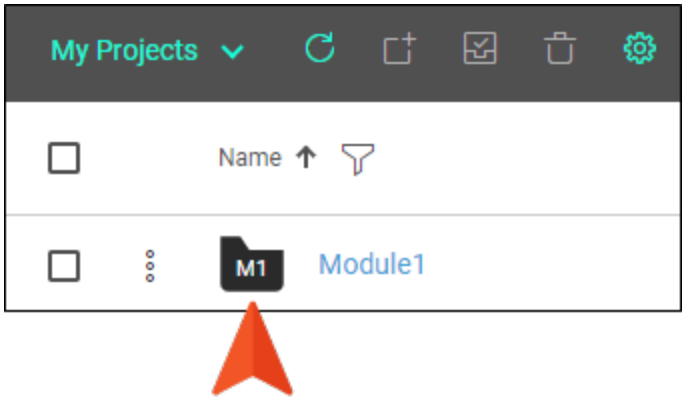
You can view the activity for a project to see what has been going on with it.

Permission Required?

No special permission is required for this activity. All authors who are associated with the project are allowed.

How to View Project Activity

1. On the left side of the Central interface, click **Projects**.
2. Click the project icon.



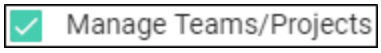
3. In the dialog, click **Activity**. The project activities over time are shown to the right.
4. (Optional) You can click any blue links or icons to view the related information. After scrolling down to the bottom, you can also click **Load More** if you want to see more entries.

Changing the Status of a Project

When you upload a project to Central, it has a status of “Active.” If necessary, you can change the status to “Archive” or “Lock.” You can always change it back to “Active” later if you need to.

Permission Required?

For this activity, you must have the following permission setting:



For more information about permissions, see the Central online Help.

How to Change the Status of a Project

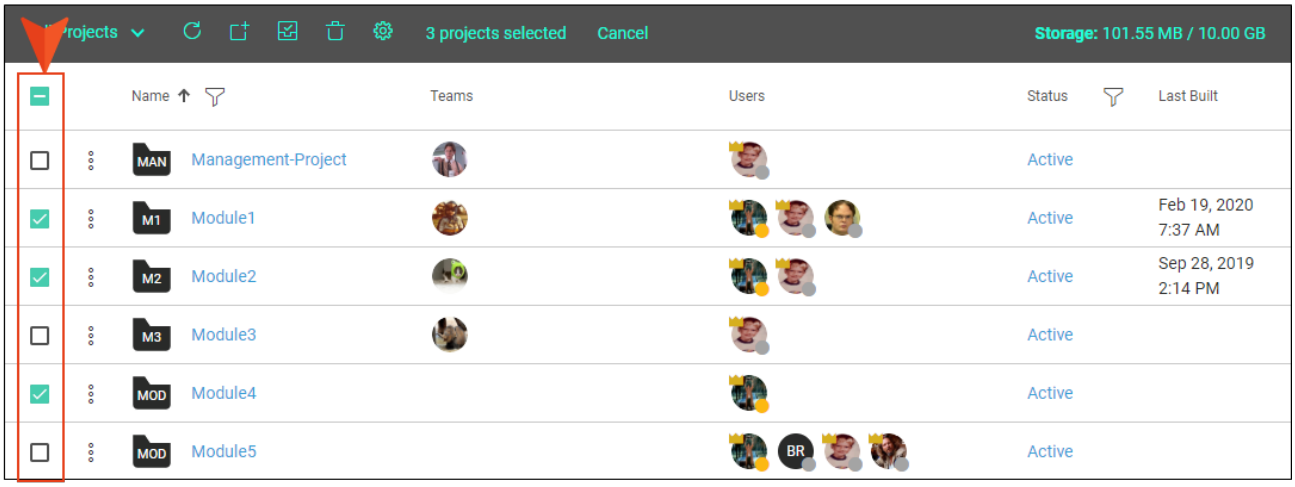
1. On the left side of the Central interface, click **Projects**.
2. In the **Status** column on the Projects page, click the current status of the project. (Alternatively, click the check box to the left of the project row. Then in the local toolbar click .)

   The Set Project Status dialog opens.
3. Click in the drop-down field and choose the status you want to set:
   - **Activate** If you have previously archived or locked a project, you can select this option to return it to an “Active” status.
   - **Archive** You might archive a project if you want to keep a copy of the files in the cloud, but you no longer need the output. An archived project cannot be opened or viewed, and all outputs become inaccessible. If you have a site set to “live,” you cannot archive the associated project unless you first remove the “live” setting.

     If you archive a project, all data is preserved. All access will be denied to the project until it is reactivated.
   - **Lock** You might lock a project if you do not want any more changes made to the project, but you need to retain the output, particularly any sites set as “live.” When a project is locked, it becomes read-only.
4. Click **Save**.

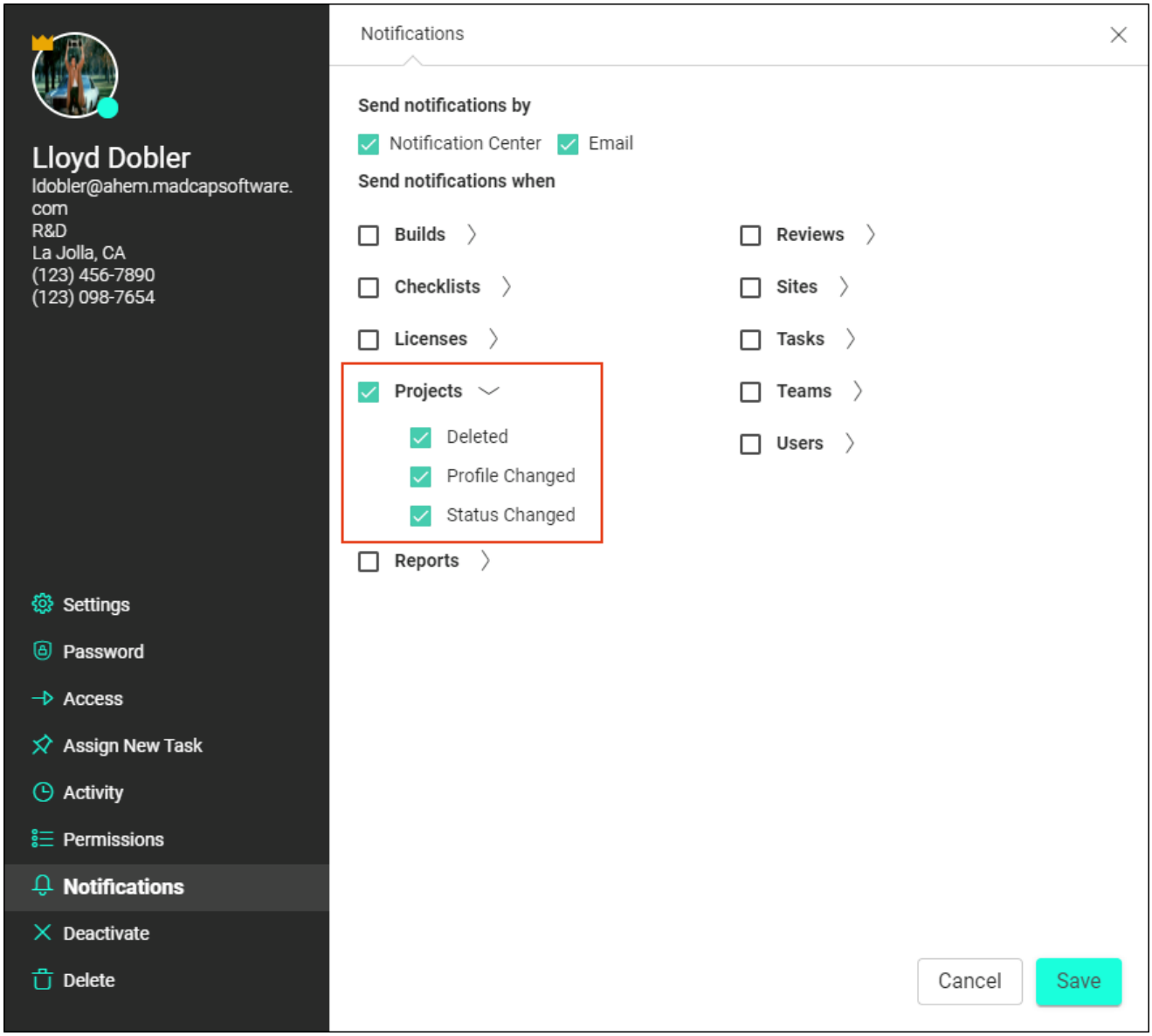
Selecting Multiple Projects

In the Projects page grid, you can use check boxes to select one or more projects. After doing this, you can perform the same action on those projects simultaneously (e.g., add widgets to their dashboards, set status, delete).



Setting Notifications for Projects

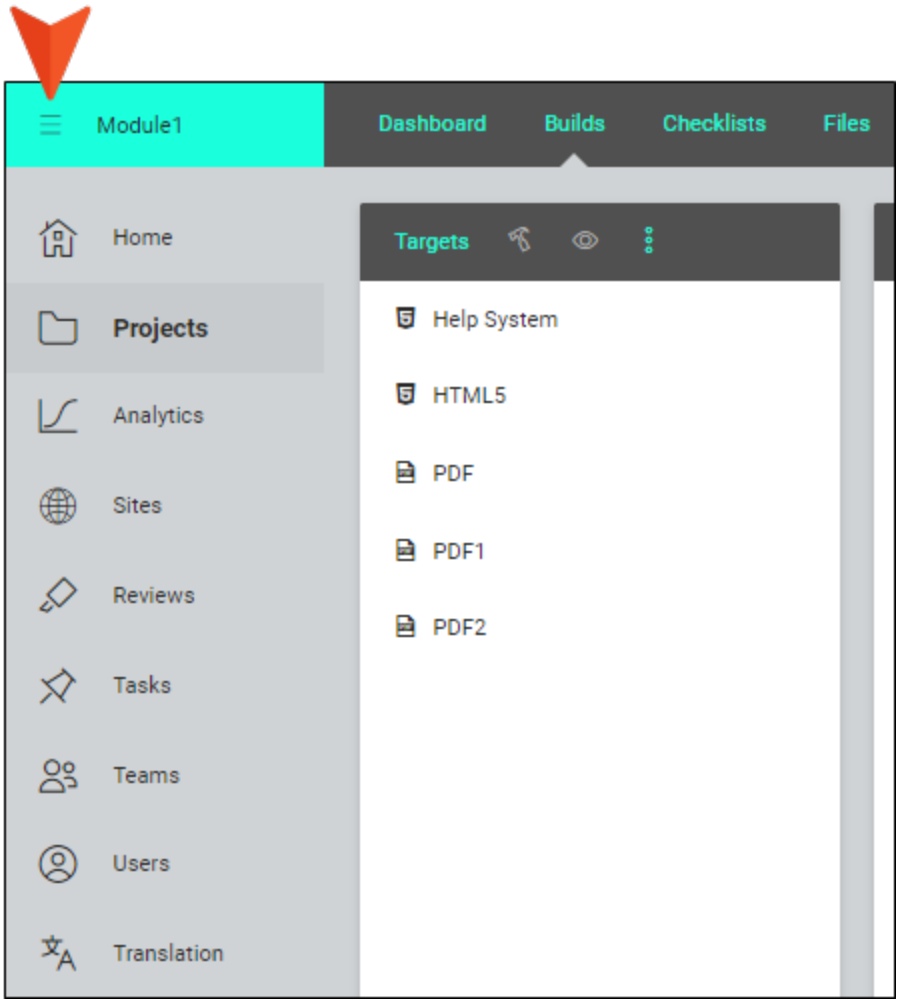
In your account settings, you can choose when to be notified for specific activities. The user who is notified depends on the activity. For more information, see the online Help.



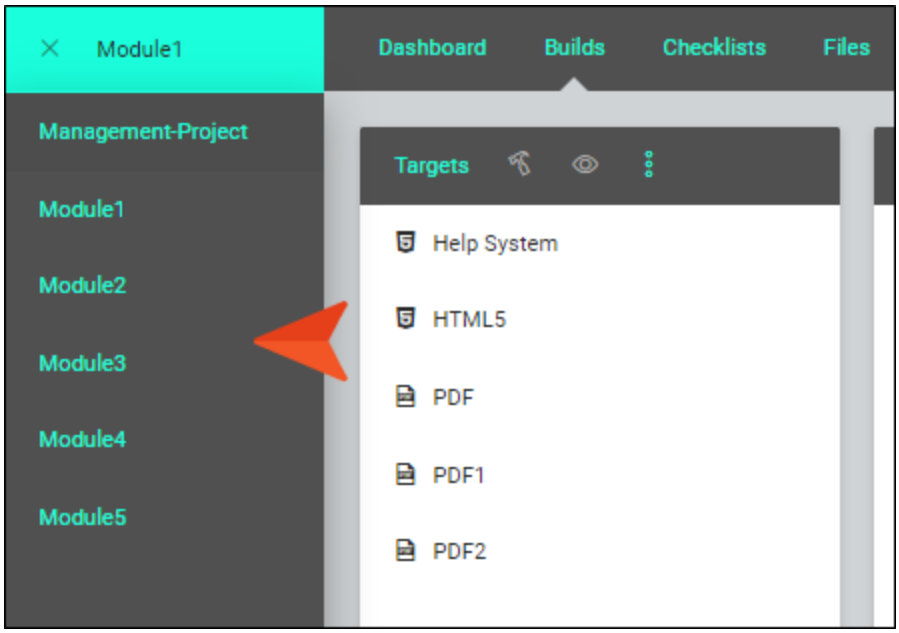
You can also set notifications for project builds, checklists, and reviews.

Accessing Other Projects

Once you've opened a project in Central, you can access a drop-down in the upper-left corner of the interface.



From this drop-down, you can quickly navigate to any other projects that you have permission to open. This is quicker than going back to the main Projects page grid to open another project.



Deleting Projects

If you no longer need to keep a project, you can remove it. Doing this does not delete your local copy of the Flare project; it only removes the project files from Central. All access will be denied to the project and all data associated with the project will be deleted. This frees up space on your license.

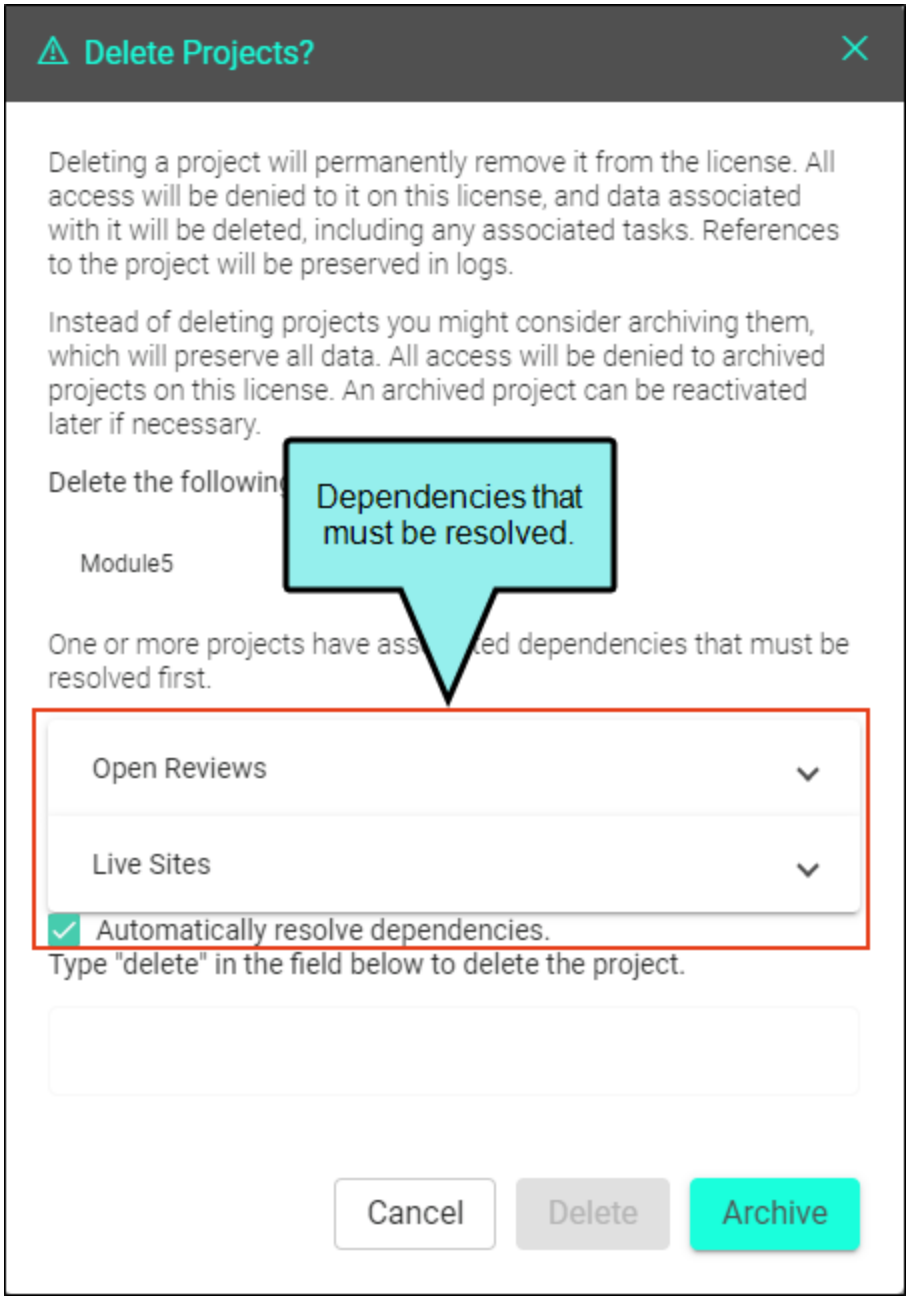
Permission Required?

For this activity, you must have the following permission setting:

For more information about permissions, see the Central online Help.

How to Delete a Project

1. On the left side of the Central interface, click **Projects**.
2. In the grid, click the check box to the left of the project row. Then in the local toolbar click .
3. If the project is associated with any live sites or pending reviews, a drop-down displaying the dependencies is shown in the dialog.



You can open the drop-down(s) to view the files or sites in question. Click **Automatically resolve dependencies** if you want Central to close the reviews and/or remove the live site associations. If you do not select this check box, you cannot continue with the deletion.

4. Click in the field at the bottom (under the least piece of text), and type `delete`. Then click **Delete**.

> **IMPORTANT** This extra action is necessary as a precaution, because once a project is deleted, it is permanently removed with no recovery.

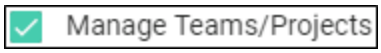
> **NOTE** As an alternative to deleting a project, you might consider archiving it instead. If you archive a project, all data is preserved. All access will be denied to the project until it is reactivated. See "Changing the Status of a Project" on page 26.

Renaming Projects

There are a couple of ways to rename a Flare project that is bound to Central. You can rename the project in Central only in order to modify how it is displayed in the interface. You can also change the name of the actual project file (FLPRJ) in both Central and Flare. You might even want to perform both sets of steps below so that the project name is the same everywhere.

Permission Required?

For this activity, you must have the following permission setting:

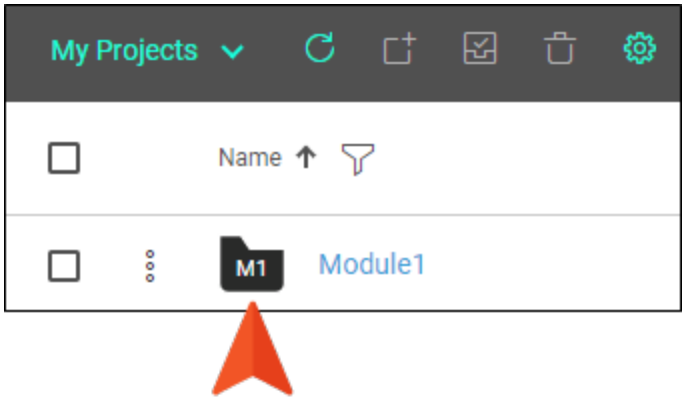


For more information about permissions, see the Central online Help.

How to Rename the Project in Central (Interface Display)

Changing the project name in Central is easy, but doing this will not also change the name of the actual Flare project. It only affects how the project name is displayed in various places in the Central interface; if you open the Files view for the project, you will see that the Flare FLPRJ file still has the original name.

1. On the left side of the Central interface, click **Projects**.
2. Click the project icon.

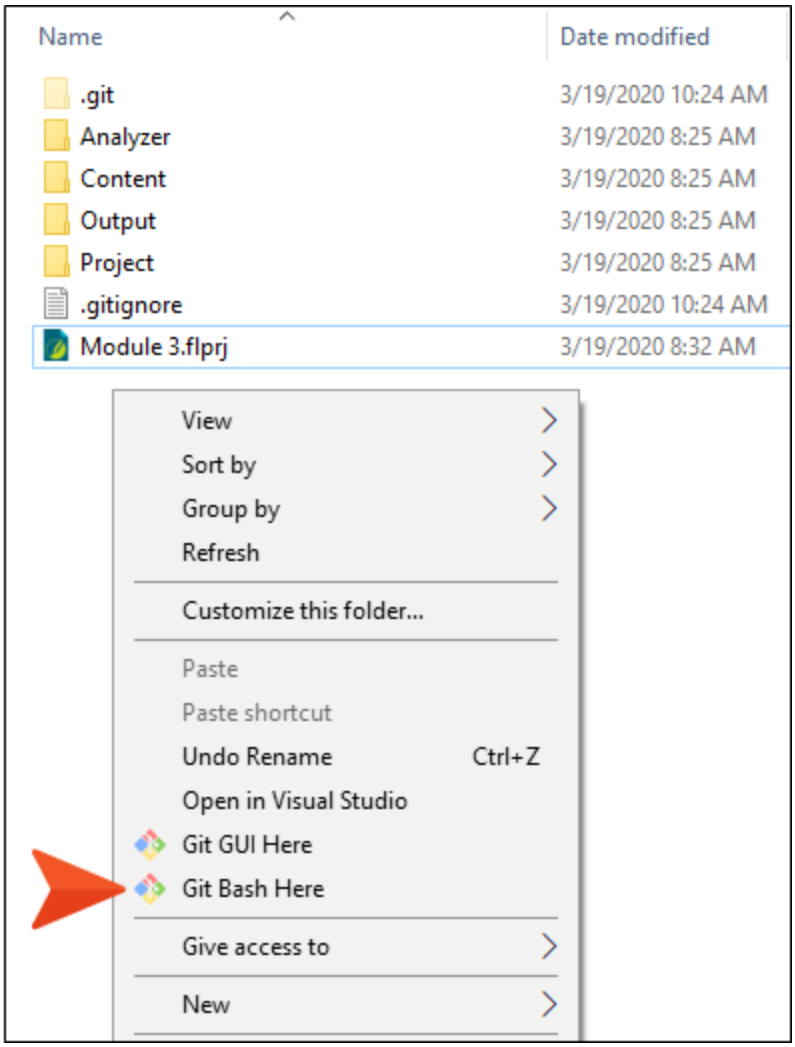


3. Select **Settings**.
4. In the **Name** field, change the name for the project.
5. Click **Save**.

How to Rename the Project in Central and Flare (File Name)

This requires you to perform steps outside of Flare and Central. You need to use Windows and Git Bash for synchronizing the new name between Flare and Central.

1. Close the Flare project, and open Windows to the location where the project is stored.
2. Rename the FLPRJ file.
3. Right-click in an empty area of Windows (not on a file or folder), and from the context menu select Git Bash Here.



4. In Git Bash type `git add -A` and press ENTER.

```
al Projects/Prod/Mod 3 (master)
$ git add -A
```

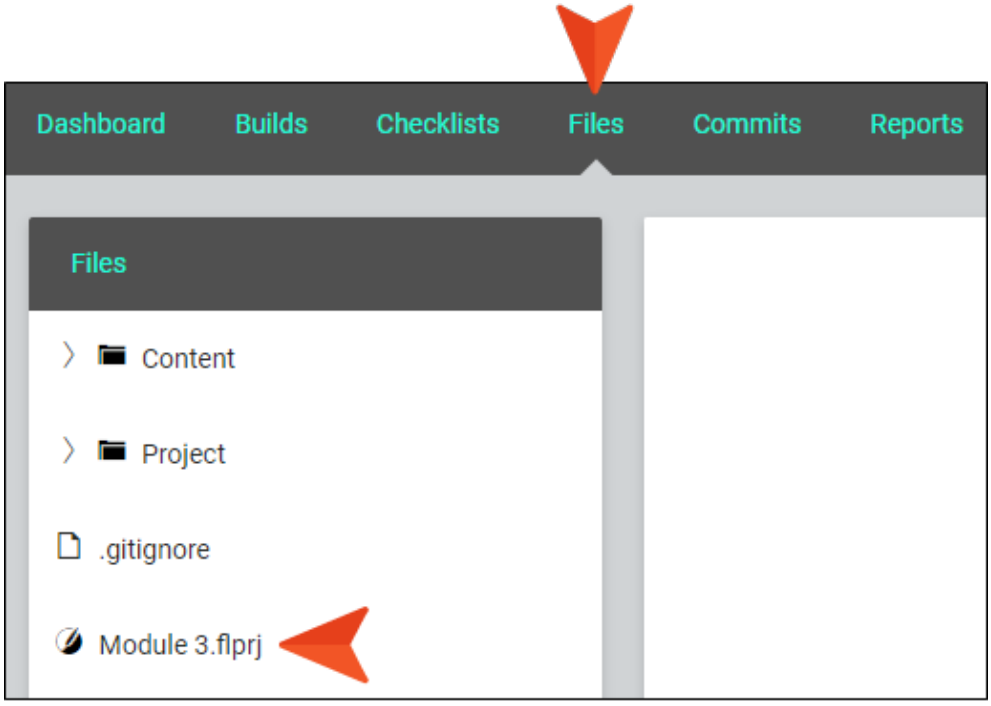
5. Type `git commit -m "My comment"` and press ENTER. (The text in quotes can be any comment you want to enter.)

```
al Projects/Prod/Mod 3 (master)
$ git commit -m "My comment"
```

6. Type `git push` and press ENTER.

```
al Projects/Prod/Mod 3 (master)
$ git push
```

In Central, you will notice on the Files page view for the project that the new name is shown.



CHAPTER 4

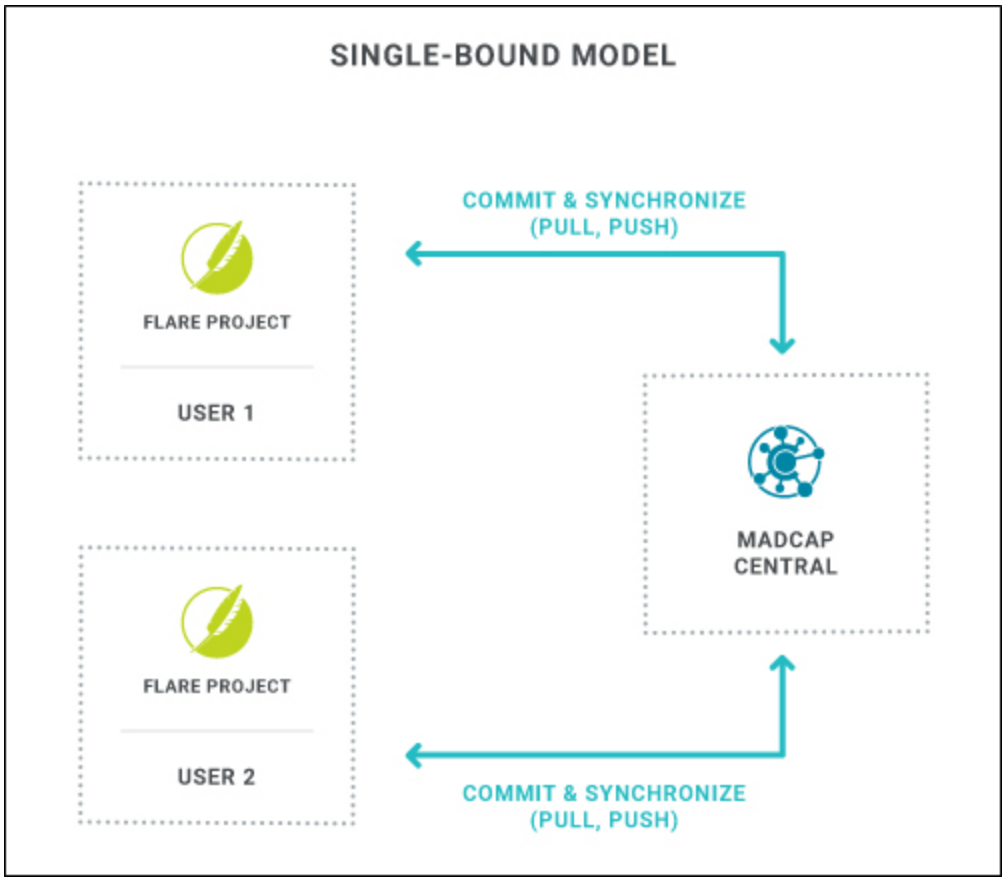
Source Control and MadCap Central

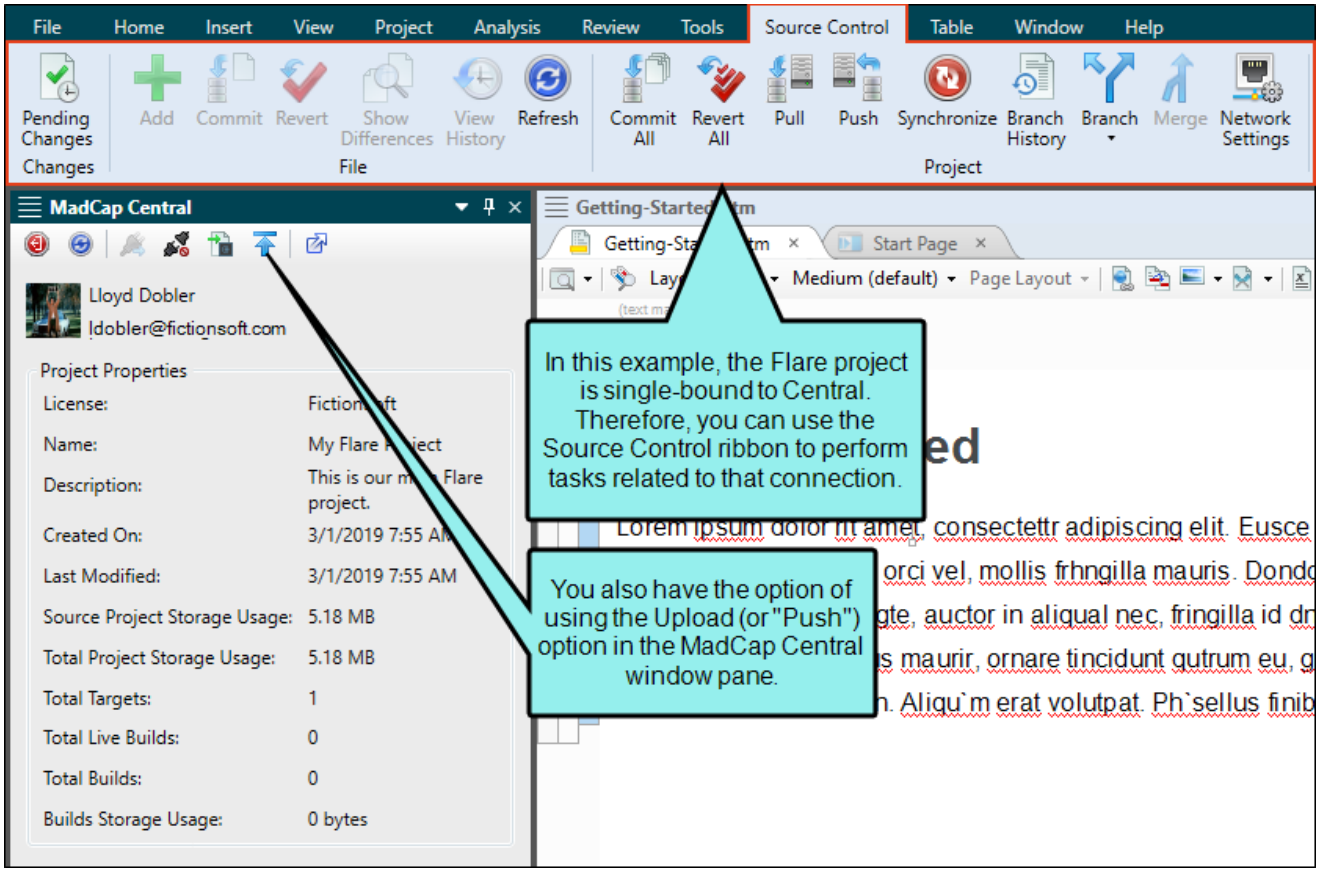
When you upload a Flare project to Central, the files are connected to Central via an integrated source control system (Git). Your interaction with source control can follow one of two models—single-bound (recommended) or dual-bound. Git is the source control solution that works behind the scenes to integrate Flare with Central.

This chapter discusses the following:

Single-Bound Projects

If your Flare project is not already bound to a third-party source control tool and you upload the project to Central, you will have only one source control binding. Therefore, you can use the source control connection between your local Flare project repository and the remote Central repository as your primary source control solution. With this model, you can use the Source Control ribbon in Flare to perform regular source control tasks such as committing and synchronizing (i.e., pulling and pushing) files. However, you also have the option of using tools outside of Flare (e.g., Git Bash) to perform source control tasks (e.g., in case there are certain source control tasks not supported in the Flare interface that you need to complete).




File
Home
Insert
View
Project
Analysis
Review
Tools
Source Control
Table
Window
Help
Pending Changes
Changes
Add
Commit
Revert
Show Differences
View History
Refresh
File
Commit All
Revert All
Pull
Push
Synchronize
Branch History
Branch
Merge
Network Settings
Project
MadCap Central
Lloyd Dobler
ldobler@fictionsoft.com
Project Properties
License:
Name:
Description:
Created On:
Last Modified: 3/1/2019 7:55 AM
Source Project Storage Usage: 5.18 MB
Total Project Storage Usage: 5.18 MB
Total Targets: 1
Total Live Builds: 0
Total Builds: 0
Builds Storage Usage: 0 bytes
Start Page
Medium (default)
Page Layout
In this example, the Flare project is single-bound to Central. Therefore, you can use the Source Control ribbon to perform tasks related to that connection.
You also have the option of using the Upload (or "Push") option in the MadCap Central window pane.

How to Use a Single Binding

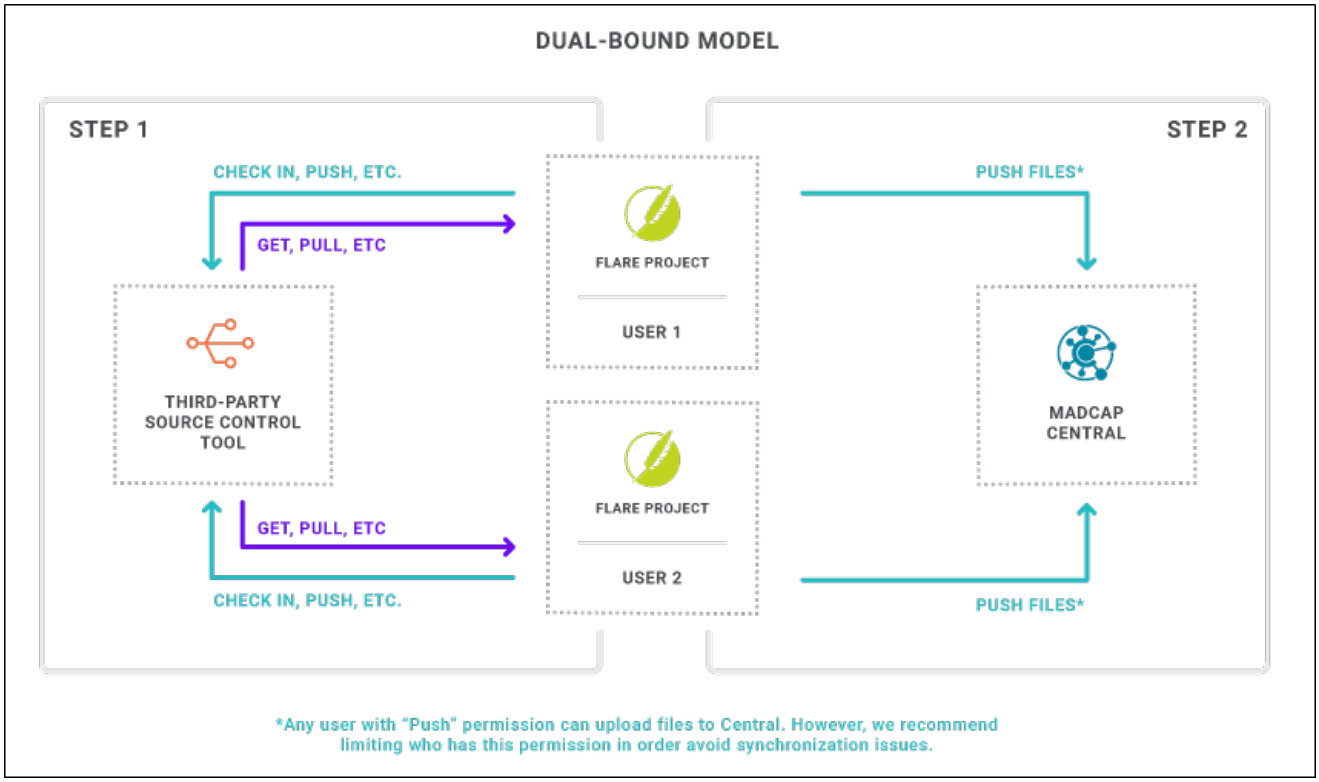
1. Open the project in MadCap Flare.
2. Select **View > MadCap Central**. The MadCap Central window pane opens.
3. Be sure to log in if you aren't already.
4. Upload (bind) your project to Central. See "Uploading (Binding) Projects" on page 11.
5. In Central, assign users to the new project and make sure they have “Import/Pull” and “Push” permissions. This enables users to import the project using the MadCap Central window pane. It also allows users to push changes up to Central. See the online Help.
6. Newly assigned users must now import the project from Central using the MadCap Central window pane in Flare. This allows each user to work on the project locally. See "Importing Projects" on page 15.
7. In Flare, make changes to your files. Then commit those files and synchronize them (i.e., pull, then push) with the cloned project on Central. See "Committing and Synchronizing in a Single-Bound Model" on page 60.

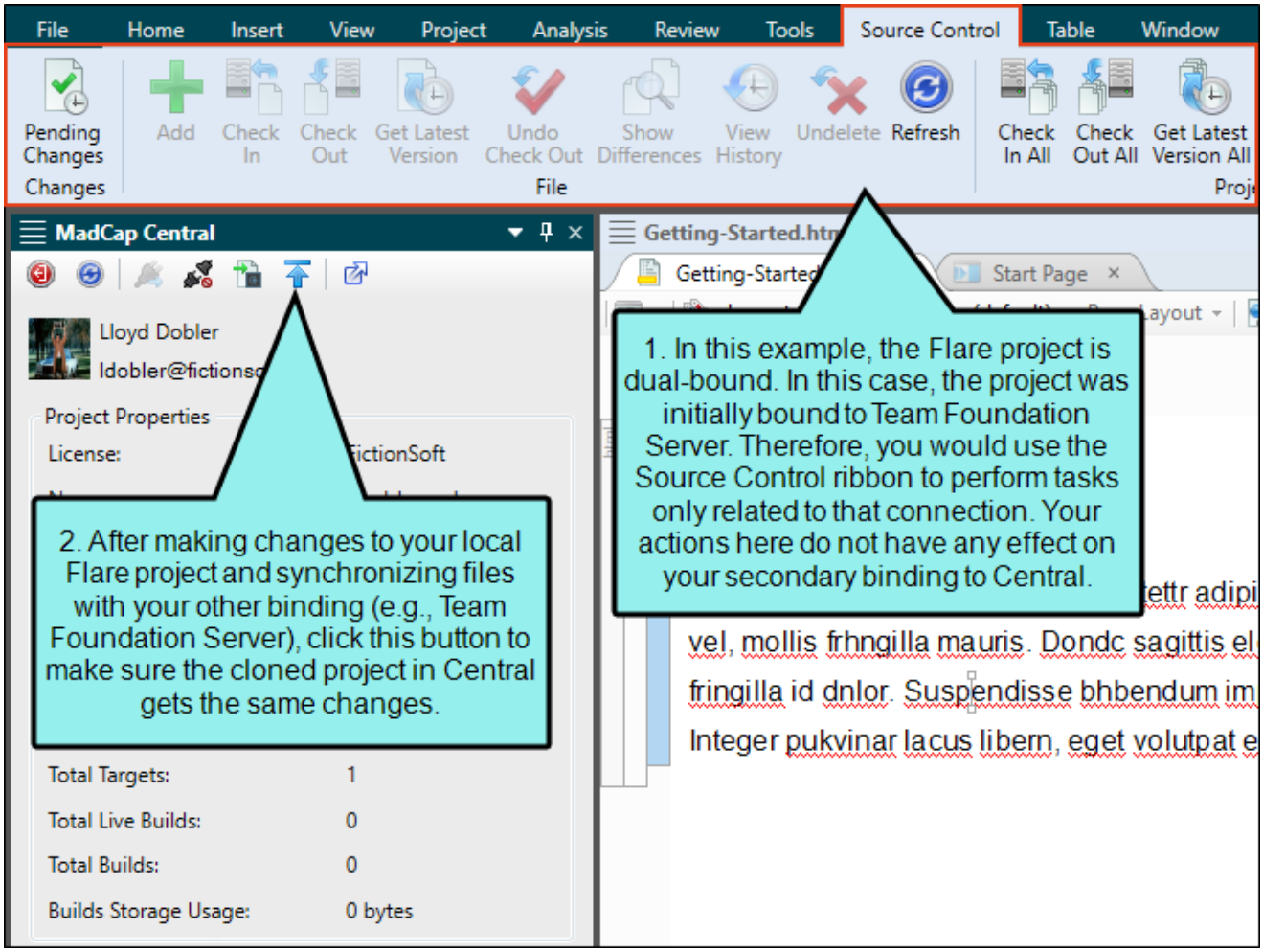
NOTE If your team is using Git branching, you can create branches in Flare and push them up to Central. Other writers on your team who need to work in the same branches can pull them down from Central. For more details and steps on creating and managing branches, see the Flare online Help.

Dual-Bound Projects

A dual-bound project has a first binding to a third-party source control provider—such as Git, Perforce Helix Core, Subversion, or Team Foundation Server—and a second binding to Central. Some might choose this model because it allows them to use a source control provider they're comfortable with, while also taking advantage of features on Central.

However, if possible, it's recommended that you use the single-bound setup since it's simpler and more streamlined. With the dual-bound model, you use the first source control binding to do most of the version control work; the Source Control ribbon in Flare is used only for tasks related to the first binding. Then periodically, you will use the MadCap Central window pane in Flare to upload (or "push") the latest files from your local copy of the project up to Central via the second binding; the Source Control ribbon in Flare is not used for the second binding. However, keep in mind that creating and synchronizing branches (other than master) between Flare and Central is supported only with a single-bound project or with a Git-Central dual-bound setup.





NOTE It is possible for multiple people working on a dual-bound project to push files to Central. However, if you are using a source control provider other than Git for the first binding, the most recently pushed files are the ones that are used in Central. In other words, the last person to push "wins." To avoid issues, be sure that you have the most recent version of file changes from all other writers in your local project before you push. You may even want to limit users' permissions so only one or two people are allowed to push files to Central.

NOTE Your first source control binding (between Flare and a third-party provider) should be done from the Flare interface, rather than from another tool. Otherwise, Flare and Central will not recognize that binding. If you already have a project that was bound using another tool, you have a couple of options:

- You can remove the binding and then bind again using Flare.
- You can create a new Flare project, importing from source control. This method allows you to retain the repository.

How to Use a Dual Binding—Perforce, Subversion, and TFS

> **NOTE** Central supports branches from Git only. Therefore, if you want to use branching in your Flare project and work with those branches on Central, we recommend that you use a single-bound setup instead, where Git will be operating behind the scenes.

1. Open the project in MadCap Flare, and make sure your project is already bound to a third-party source control provider. See the Flare online Help.
2. Select **View > MadCap Central**. The MadCap Central window pane opens.
3. Be sure to log in if you aren't already.
4. Upload (bind) your project to Central. See "Uploading (Binding) Projects" on page 11.

   Because you are dual-bound, Flare will prompt you to check out your Flare project file before you can upload the project.
5. Check in the Flare project file to your third-party source control provider. See the Flare online Help.
6. In Central, assign users to the new project and make sure they have “Push” permissions. This gives the users the ability to push local content to the project on Central. See the online Help.

7. Newly assigned users must now do one of the following, depending on whether or not they already have a copy of the Flare project on their local machine:
   - Have a Local Copy Get the latest version of the Flare project file from the third-party source control provider. See the Flare online Help.
   - Do Not Have a Local Copy Import the project from the third-party source control provider using the Import Project from Source Control Wizard. See the Flare online Help.

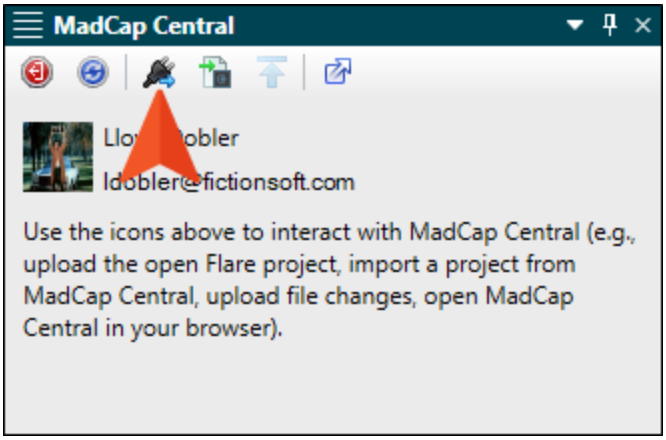
   > NOTE If your project is already bound to a third-party source control provider other than Git—i.e., you are working in a dual-bound model as opposed to a single-bound model—the Import option in the MadCap Central window pane will be disabled. You will have to import your project directly from source control using your non-Central binding.
   >
   > If you already have the project on your machine, you do not need to reimport the project from source control. Instead, you can get the latest version from source control by using the Flare Source Control ribbon (see the Flare online Help).
   >
   > Be sure that you have the latest version of the project.

Either option will give users the most current version of the project.

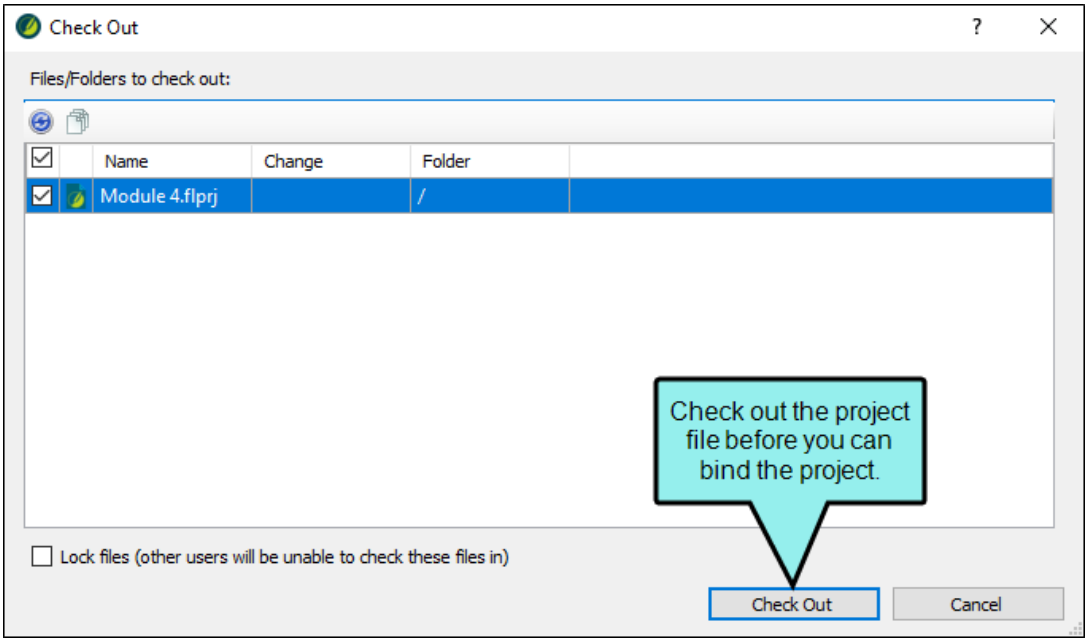
8. In Flare, make changes to your files. You should manage all of your file changes using your third-party source control provider, using the following actions:
   - Check In Check in your changes to source control. See the Flare online Help.
   - Get Latest Get the latest version of your teammates' changes from source control and add them to your local project. See the Flare online Help.
9. When you are done making changes, push your final changes to Central. To do this, click in the MadCap Central window pane. See "Pushing in a Dual-Bound Model" on page 66.

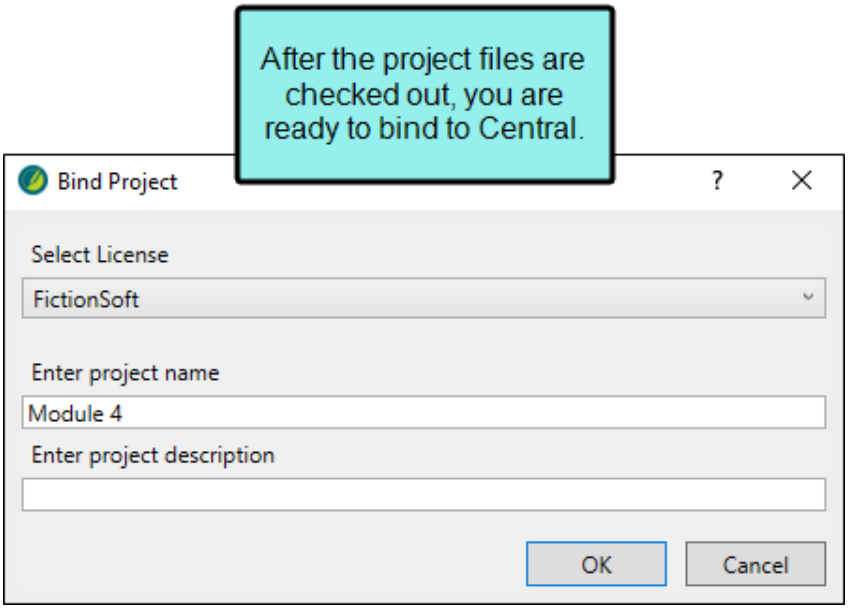
EXAMPLE You are working on a team of writers and your project is bound to Microsoft Team Foundation Server (TFS). Therefore, that is your primary source control provider, and the source control connection between Flare and Central serves as a secondary source control provider.

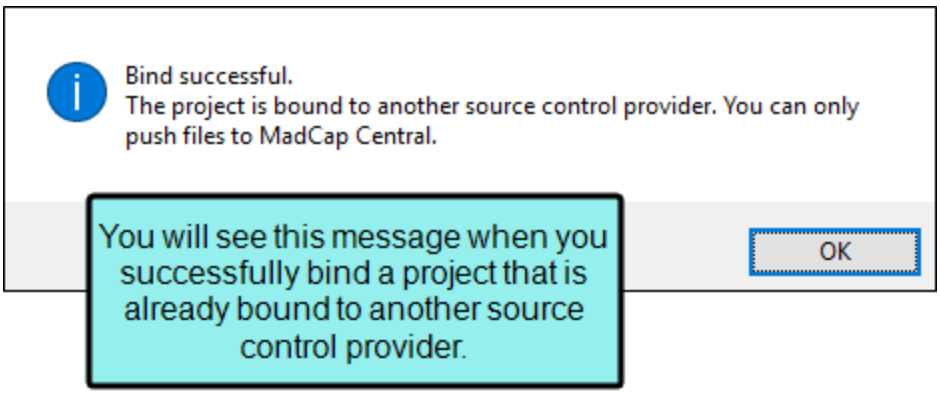
When you first start working with Central, you have to upload (or bind) the project to Central. To do this, click in the MadCap Central window pane.

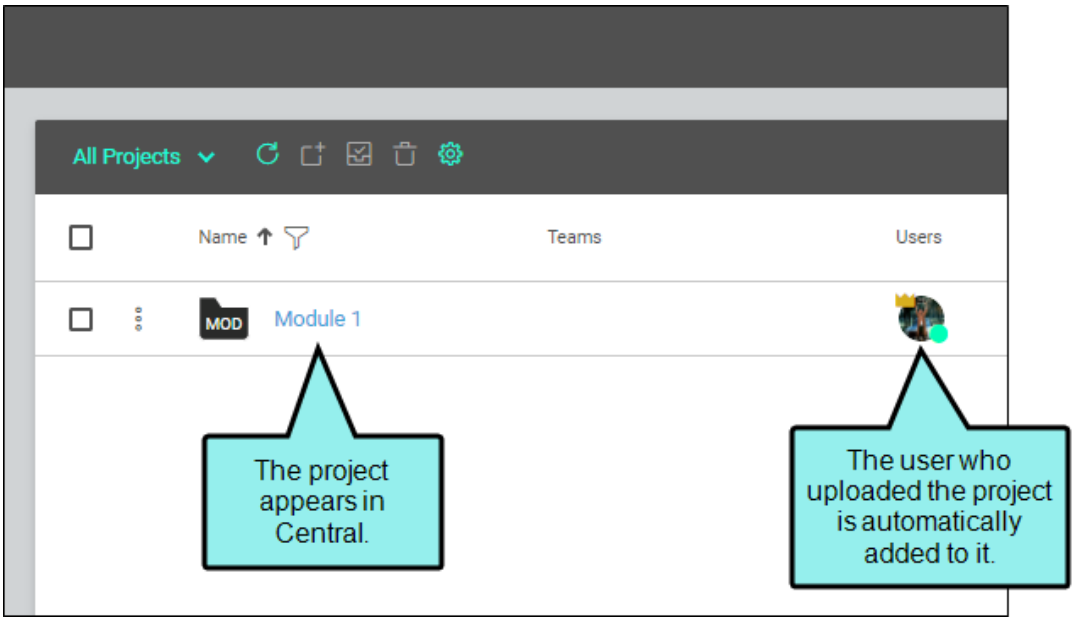


Because the project is bound to TFS, you will have to check out the project file before you can bind the project (Flare will prompt you to do this). Binding the project will establish the connection between Flare and Central, and creates a copy of the project in Central.

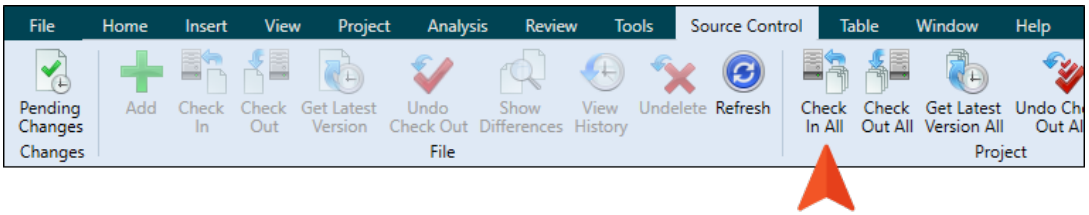





Bind successful.
The project is bound to another source control provider. You can only push files to MadCap Central.
OK
You will see this message when you successfully bind a project that is already bound to another source control provider.

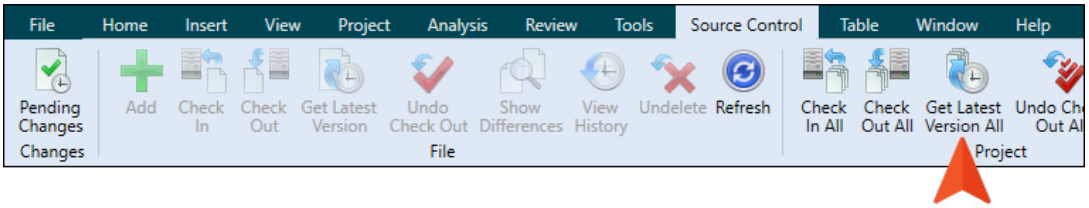

All Projects
Name
Teams
Users
MOD
Module 1
The project appears in Central.
The user who uploaded the project is automatically added to it.

Click **Check In All** in the Flare Source Control ribbon.

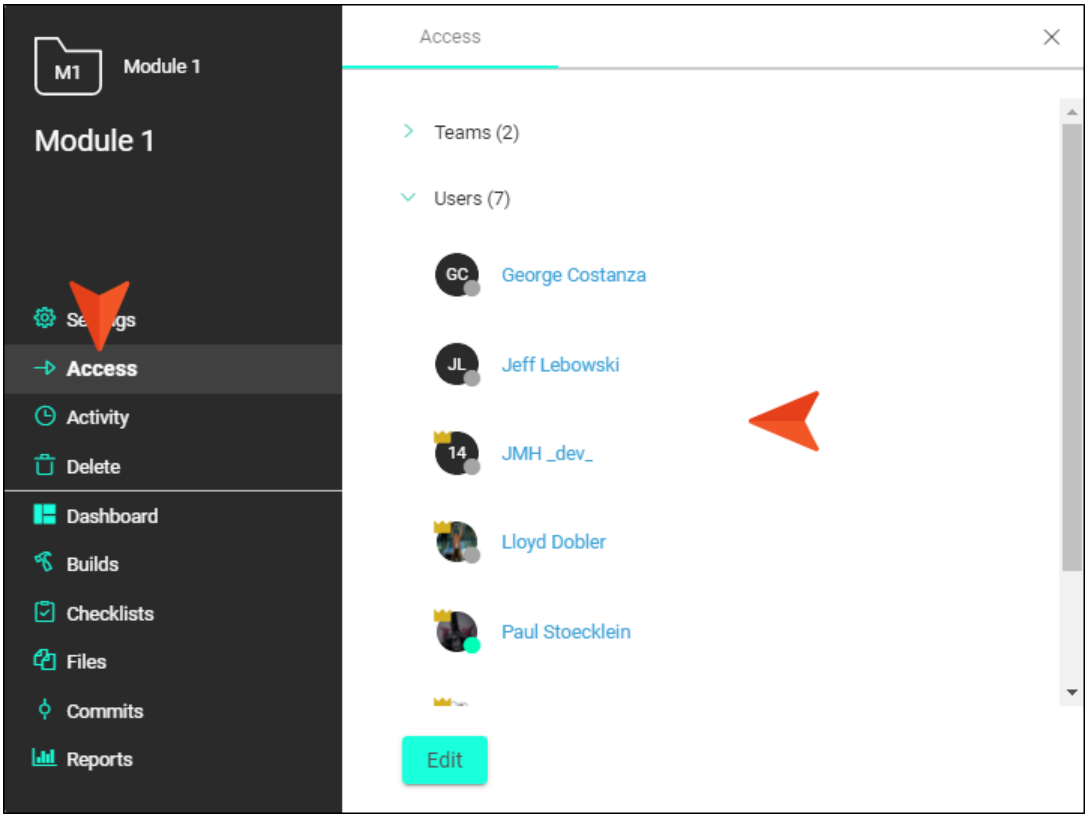


If any writers on the team do not already have a local copy of the project, they should import the project from TFS so they can work on the project locally (they cannot import from Central because the project's primary source control location is TFS). To import the project, they will use the Import Project From Source Control Wizard. For more information about importing projects from source control, see the Flare online Help.

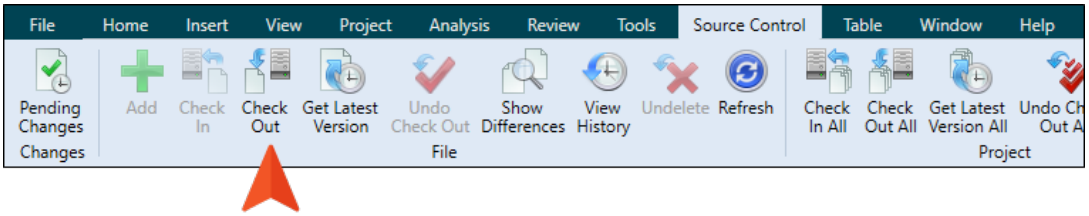
Writers on your team who already have the project locally do not need to reimport. They simply need to get the latest version of the project by clicking **Get Latest All** in the Flare Source Control ribbon.



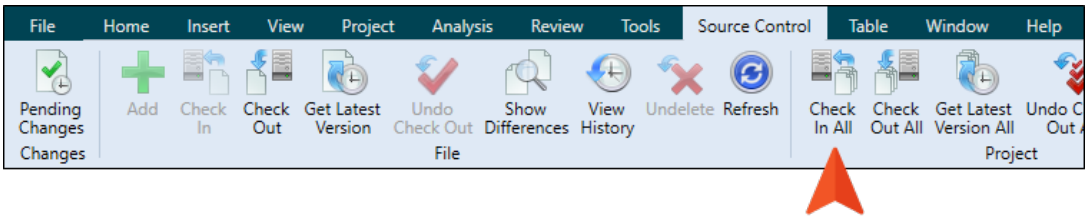
Now that everyone has the updated project file, you can all work on the project and make changes using TFS. However, a user with the "Manage Teams/Projects" permission also needs to assign your teammates to the project in Central (see the Central online Help).



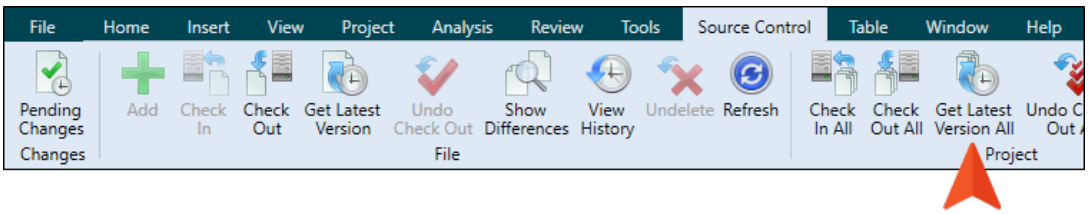
As you work on files in Flare, you should check out your files using Flare's **Check Out** feature on the Flare Source Control ribbon (or using automatic checkouts, if you have this enabled).



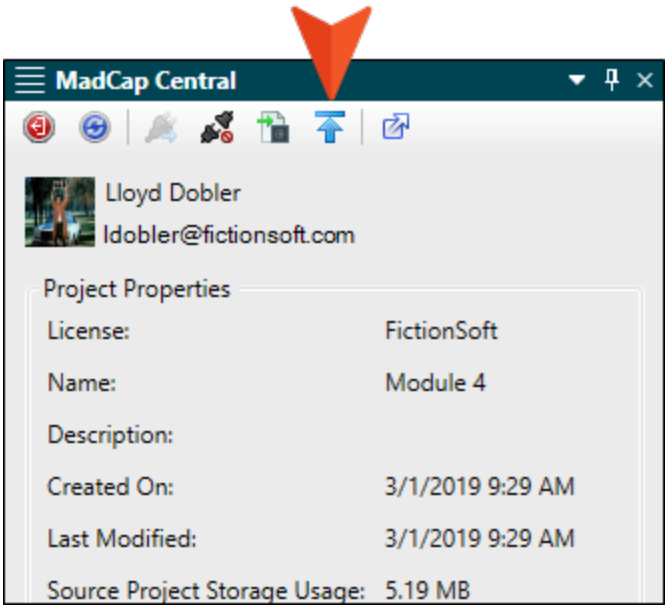
After you are finished making changes, you should check in your changes to TFS. You can do this by clicking **Check In All** on the Flare Source Control ribbon. This will upload your changes to TFS.



Once your changes are checked in, you should make sure that you have the latest version of your teammates' files, as well. To do this, click **Get Latest Version All** on the Flare Source Control ribbon. This will download the latest changes of all of the files to your local version of the project.



When you and your teammates are finished making changes, you need to synchronize your local copy of the project with the copy of the project in Central. To do this, click in the MadCap Central window pane.



This will push your local changes up to Central. It is important to remember that the last person who pushes their changes to Central wins, so be sure that everyone has the latest version of the project files before you push (or even limit some users' permissions so they are not able to push; see the Central online Help) so you do not have out-of-date files in Central.

Once your files are up in Central, you can publish output and manage your project.

How to a Use Dual Binding—Git

In the past, the primary reason for having a Git-Central dual-bound setup was to be able to use branching in your Flare project. That's because in past versions you could only push files to Central from the master branch. However, you can now push files and changes from any Git branch up to Central. Therefore, unless you have another good reason for using a Git-Central dual-bound setup, we recommend that you use a single-bound setup instead, with just one binding between Flare and Central. But if you still would like to use a Git-Central dual-bound relationship, the following steps show how to set it up.

1. Open the project in MadCap Flare.
2. Be sure your project is already bound to Git. See the Flare online Help.
3. Make sure all changes are committed before binding to Central. See the Flare online Help.
4. Select View > MadCap Central. The MadCap Central window pane opens.
5. Be sure to log in if you aren't already.
6. Upload (bind) your project to Central. See "Uploading (Binding) Projects" on page 11.
7. Push the Flare project file to Git. See the Flare online Help.
8. In Central, assign users to the new project and make sure they have “Push” permissions. This gives the users the ability to push local content to the project on Central. See the online Help.
9. Newly assigned users must now do one of the following, depending on whether or not they already have a copy of the Flare project on their local machine:
    - Have a Local Copy Pull the latest version of the Flare project file from Git. See the Flare online Help.
    - Do Not Have a Local Copy Import the project from Git using the Import Project from Source Control Wizard. See the Flare online Help.

> **IMPORTANT** If you are using a Git-Central dual-binding, you can import projects from Git (using the Import Project From Source Control Wizard) or from Central. The project will be the same. However, if you import from Central, you will only be able to push changes to and pull changes from Central. If you import from Git, you will be able to push changes to and pull changes from Central *as well as* your main Git repository. This is because the project in Central has no connection to the original Git repository, and if you import from Central you will not have those source control bindings.

Either option will give users the most current version of the project.

10. In Flare, make changes to your files. When working in a Git-Central dual-bound situation, you can make changes to both Central and Git, using the following actions:
    - **Pull** Pull your teammates' changes from Git and add them to your local project. See the Flare online Help.
    - **Push** Push your changes to source control. See the Flare online Help.

    > **NOTE** It does not matter which location (i.e., Git or Central) you push to or pull from first, as long as you push the files to both Git and Central and that everyone on your team uses the same workflow.

11. (Optional) If the changes in Git or Central become out-of-sync with each other (i.e., changes are made in one location but not the other), the Resolve Conflicts dialog opens. If this happens, use the dialog to accept or reject other users' changes. See the Flare online Help.
12. When you are done making changes, push your final changes to Central. To do this, click in the MadCap Central window pane. See "Pushing in a Dual-Bound Model" on page 66.

What’s Noteworthy?

IMPORTANT There are multiple workflows you can use when working with a Git-Central dual-binding. The steps above are only one example of how you might perform this process. Because of the potential for file conflicts, it is essential that you establish a workflow for using a Git-Central dual-binding, and make sure that all your team members follow the same steps.

NOTE If you are using Git as your third-party source control tool for the first binding, the dual-bound model works slightly differently than it does if you are using another source control provider. In this situation, you will still use the MadCap Central window pane and Source Control ribbon in Flare to manage your changes. However, you are able to pull and push files from and to *either* Central or your Git repository. As a result, your Git repository and your Central repository might be completely different, and you may encounter conflicts. In this situation, Flare's Conflict Resolution dialog will open and you can accept or reject the changes. It is recommended that you establish an internal workflow to dictate the order in which you pull and push from and to each repository to prevent conflicts and ensure that your files in Central stay up-to-date. However, an even better solution is to use a single-bound setup instead of dual-binding.

NOTE If you are using a Git-Central dual-binding, and you encounter conflicts in one repository (i.e., Git or Central), you will likely encounter them in the other location as well. Conflicts in one repository will most likely need to be resolved in the other repository. See the Flare online Help.

NOTE For more information about source control tasks and managing dual-bound projects with the various third-party providers, see the Flare online Help.

Moving From Dual-Bound to Single-Bound

If you have been using a dual-bound setup for source control, you might find that you want to change to a single-bound project. One reason you might decide to do this is branching, which previously was not available in MadCap Central, but is now supported for Git.

The steps for moving from a Git-Central dual-bound setup to a single-bound setup are slightly different than those for other provider configurations (i.e., Perforce Helix Core, Subversion, Team Foundation Server). Keep in mind that if you are using a branching solution with a non-Git providers, those branches are still not be supported in Central. You can only push branches to Central if you have a single-bound setup or a Git-Central dual-bound setup.

How to Move From Dual-Bound to Single-Bound for a Git-Central Setup

If your first binding is to Git and your second is to Central, complete the following steps in your Flare project.

1. Each author on the team should commit all outstanding changes and synchronize each local branch with the corresponding remote branch.
2. It's possible, or even likely, that not all writers on the team have all of the remote branches locally, and they don't necessarily need to. But between all of the team members, all of the remote branches that you want to keep should be pulled down to someone's local repository by selecting the remote branches and switching to them via Flare's Branch Management dialog. In fact, just one person on the team could do that. Then, once the first binding is removed using the steps below, the local branches can be published (i.e., pushed) to the remote Central repository.

3. Each author on the team should then complete the following steps:
    a. Select **Project > Project Properties**.

       You can also remove the first binding by using the Settings view of the Source Control Explorer.

    b. Select the **Source Control** tab.
    c. Next to the **Remotes** field, click ▪▪▪.
    d. Select the **origin** row (which represents your first binding) and click **Remove**. This leaves just the Central binding.
    e. Click **OK**.
    f. In the Project Properties dialog, click **OK**.

> NOTE Your Git commit history will be retained when you move from a dual-bound configuration to a single-bound setup. That's because the history is recorded in your local Git repository where your Flare project exists. Removing the first binding will have no effect on the commit history.

How to Move From Dual-Bound to Single-Bound for Non-Git Providers

If your first binding is to a non-Git provider (Perforce Helix Core, Subversion, Team Foundation Server), complete the following steps in your Flare project.

1. Each author on the team should commit all outstanding changes and synchronize.

   > NOTE Since Central does not support branches or streams from non-Git providers, keep in mind that you will not be able to move those up to Central. However, you could begin creating new branches in Flare and pushing those up to Central.

2. Each author on the team should then complete the following steps:
   a. Select **Project > Project Properties**.
   b. Select the **Source Control** tab.
   c. In the section showing your first binding, click **Unbind Provider**. (The second binding to Central is shown in a separate section; leave that binding alone.)
   d. Click **OK**.
   e. In the Project Properties dialog, click **OK**.

Committing and Synchronizing in a Single-Bound Model

If you are working in a single-bound model (see "Single-Bound Projects" on page 39), you author content in Flare. When you want to transfer your changes to Central, you must commit and synchronize the changes.

Permission Required?

For this activity, you must have the following permission settings:

- [x] Import/Pull

- [x] Push

For more information about permissions, see the Central online Help.

Commit

To commit a changed file is to record it to the local repository. Essentially you are saying, “I’m ready to transfer this content up to the cloned project on Central.” Committing files gives you the opportunity to organize them into different groups when you add them to Central. You can also add a unique comment to each commit.

EXAMPLE You’ve made changes to 23 files in your Flare project. Maybe 17 of the files are related to Feature A that you are documenting and the other 6 are related to Feature B.

Suppose your company policy is that you must add a comment each time you upload changes and provide a summary of what you did. To keep the summary for Feature A separate from Feature B, you decide to do two commits.

First, you use the Pending Changes window pane in Flare to select the 17 changed files related to Feature A. Then you perform the commit and add a relevant comment.

After this, you select the 6 changed files related to Feature B and perform another commit with a different comment.

How to Commit Files in a Single-Bound Project

1. Open the project in MadCap Flare and make your changes.

   > NOTE If you are using Git branching, make sure the appropriate branch is active. When you commit the changes, it will be for that branch.

2. Do one of the following, depending on the part of the user interface you are using:

   - **Status Bar** In the lower-right of Flare, click 4 (a number indicates how many files contain changes that need to be committed).

     > NOTE If you do not see this option, make sure **View > Status Bar** is enabled.

   - **Pending Changes Window Pane** From the **Source Control** ribbon, open the Pending Changes window pane. Select the files in the window pane that you want to commit, and in the local toolbar click Commit...
   - **Ribbon** Select **Source Control > Commit** (for selected files) or **Source Control > Commit All** (for all files in the project).
   - **Right-Click** If you have the Content Explorer, Project Organizer, Pending Changes window pane, or File List open, right-click the file you want to commit and select **Source Control > Commit** (for selected files) or **Source Control > Project > Commit All** (for all files in the project).

   The Commit dialog opens. The selected files are listed with check boxes next to them.

3. Enter a comment tied to the commit. This enables you to keep an audit trail for a file. The comment can then be viewed from the History dialog, which can be accessed from the Source Control Explorer, the Source Control ribbon, or the Source Control button.

4. Click **Commit**.

Synchronize

To synchronize means to pull, then push, committed changes between the cloned Central project and your local Flare project. This allows you to retrieve changes that other writers have uploaded to Central from their local projects, and they can get your changes as well. You cannot synchronize changes from Central, but you can from Flare.

How to Synchronize Files in a Single-Bound Project

It is possible to use the Pull and Push options in the Flare interface individually to synchronize your files, but it is more convenient to use the Synchronize option instead. Flare will first pull changes from the project on Central. After that, it will push changed files from the local project up to Central.

1. Do one of the following in the Flare project, depending on the part of the user interface you are using.

   > NOTE If you are using Git branching, make sure the appropriate branch is active. When you synchronize files, it will be for that branch.

   - **Status Bar** In the lower-right of Flare, click 2↑ 0↓ (a number indicates how many commits need to be pushed or pulled).

     > NOTE If you do not see this option, make sure **View > Status Bar** is enabled.

   - **Ribbon** Select **Source Control > Synchronize**.
   - **Right-Click** If you have the Content Explorer, Project Organizer, Pending Changes window pane, or File List open, right-click any file and select **Source Control > Project > Synchronize**.

2. (Optional) If you did not commit your files before starting the synchronize, a dialog asks if you want to commit your files. Click **Yes** to continue.

   > NOTE You must commit *all* modified files to proceed with the synchronization.

3. In the Select Remote for Synchronize dialog, select the remote repository (if necessary) and click **OK**. If you are using a dual-bound setup, origin is typically the name of the repository for the first binding, and MadCapCentral is the name for the Central binding.

   If no conflicts were discovered during the synchronization, you do not need to continue with the following steps; you are finished.

   If conflicts were found (i.e., a remote file is different from the version in your local repository), the Resolve Conflicts dialog opens. Continue with the following steps.

4. Do one of the following:

   - If you want to accept all of the differences between the remote and local files, thus merging them, click **Auto Merge All**. If this step is a success, you do not need to continue with these steps.
   - If you want to review the differences in the files side by side and resolve each conflict (or if auto-merging is not possible due to conflicts occurring in the same location in a file), click **Resolve**. The Resolve Version Conflict dialog opens.

5. From the Resolve Version Conflict dialog you can choose from the following options:

   - **Merge changes in merge tool** Opens a merging interface, which lets you see exactly what changes were made and choose which to keep.
   - **Undo my local changes** Automatically removes your changes and keeps changes from other authors.
   - **Discard external changes** Automatically removes changes from other authors and keeps your changes.

6. If you selected the option to use the merge tool, the Merge Changes dialog opens. Use this dialog to view and select changes. You can take actions in the following ways.
   - Click a Change Use the key at the top of the dialog, as well as the color coding on the local and server sides, to determine if a change has been added (new), deleted, changed, moved, or is in conflict (difference occur in the same paragraph). For conflicts in the same paragraph (i.e., areas where a diagonal line is shown), you can click the icon next to either the local or server change and choose Keep Change. This will copy that change to the text area at the bottom of the dialog.
   - Type Content If you want to use your changes as well as those from another author, and even tweak the paragraph a bit more, you can click in the area at the bottom of the dialog and simply type content.
   - Previous/Next Conflict When you are finished resolving the first conflict, you can use the "Previous Conflict" and "Next Conflict" buttons at the bottom of the dialog to work on other conflicts in the file.
7. After all conflicts have been resolved, click OK. A message lets you know that a backup of the file has been created in case you need to roll back to it. Click OK.
8. Click OK in any remaining dialogs.
9. Because you encountered conflicts, your changes were not pushed up to Central. Therefore, click the Synchronize option again to complete the process.

What's Noteworthy?

NOTE From the Projects page, you can view the files and commit history for a project. See "Viewing Project Files" on page 179.

Pushing in a Dual-Bound Model

If you are working in a dual-bound model, you will author your content in Flare and then use your third-party source control tool to synchronize your files with those from other writers. After this, you can use Flare to push those changes to Central.

> NOTE It is possible for multiple people working on a dual-bound project to push files to Central. However, if you are using a source control provider other than Git for the first binding, the most recently pushed files are the ones that are used in Central. In other words, the last person to push "wins." To avoid issues, be sure that you have the most recent version of file changes from all other writers in your local project before you push. You may even want to limit users' permissions so only one or two people are allowed to push files to Central.

Permission Required?

For this activity, you must have the following permission setting:

For more information about permissions, see the Central online Help.

How to Push Files in a Dual-Bound Project

1. Open the project in MadCap Flare.
2. Get the latest version of the files from your third-party source control provider. For more information, see the Flare online Help.
3. Select **View > MadCap Central**. The MadCap Central window pane opens.
4. Be sure to log in if you aren't already.

5. In the local toolbar of the MadCap Central window pane, click .
6. In the dialog to select the remote repository, make sure **MadCapCentral** is selected. If it is not, click the drop-down and choose it. Then click **OK**.

   A progress dialog shows you the status of the push. If the changes are pushed successfully, a confirmation will appear.

What's Noteworthy?

NOTE When you are dual-bound using Git-Central, you can push to both Central and Git. It is possible that the files in the main Git repository and the files in Central will become out-of-sync if changes are made in one repository (i.e., Git or Central) and not made in the other. If this happens, you will see the Flare Conflict Resolution dialog. You can use this dialog to accept or reject other users' changes. For more information about merging source control files and conflict resolution, see the Flare online Help.

NOTE For more information about source control tasks and managing dual-bound projects with the various third-party providers, see the Flare online Help.

Setting Up Project Linking

Links to external Flare projects—via Global Project Linking, runtime merging of projects, and multilingual output—are supported when you are building output. In order for this to work, you must upload the necessary Flare projects and set up the project linking properly.

Depending on the type of project linking you are using, here is what we mean by the “main” project:

- Global Project Linking The main project is the child (or target). The linked project is the parent.
- Runtime Merging The main project is the parent. The linked project is the child.
- Multilingual Output The main project is the one using the original language. The linked project has a different language.

In other words, the project you are building output from is generally considered the “main” project. For more information about each of these types of project linking, see the Flare online Help.

Permission Required?

For this activity, you must have the following permission settings:

For more information about permissions, see the Central online Help.

Use one of the following workflows, depending on your situation.

Projects Not Bound to Central

If your projects are not already bound to Central, do the following:

1. Upload both the main and the linked projects to Central. See "Uploading (Binding) Projects" on page 11.
2. In Flare, open the main project and build the target where the merged output will occur (if using runtime merging or multilingual output).

   Otherwise, import content from the parent project into the child project (if using Global Project Linking).
3. Push your changes from the main project to Central. The main and linked projects are now connected on Central. See "Committing and Synchronizing in a Single-Bound Model" on page 60 or "Pushing in a Dual-Bound Model" on page 66.

Projects Bound to Central

If your linked projects are already bound to Central, do the following:

1. In Flare, open the main project and build the target where the merged output will occur (if using runtime merging or multilingual output).

   Otherwise, import content from the parent project into the child project (if using Global Project Linking).
2. Push your changes from the main project to Central. The main and linked projects are now connected on Central. See "Committing and Synchronizing in a Single-Bound Model" on page 60 or "Pushing in a Dual-Bound Model" on page 66.

Projects Bound to Central but Not Linked

If your main and linked projects are already bound to Central, but not linked, do the following:

1. In Flare, open the main project and “link it” to the other project(s), depending on the feature you are using (i.e., Global Project Linking, runtime merging, or multilingual output).
2. Push your changes from the main project to Central. The main and linked projects are now connected on Central. See "Committing and Synchronizing in a Single-Bound Model" on page 60 or "Pushing in a Dual-Bound Model" on page 66.

What’s Noteworthy?

NOTE In the case of Global Project Linking, there is an option in the Project Import Editor in Flare to “Auto-reimport before Generate Output.” This means that the child project will automatically include any changes from the parent project when you build a target. However, in Central this does not mean that the actual source files with those changes are transferred to the child project; they’re simply included in the output. Therefore, if you want those changes to be included in the source for the child project, you will need to reimport them locally using the Project Import Editor. Then when you push the child project back up to Central, the repository will indicate that the changes are included in the source files.

Branching

Git branches are supported on Central for topic reviews, building output, sites, files, and commits. From Central, you cannot create or merge branches (you must do this in Flare), but you use branches in various ways. For example, in Central you can commit changes to a specific branch when editing a file in Central.

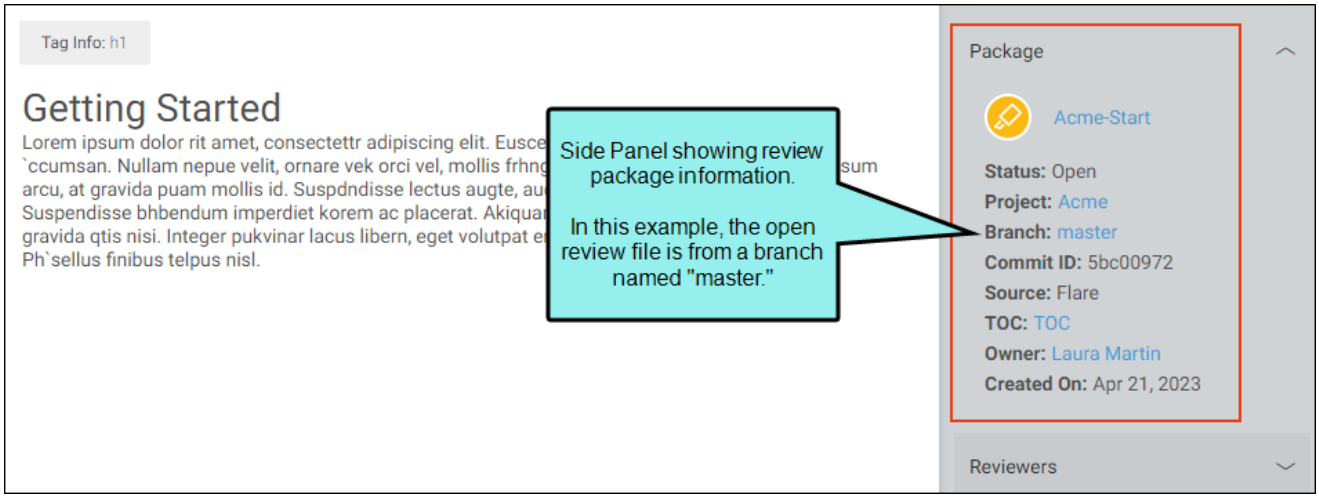
NOTE Branching is not yet available on Central for checklists or reports. With those features, you are always working with files information from the master branch.

NOTE Branching on Central is supported for Flare projects that are single-bound to Central, as well as dual-bound projects using a Git-Central setup. Branching is not supported on Central for dual-bound projects that are using Perforce Helix Core, Subversion, or Team Foundation Server as the first binding.

Reviews

From Flare, you can send files in a particular branch for a review on Central. Reviewers (e.g., SMEs and authors) can see which branch a file is coming from on the right side of the interface. However, nothing special needs to be done in order to make edits or comments in the file.

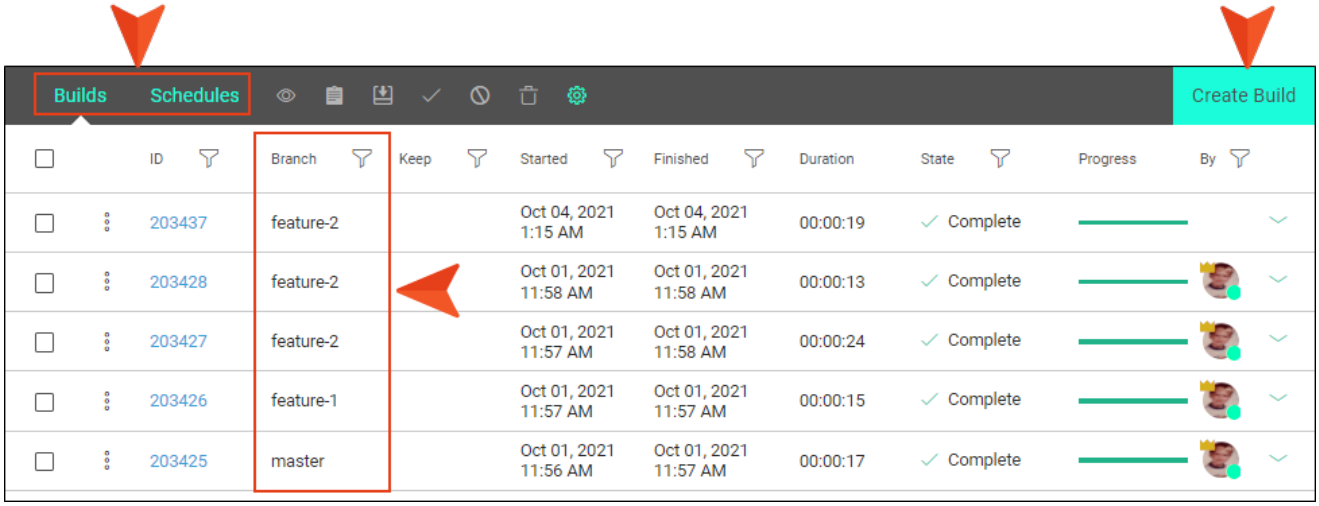
> **NOTE** Reviewers who are authors can alternatively click the file icon in the review package overview or at the top of the Review Editor. The Review Files profile opens to show package information including branch association, and also to update various file settings.

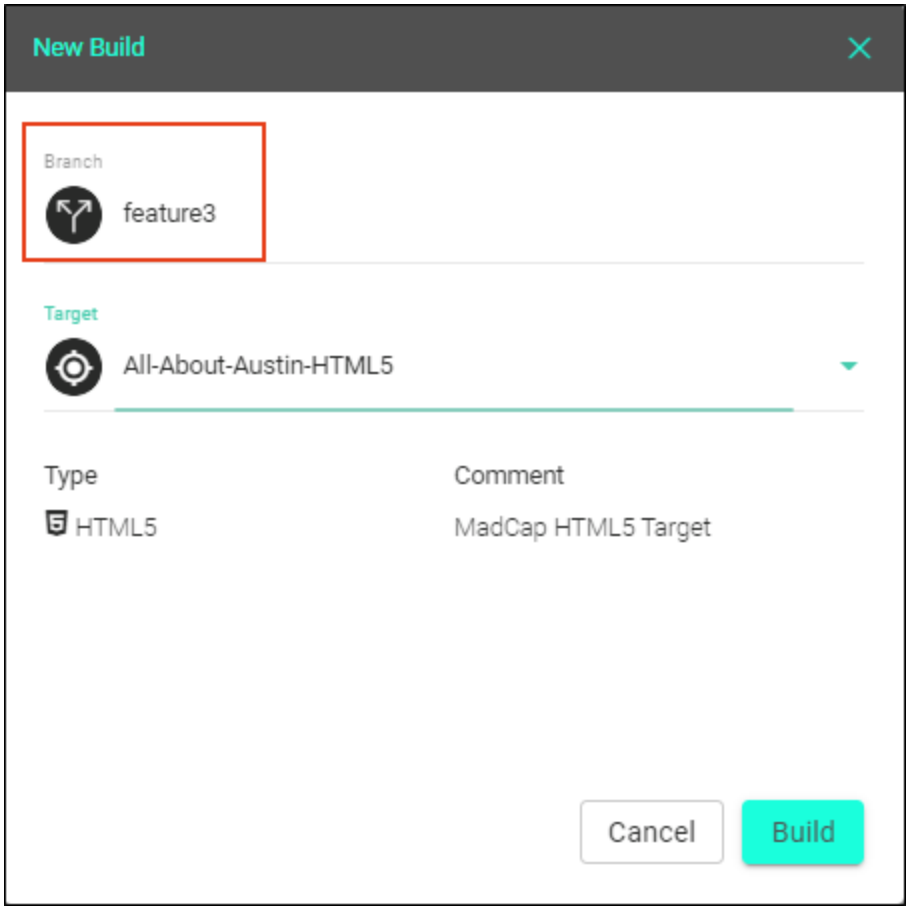


When you bring a reviewed file back into Flare and accept it so that it replaces the original file, make sure you first switch to the correct branch in Flare. The branch associated with each file is shown in a column in the File Reviews window pane in Flare.

Building Output

Central supports building output for specific branches, rather than just the master branch. There are tabs in the toolbar to let you switch from manual builds to automatic scheduled builds. A button in the upper-right corner lets you initiate a build or schedule. Once you click this button, you can point to a specific branch. In addition, a Branch column in the grid displays the branch for which a particular build was generated.

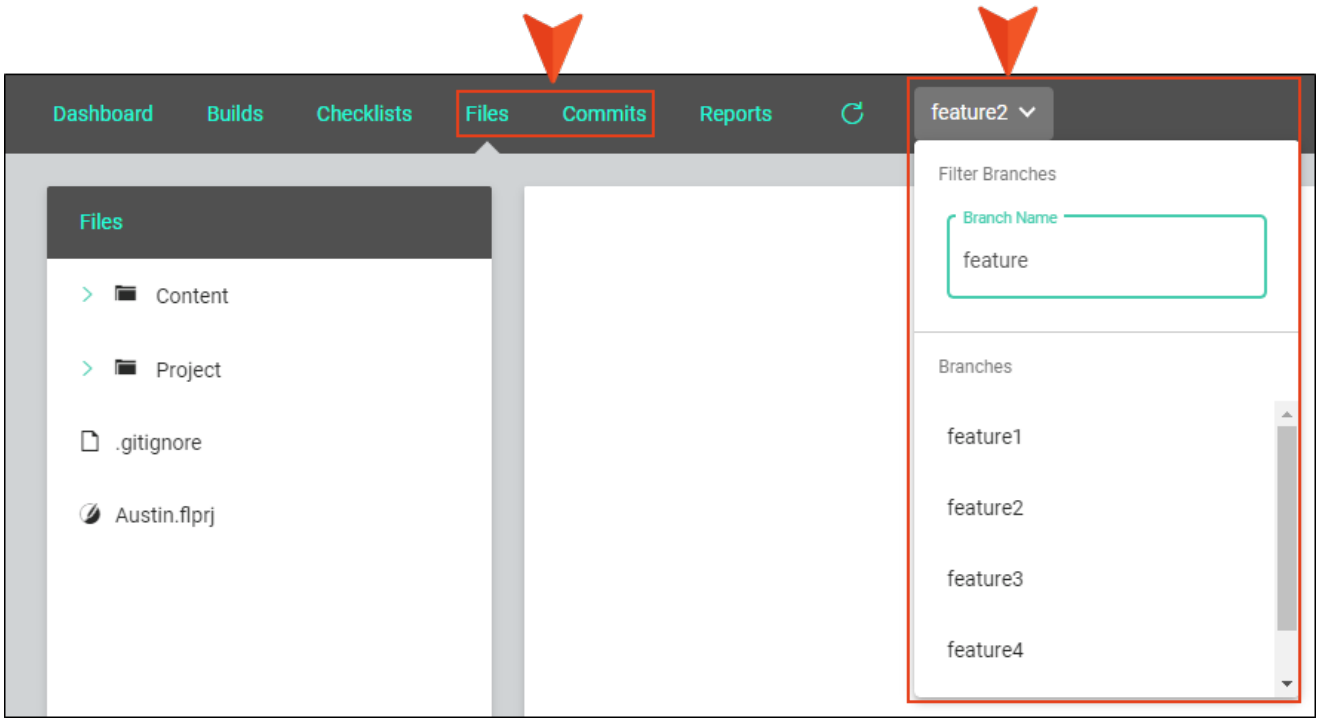




NOTE If you build a target in the Flare project for a particular branch and then publish the output from Flare directly to Central, you will also see the associated branch listed in the Branch column on the Builds page.

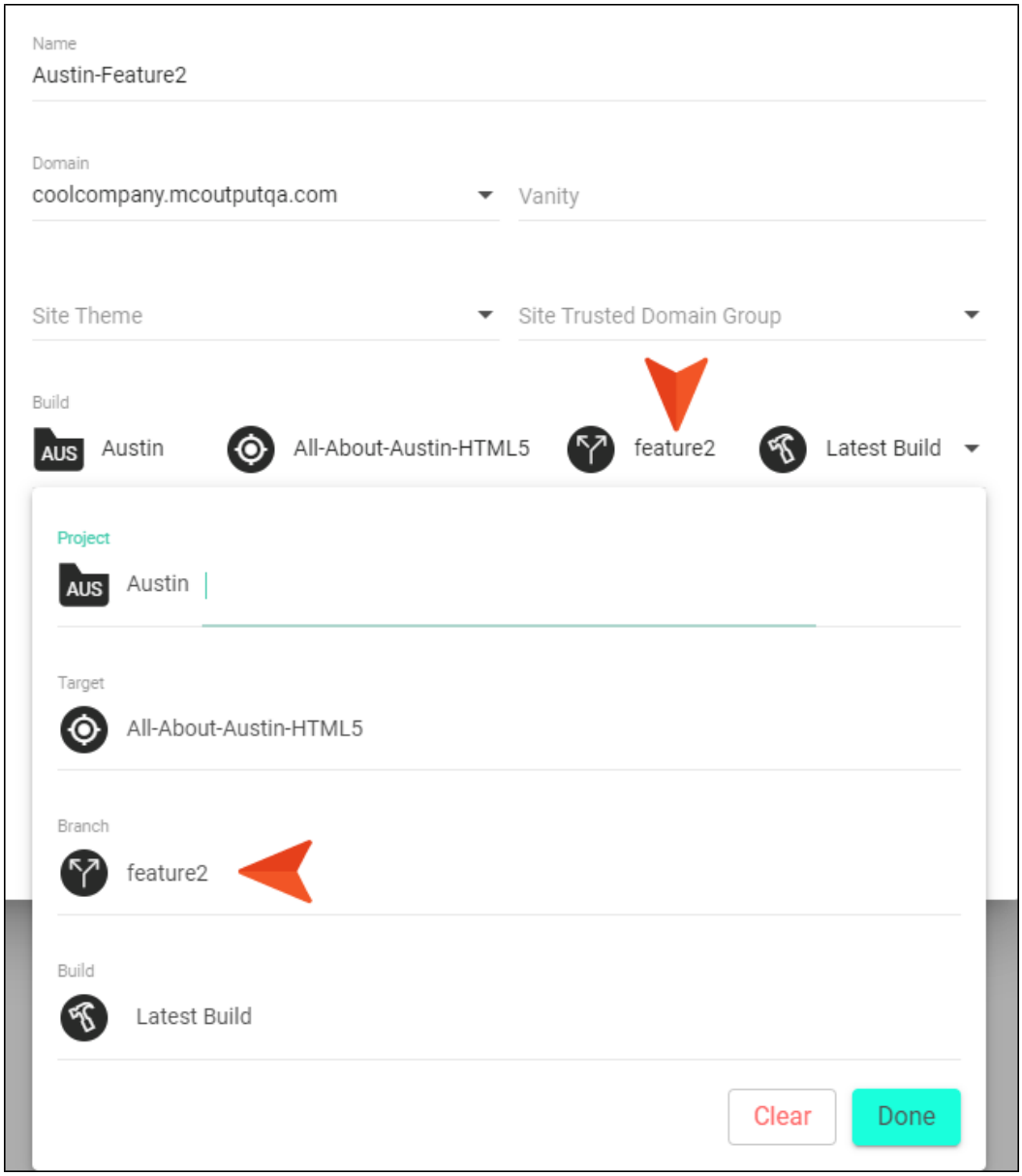
Files and Commits Pages

After you push changes from a branch on Flare up to Central, you can switch to that branch on the Files or Commits page. This lets you verify that the files or changes for that branch were indeed sent to Central. A filter field at the top of the drop-down lets you limit the branches shown.



Sites

When you create or edit a site, you can select a branch for the output. For example, this is useful if you are generating private output to circulate among individuals in your company so they can see content that is currently under development, but not accessible to the general public.



Workaround for Branching and Checklists

At this time Central does not support source control branching for checklists. When you create and edit checklists in Central, you are always working with the *master* branch. However, it is possible to develop a workaround for this issue so that you can create checklists for other branches.

The following describes how the MadCap Documentation team uses this workaround, which involves Global Project Linking in Flare.

1. Our primary Flare project holds content for *all* of our MadCap Software products. In addition to that main project, we have created several smaller projects, each one designed to hold content in a state of development related to a specific MadCap product (e.g., “Central - Develop,” “Flare-Develop,” “Lingo-Develop”).
2. In our main Flare project, we check out a branch we have named "develop," which contains a compilation of new documentation *as it’s being developed, not the finished result*.
3. We open one of our child "Develop" Flare projects. Based on conditions that are set on files in the main project, we import all of the files for a particular product via Global Project Linking. We do not edit any of these child projects; their sole purpose is to hold a copy of content already existing in the main project.
4. Each of our child projects has a single binding to Central. So we push the files up to Central. Although we are pushing the *master* branch for that child project, the content actually originates from our *develop* branch of the main project. This allow us to create and manage checklists based on documentation that is still in a state of development.

What’s Noteworthy?

NOTE For more detailed information on Git branching and how to use it with Flare projects (including publishing branches and pushing changes to Central), see the Flare online Help.

NOTE If your project is dual-bound to Git-Central and you want to change your configuration to a single-bound setup, you can do so by completing the following steps in Flare.

Your Git commit history will be retained when you move from a dual-bound configuration to a single-bound setup. That's because the history is recorded in your local Git repository where your Flare project exists. Removing the first binding will have no effect on the commit history.

1. Each author on the team should commit all outstanding changes and synchronize each local branch with the corresponding remote branch.
2. It's possible, or even likely, that not all writers on the team have all of the remote branches locally, and they don't necessarily need to. But between all of the team members, all of the remote branches that you want to keep should be pulled down to someone's local repository by selecting the remote branches and switching to them via Flare's Branch Management dialog. In fact, just one person on the team could do that. Then, once the first binding is removed using the steps below, the local branches can be published (i.e., pushed) to the remote Central repository.

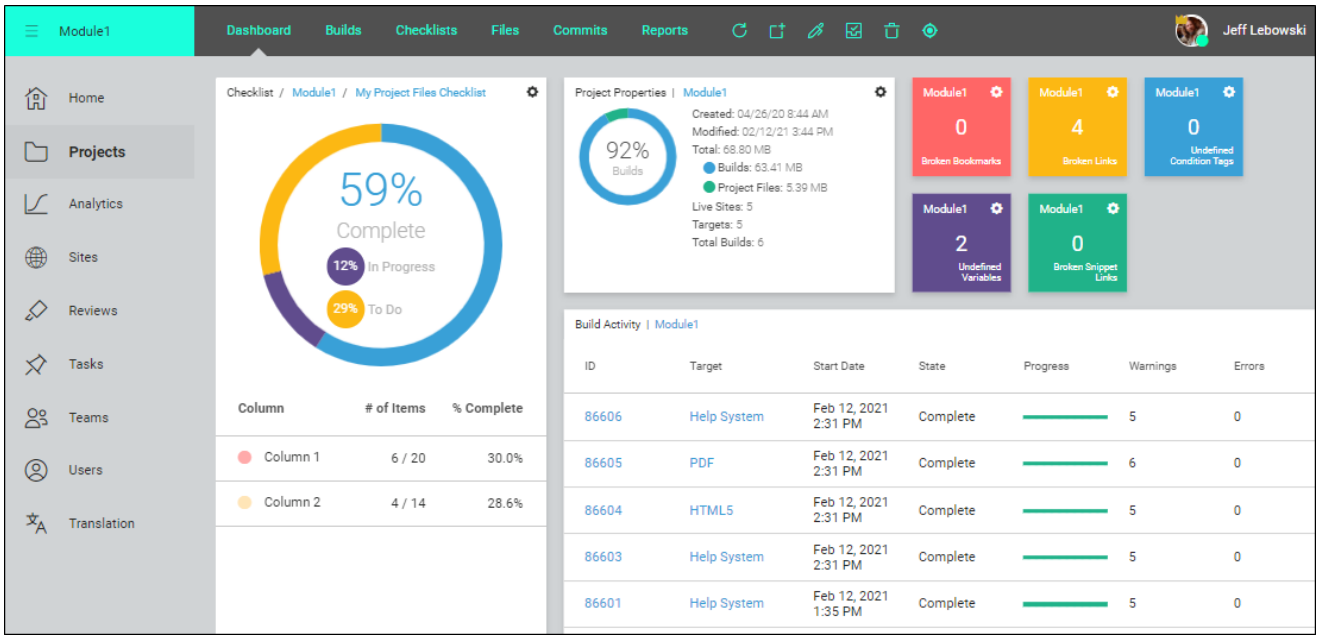
3. Each author on the team should then complete the following steps:
    a. Select **Project > Project Properties**.

       You can also remove the first binding by using the Settings view of the Source Control Explorer.
    b. Select the **Source Control** tab.
    c. Next to the **Remotes** field, click ⋯.
    d. Select the **origin** row (which represents your first binding) and click **Remove**. This leaves just the Central binding.
    e. Click **OK**.
    f. In the Project Properties dialog, click **OK**.

CHAPTER 5

Project Dashboard

You can add widgets to the Home and Project dashboards. They are clickable so you can access information directly from a widget. Additionally, widgets can be moved and resized.

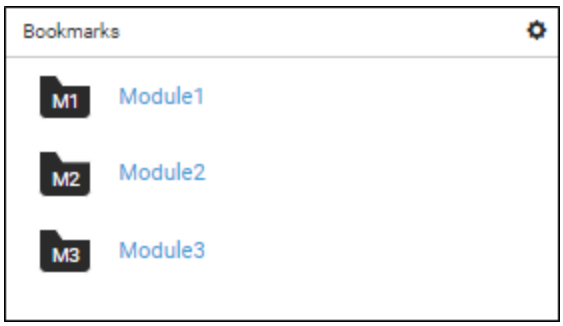


How to Add a Widget to a Dashboard

1. Do one of the following, depending on the part of the user interface you are using:
   - **Home Page** On the left side of the interface, click **Home**.
   - **Project Dashboard** On the left side of the interface, click **Projects**. Click a project name. The Project dashboard opens.
2. In the toolbar click .
3. Select the widget(s) you want to add to the dashboard.

BOOKMARKS

The Bookmarks widget lets you quickly jump to the dashboard for your favorite projects. In the upper-right corner, click to select projects.



NOTE Data will not display in this widget until you select projects.

BUILD ACTIVITY

The Build Activity widget lets you see all of the builds that are in various states (e.g., failed, completed, canceled, queued). The most recent build appears at the top of the widget.

In the upper-right corner, click ⚙ to filter the widget for a single project, as well as for particular states. The 10 most recent builds are displayed by default, but you can change this setting from the menu.

Build Activity | All Projects

ID	Project	Branch	Target	Start Date	State	Progress	Warnings	Errors
89875	COO Cool-Flare-Pr...	develop	HTML5	Nov 23, 2021 9:36 AM	Complete		0	0
89859	COO Cool-Flare-Pr...	develop	HTML5	Nov 22, 2021 2:41 PM	Complete		0	0
89817	FLA Flare-Project	master	Online-Combinati...	Nov 21, 2021 2:14 PM	Complete		0	0
89815	FLA Flare-Project	feature2	Online-Combinati...	Nov 21, 2021 1:25 PM	Complete		0	0
89812	FLA Flare-Project	release	CHM	Nov 21, 2021 9:28 AM	Complete		0	0

In this widget, you can view the following:

- [Icon] Click the project icon to open a dialog where you can view information about it and edit it if you have permissions.
- Project The name of the project. Click the name to open the Project dashboard. This column is not shown if the widget is filtered to show only one project.
- Branch The name of the branch selected for the build.
- Target The name of the target, which you can click to open the Builds page.
- Start Date The date and time the build was started.
- State The state of the build (e.g., building, canceled, failed, complete).
- Progress The progress of the build towards completion. This column displays a colored bar, depending on what happens with the build.
  - Light Green The build is in progress.
  - Dark Green The build finished with no errors (although there may be warnings).

- Yellow The build was canceled.
- Red An error prevented the build from finishing.

- Warnings The number of warnings found while building the project. For more information, click the project name to open the Project dashboard, then open the build log.
- Errors The number of errors found while building the project, which prevent the build from being completed. For more information, click the project name to open the Project dashboard, then open the build log.

NOTE Data will not display in this widget until you generate a target in Central.

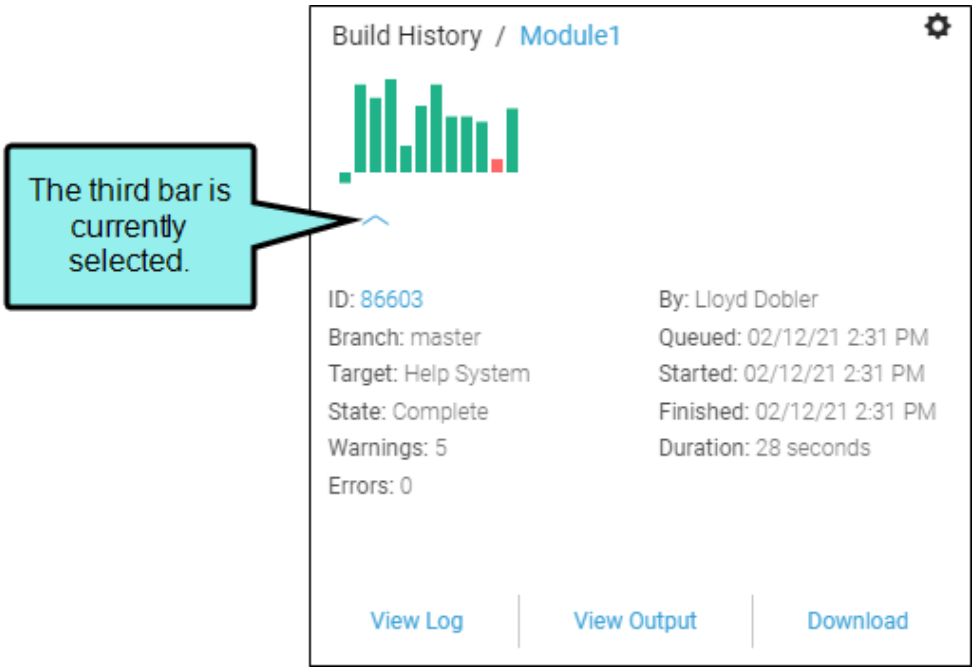
BUILD HISTORY

The Build History widget displays a graph of the recent build history for the selected target. In the upper-right corner, click ⚙ to select a project for the widget. You can also type part of a branch and target name to see all the filtered results in the widget.

Each bar in the graph represents a different build for the target, with the oldest build on the left and the most recent one on the right. The color of each bar indicates the success of the build:

- Green Bar The build finished with no errors or warnings (although there may be ignored warnings).
- Yellow Bar The build was canceled.
- Red Bar An error prevented the build from finishing.

The information below the bar graph will change, depending on the bar selected (indicated by an arrow under the bar). You can click any bar in the graph to see its details.



For each bar, you can view the following details:

- **ID** The identification number of the build.
- **Branch** The name of the branch for the build.
- **Target** The name of the target for the build.
- **State** The state of the build (e.g., completed, failed, canceled).
- **Warnings** The number of warnings found while building the project.
- **Errors** The number of errors found while building the project, which prevent the build from completing.
- **By** The person who initiated the build.
- **Queued** The date and time that the build was queued.
- **Started** The date and time that the build started.
- **Finished** The date and time that the build finished.
- **Duration** The length of time it took to complete the build.

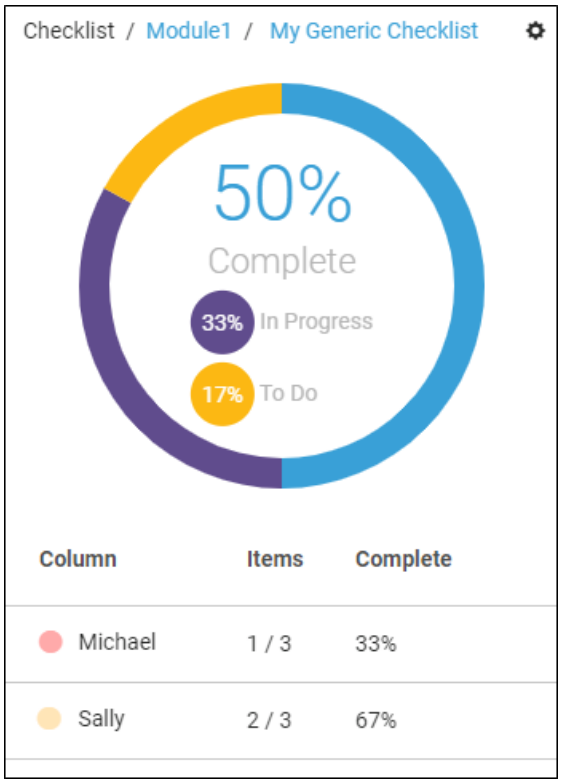
You can also use the buttons at the bottom of the widget:

- **View Log** Opens the log for the selected build so that you can see information about warnings or errors.
- **View Target** Opens the output for the selected build.
- **Download** Lets you download the output files for the selected build.

> **NOTE** Data will not display in this widget until you select a project and target, and until you generate the target in Central.

CHECKLIST

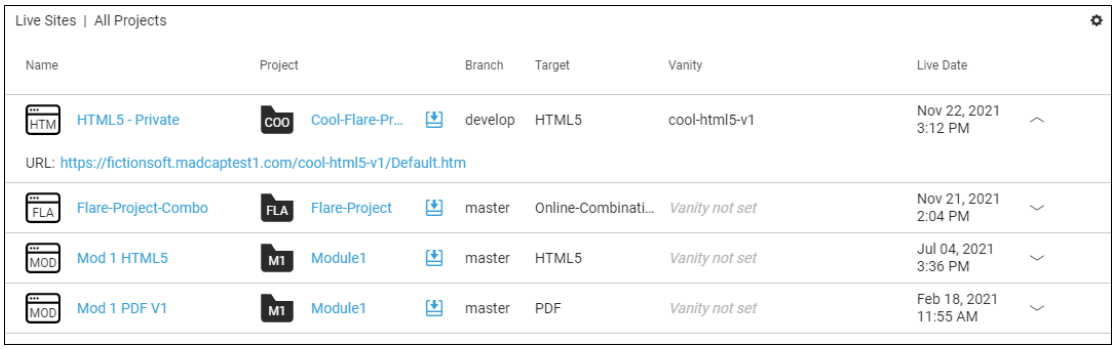
The Checklist widget displays a progress summary for a checklist. In the upper-right corner, click ⚙ to select a project and checklist for the widget.



> **NOTE** Data will not display in this widget until you select a project and checklist.

LIVE SITES

The Live Sites widget shows all of the sites that are currently set as “live.” In the upper-right corner, you can click ⚙ to filter the widget by project and the most recent number of sites to display. You can also click a row to expand that site to see the URL.



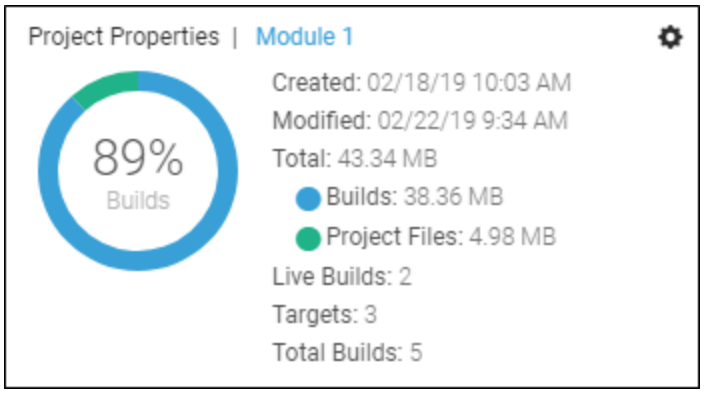
In this widget, you can view or do the following:

- **Name** The name of the site. Click the name to open the Sites page.
- **Project** The name of the project.
- **[Download Icon]** Clicking the icon downloads the output to your computer. Multi-file outputs are downloaded as a ZIP file.
- **Branch** The name of the branch built for the live site.
- **Target** The name of the target, which you can click to open the Builds page.
- **Vanity** The vanity for the live site (if one has been set).
- **Live Date** The date the site was set as “live.”

> **NOTE** Data will not display in this widget until you set a site to "live."

PROJECT PROPERTIES

The Project Properties widget shows important information about the selected project. In the upper-right corner, click ⚙ to select a project.



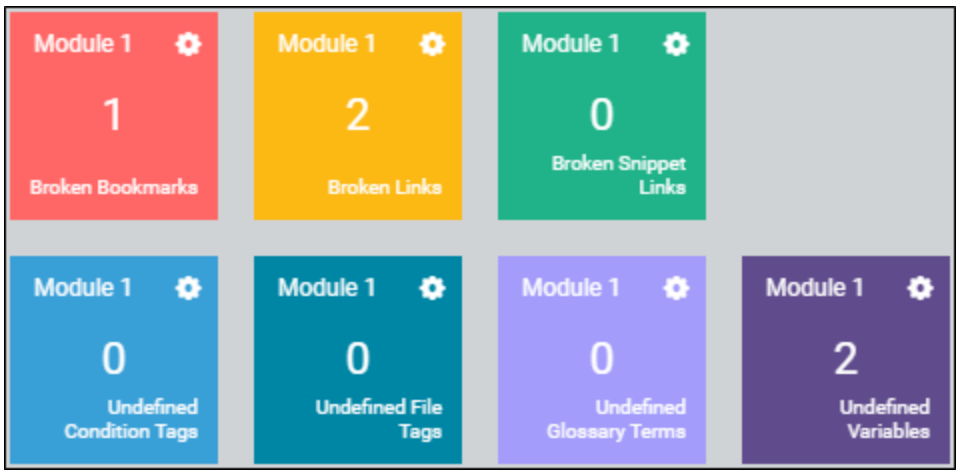
In this widget, you can view the following:

- **Created** The date the project was created (i.e., added to Central)
- **Modified** The date the project was last modified (i.e., changes were uploaded to Central, or changes made in Central)
- **Total** The total size of all the source files and all builds
  - **Builds** The size of the output files
  - **Project Files** The size of the source files of the project (Content and Project Folders)
- **Live Builds** The number of sites set to "live" for this project
- **Targets** The total number of targets in this project
- **Total Builds** The total number of builds in this project

> **NOTE** Data will not display in this widget until you select a project.

REPORTS

The Report widgets are small tiles, each representing a different report. The number on each widget represents the quantity of issues found in the latest scan for that report. You can click the number and the Reports page will open with that report automatically selected.



The following types of reports can be included in widgets:

- Broken Bookmarks These are links to bookmarks that are broken. In Flare, you need to fix the issue (e.g., remove or replace the bookmark link).
- Broken Links These are links—such as cross-references—that are broken. In Flare, you need to fix the issue (e.g., remove or replace the link).
- Broken Snippet Links These are links to snippets that are broken. In Flare, you need to fix the issue (e.g., remove or replace the snippet).
- Undefined Condition Tags These are conditions that have been applied in the project, but the name or location of those conditions cannot be found. This might occur, for example, if you have applied a condition in a topic and then later change the name of the tag or condition tag set. The old tag or set name is still used at the location where it was previously applied in the topic. In Flare, you need to fix the issue (e.g., rename the condition tag or set, remove the tag where it was applied).
- Undefined File Tags These are file tags that have been applied in the project, but the name or location of those file tags cannot be found. This might occur, for example, if you have associated a file tag with a topic and then later change the name of the tag or file tag set. The old tag or set name is still used for the topic. In Flare, you need to fix the issue (e.g., rename the file tag or set, remove the tag where it was applied).

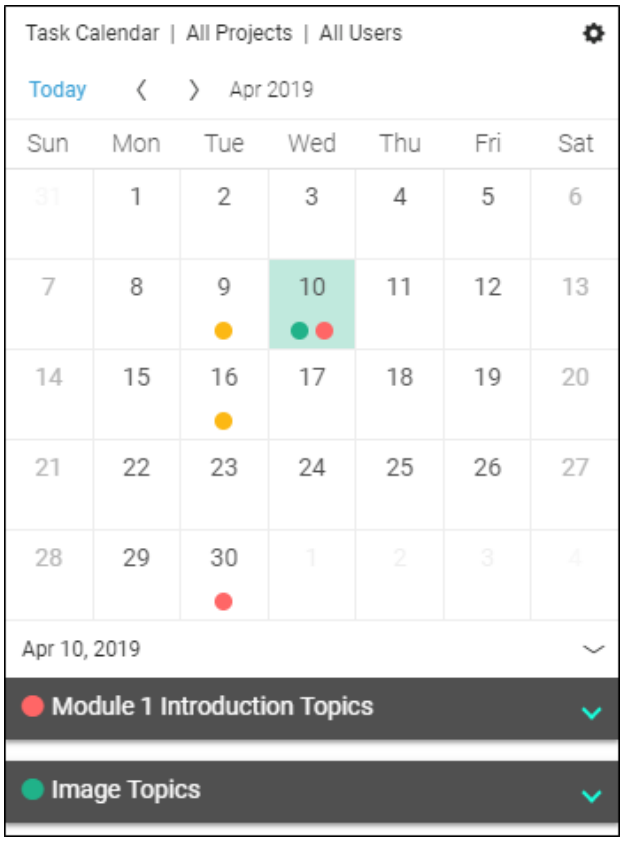
- Undefined Glossary Term Links These are glossary term links in your content that are broken. This might occur, for example, if you have created a new term in a glossary and then inserted that term into a topic as a glossary term link. Later, you remove that term from the glossary. You now have a glossary term link in a topic, but the term no longer exists in the glossary. In Flare, you need to either remove the glossary term link from the content, or you need to add the term to the glossary again.
- Undefined Variables These are variables that have been inserted in the project, but the name or location of those variables cannot be found. This might occur, for example, if you have inserted a variable in a topic and then later change the name of the variable or variable set. The old variable or set name is still used at the location where it was previously inserted in the topic. In Flare, you need to fix the issue (e.g., rename the variable or set, remove the variable where it was applied).

When you add a new Reports widget to a dashboard, it is initially empty. You need to select a project to populate it. To do this, click ⚙ in the upper-right corner and a project and report.

However, it is recommended that you open the Reports page and add widgets from there instead. The reason for this is that you can select multiple reports (rather than one at a time), and you can add them to both the Home and Project dashboard simultaneously.

TASK CALENDAR

The Task Calendar widget shows tasks that are due in a given month. In the upper-right corner, you can click ⚙ to filter the widget by project, user, and priority.

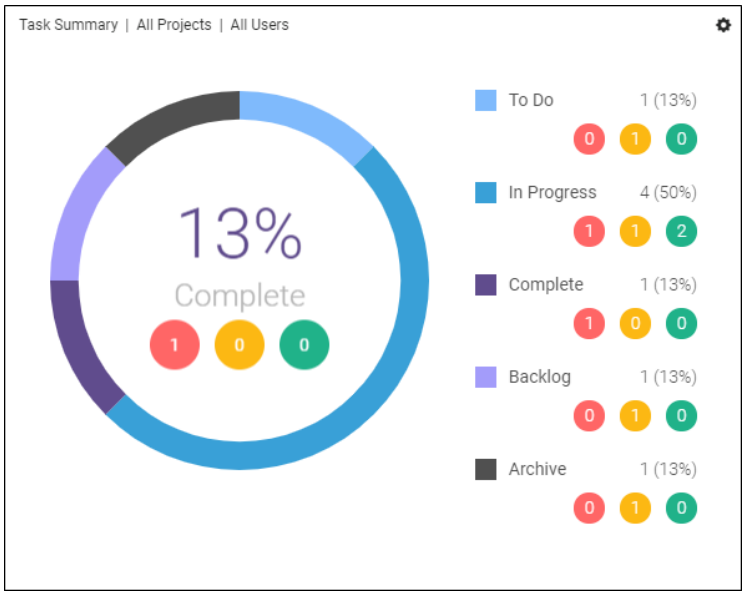


In this widget, you can view the following:

- [Calendar] Colored circles indicate tasks that are due on that date. The color of each circle represents the priority of the task due (red=high, yellow=medium, green=low). Use the arrows on the header bar to navigate between months.
- [Tasks Cards] If you click on a day in the calendar that contains colored circles, the area below has cards for those tasks that are due. Initially, the task cards are collapsed, but you can expand them to see more information. And if you click the name of the task, you can edit it and see even more details.

TASK SUMMARY

The Task Summary widget shows a doughnut chart that breaks down the percentage of tasks in each milestone and storage area (To Do, In Progress, Complete, Archive, Backlog). To the right of the doughnut chart, you can see a summary with the number of tasks at each priority level (High, Medium, Low). In the upper-right corner, you can click ⚙ to filter the widget by project and user.



4. Click **Add**. The widgets are added to the dashboard.

CHAPTER 6

Builds

If you are associated with a project and have the appropriate permissions, you can open the Builds view of a project. From here, you can generate builds and manage various aspects of the project's output.

This chapter discusses the following:

Main Activities for Builds

Some activities are particularly common and important when it comes to this feature.

- "Generating and Scheduling Builds" on the next page
- "Publishing Directly to Central" on page 109
- "Setting Builds as Keep" on page 112
- "Viewing Output" on page 114
- "Deleting Builds" on page 115

Generating and Scheduling Builds

If you are associated with uploaded Flare projects in Central, you can open the Builds view of the Projects page. From here, you can manually build targets (such as HTML5, PDF, and others), or you can schedule targets to be generated. If you are using branching, you can build output for a specific branch.

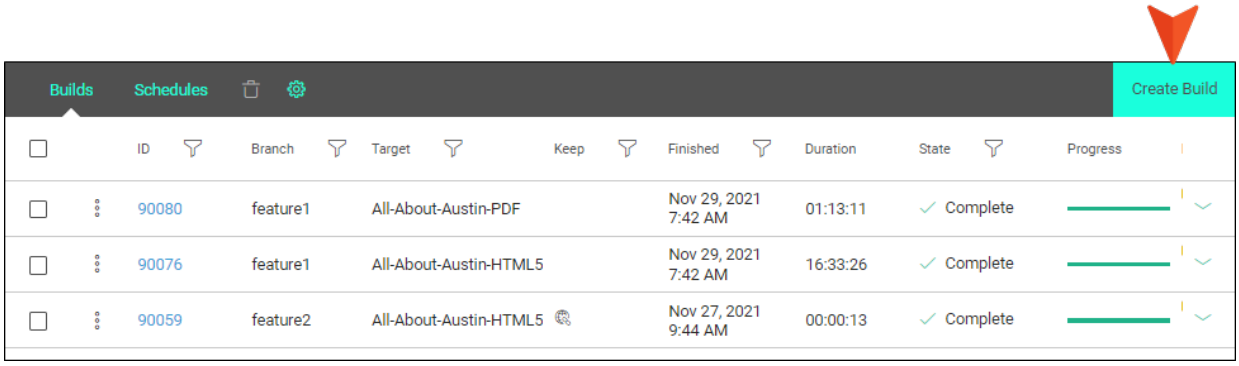
Permission Required?

For this activity, you must have the following permission setting:

For more information about permissions, see the Central online Help.

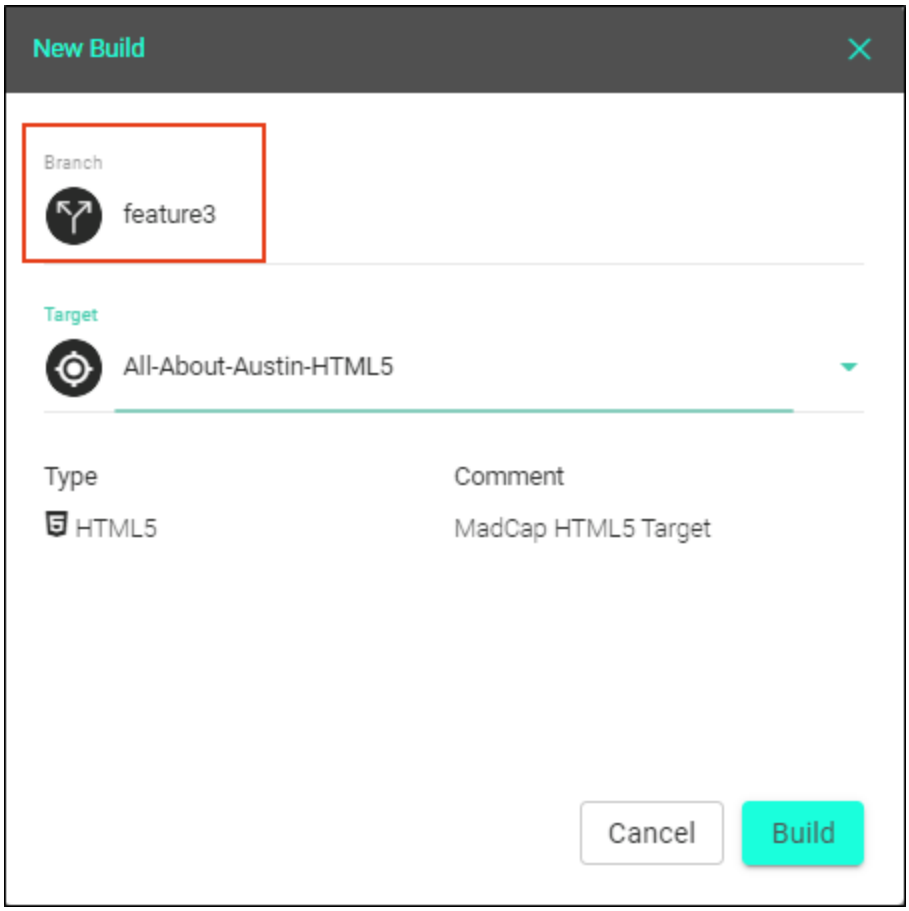
How to Generate a Target Manually

1. Select **Projects** on the left side of the interface, then click a project name to open it.
2. At the top of the page, click **Builds**.
3. In the upper-right, click **Create Build**.

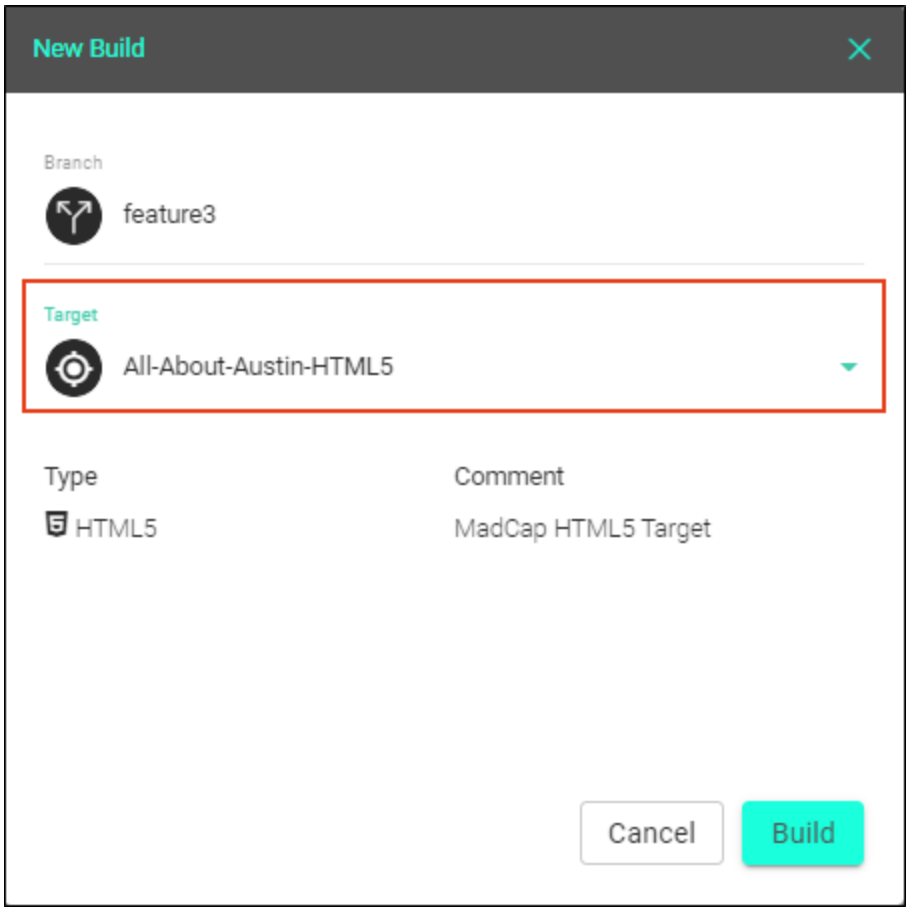


The New Build dialog opens.

4. (Optional) From the **Branch** field, you can select the appropriate branch. If you are not using branching, you will always be building from the master branch.

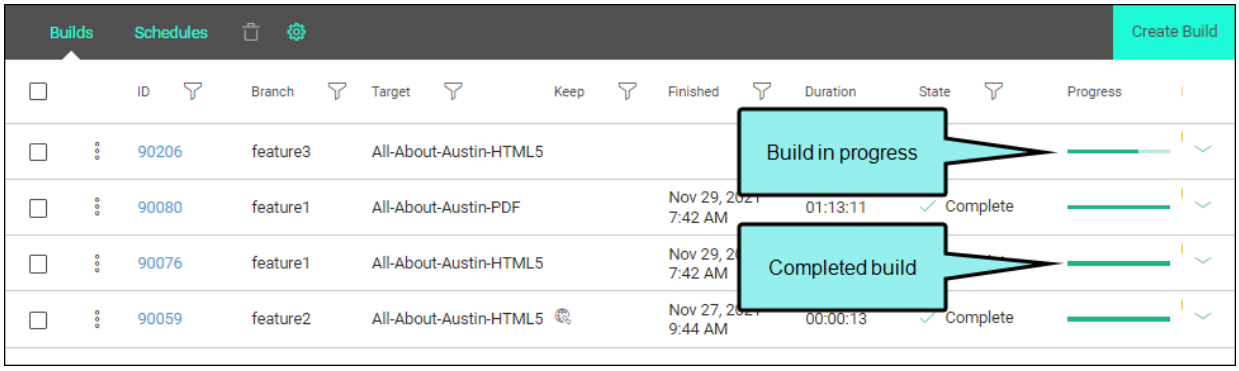


5. From the **Target** field, select the target you want to build.

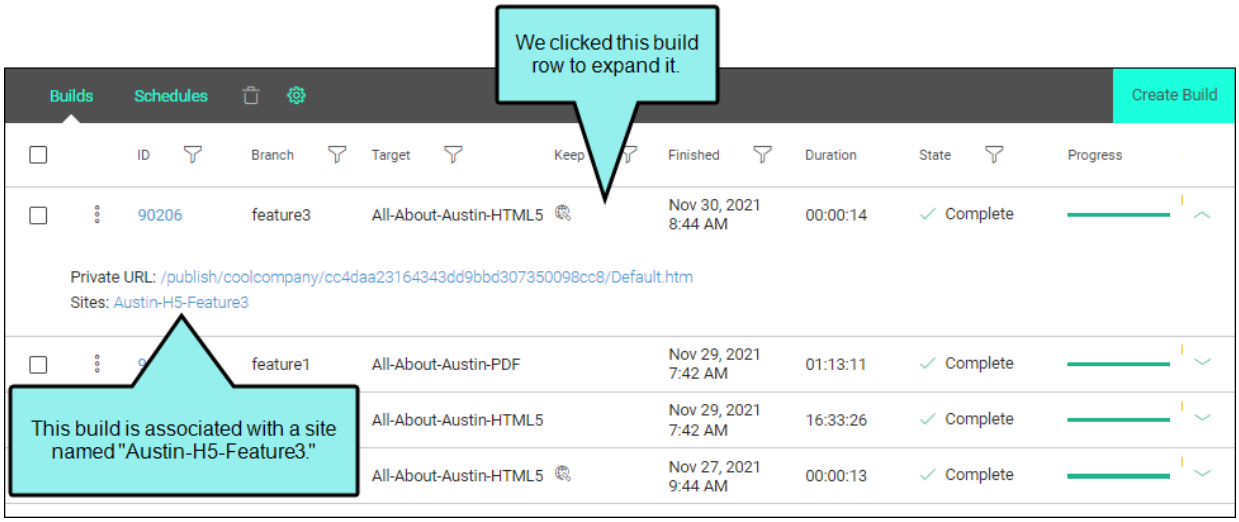


6. Click **Build**.

   A row is added to the page, showing the progress of the target as it is generated. If the target generates successfully, the color of the Progress column changes from light to dark green, and the State column displays “Complete.”



   If you click a build row (not the check box), the row expands, showing details about the selected build (i.e., the private URL and whether the build is associate with a site).



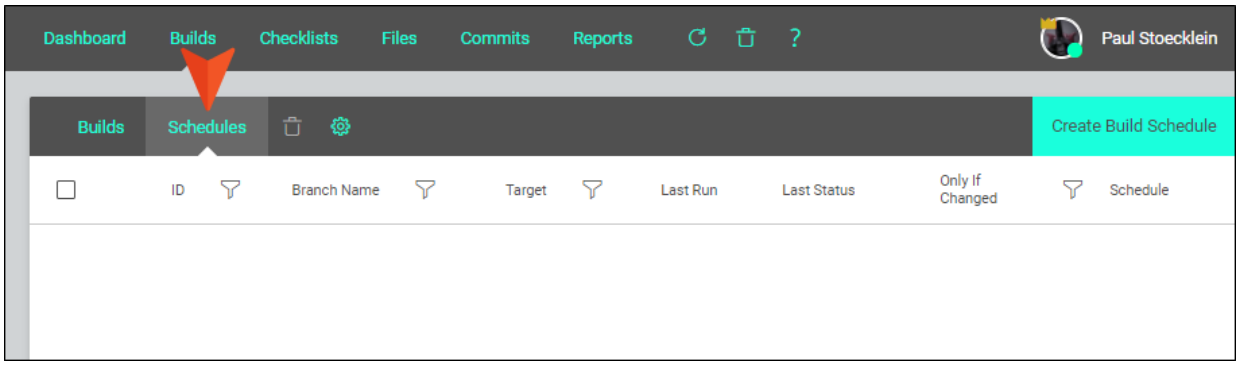
NOTE You might notice that it takes somewhat longer for a target to build on Central, compared with building it locally in Flare.

NOTE Certain Flare features are not supported when generating output from Central. This includes limitations on which output formats can be viewed from Central. See "Flare Features Not Supported in MadCap Central" on page 129.

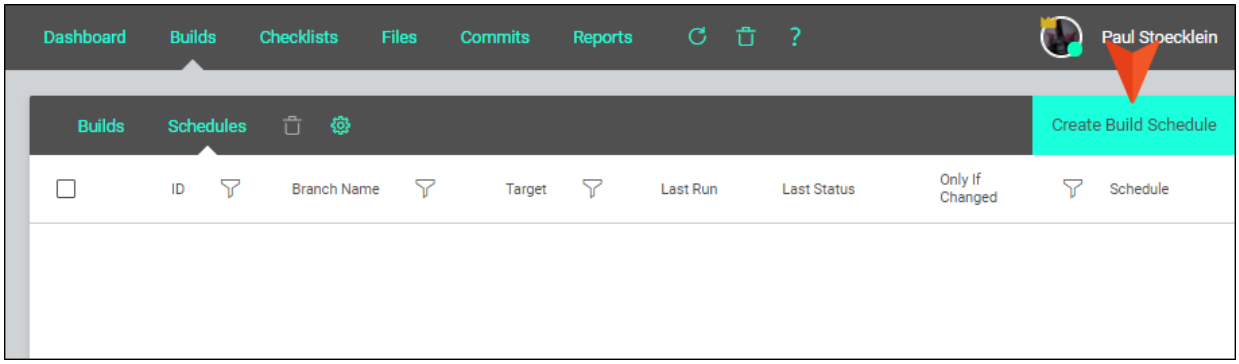
NOTE If you intend to build a target that links to an outside Flare project (e.g., for the purpose of importing files via Global Project Linking, project merging, or multilingual output), make sure that the other project is also uploaded to Central. Otherwise, the automated imports or merging will not work. See "Setting Up Project Linking" on page 68.

How to Schedule a Target to Be Generated

1. Select **Projects** on the left side of the interface, then click a project name to open it.
2. At the top of the page, click **Builds**.
3. At the top of the grid area, select **Schedules**.

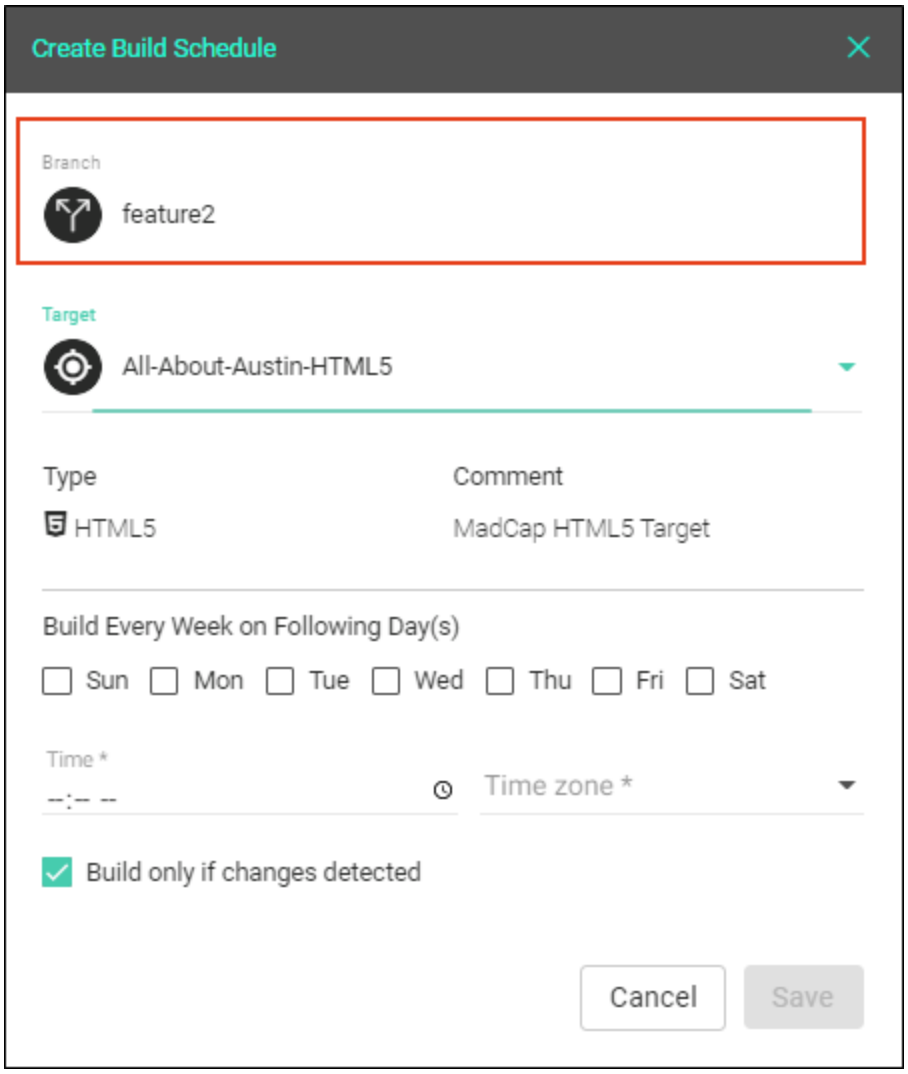


4. In the upper-right, click **Create Build Schedule**.

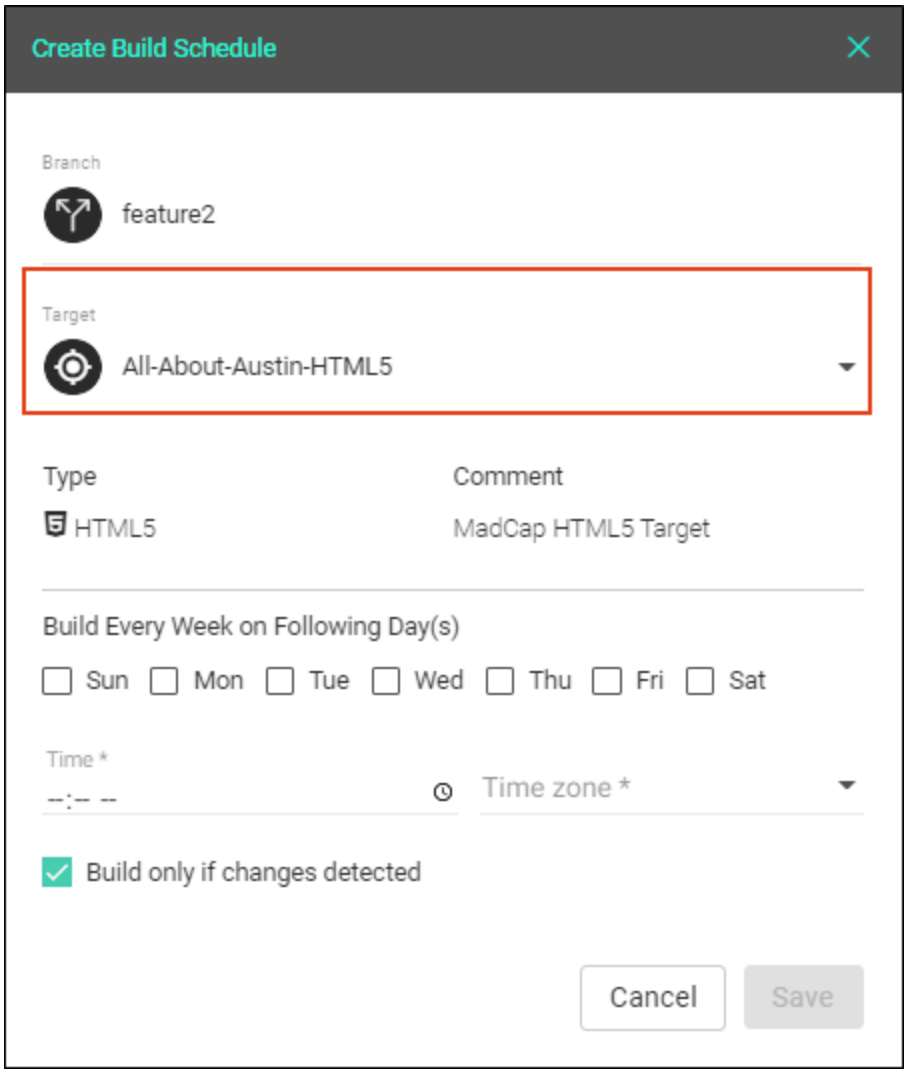


The Create Build Schedule dialog opens.

5. (Optional) From the **Branch** field, you can select the appropriate branch. If you are not using branching, you will always be building from the master branch.



6. From the **Target** field, select the target you want to build.

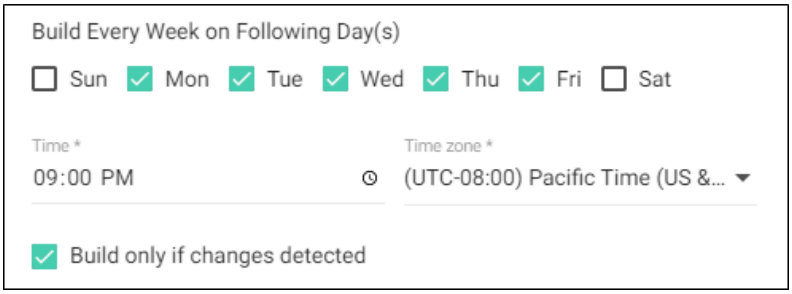


7. Complete the rest of the fields in the dialog.
   - **Build Every Week on Following Day(s)** Click on the day(s) of the week when you want the target to be generated.
   - **Time** Click the field to manually enter the hour, minutes, PM or AM. Alternatively, click the little clock to the right of the field to select a time from the columns presented in the Time Picker popup. You can use the mouse wheel to rotate through the hour and minutes columns to locate the desired time.
   - **Time zone** Click the down arrow and choose your time zone.
   - **Build only if changes detected** Select this option to generate output on the specified days and times only if no changes to source files were found. If you deselect this option, the target will always be generated, even if no changes are detected in the source files.

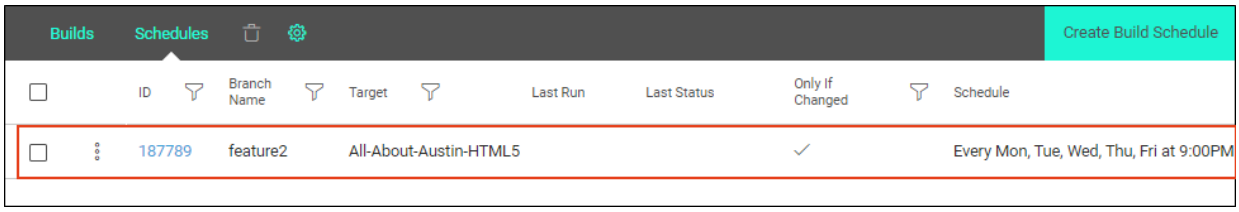
☆ EXAMPLE You want to schedule a target to be generated automatically. You want the target to be build every work day (Monday-Friday) at 9 p.m. Pacific time.

However, you only want the target to be generated if a change is detected in the source files since the previous build.

Your schedule would look something like this:

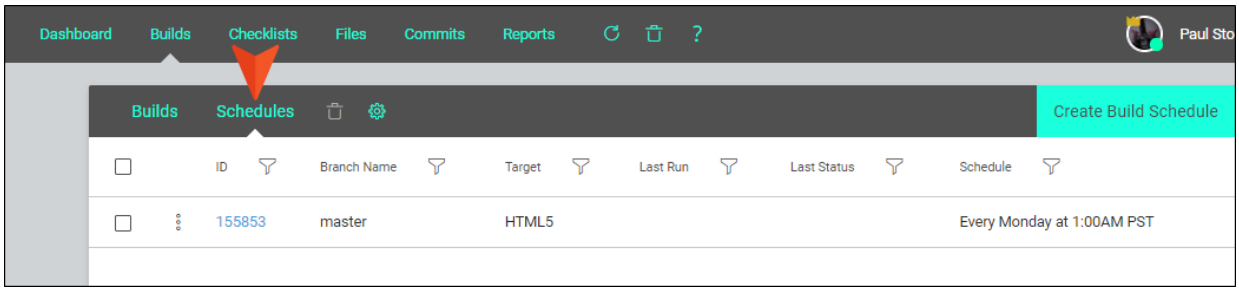


8. Click **Save**. A row is added to the grid, representing that scheduled build.

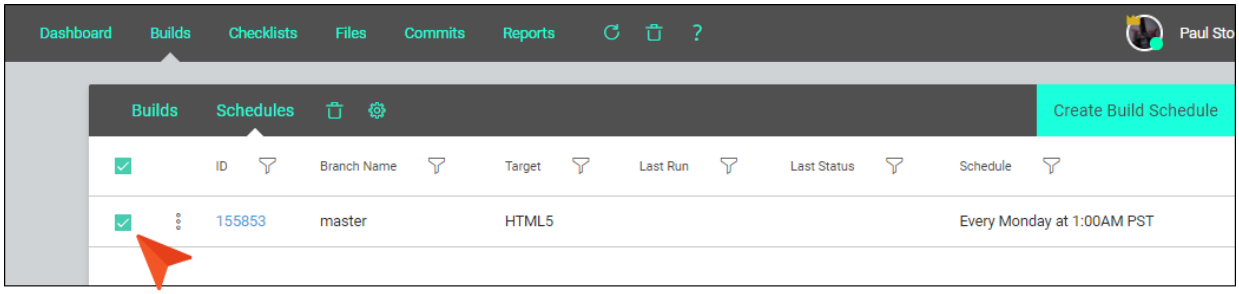


How to Remove a Scheduled Build From a Target

1. Select **Projects** on the left side of the interface, then click a project name to open it.
2. At the top of the page, click **Builds**.
3. At the top of the grid area, select **Schedules**.

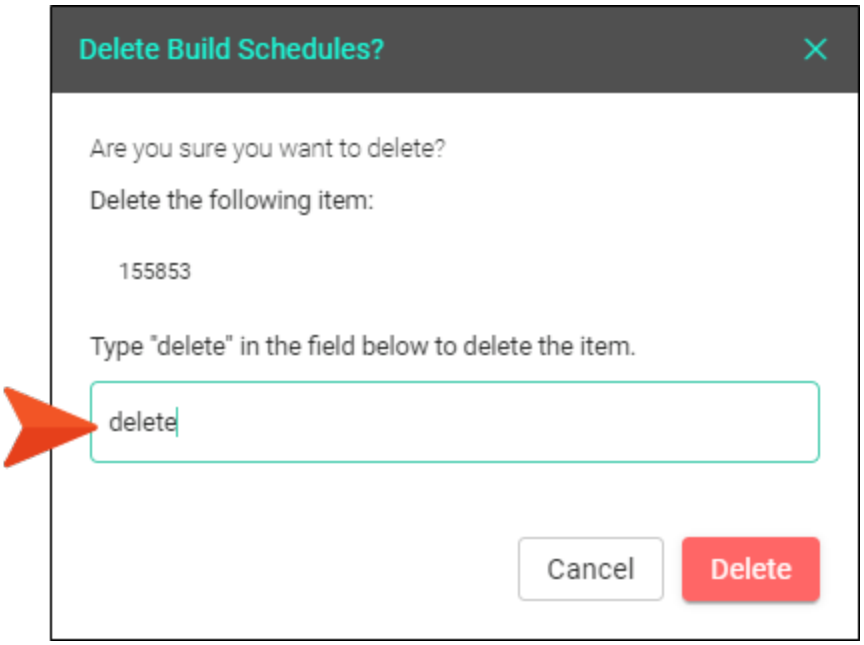


4. In the grid, select the check box next to the scheduled build you want to remove.



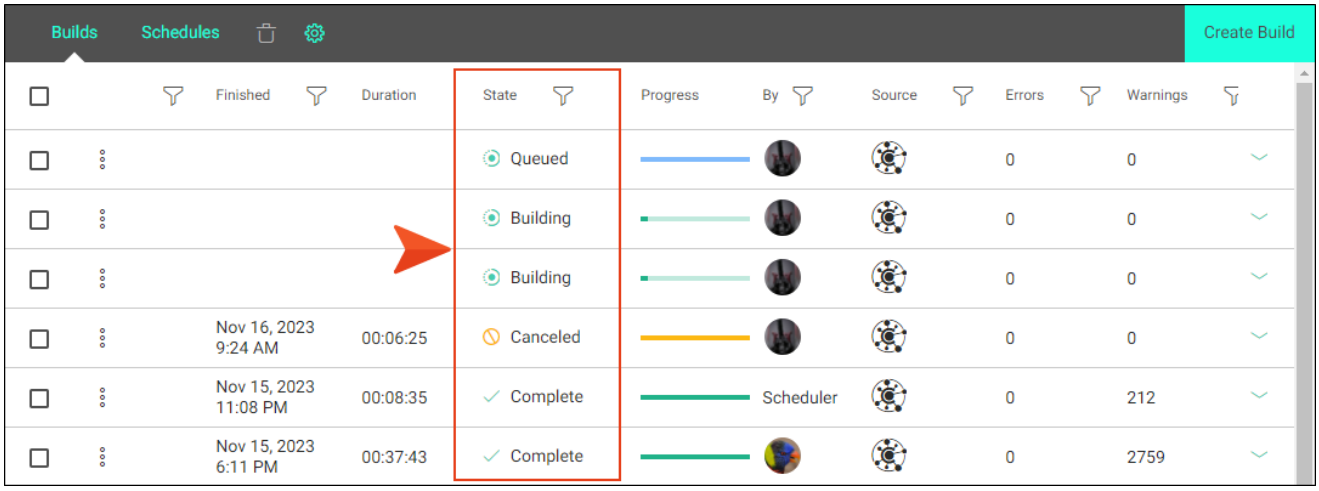
5. In the local toolbar, click .

6. In the dialog that opens, type `delete` in the field, and click **Delete**.



Build State

When you build a target in Central, it is going to have a particular state, depending on various factors.



Queued

When you generate a target in Central, your build is added to a node along with builds from other customers. Therefore, it is queued as it awaits its turn to begin processing. If one node reaches is limit, another one is started in order to process additional builds. Therefore, the wait should be brief for any build that you start.

> TIP If you notice that builds are constantly displaying as queued, you can try generating targets at a different, non-peak time of day. Also, if you have many builds that are scheduled to begin at the same time, you can try staggering those start times.

> NOTE If a build seems to be stuck in the queued state for quite a long time, please contact technical support.

Getting Files

When a queued build reaches its turn, Central undergoes a brief period where it gathers the necessary files to generate the output.

Building

The "Building" state for a target means that the output is actively being processed. A build can remain in this state for quite some time, depending on the number of files and size of the output.

Publishing

The "Publishing" state for a target means that it has finished compiling and is in the final stages before completion.

Completed

A build reaches the "Completed" state if it generates successfully. You can then expand the build row and click the link to view the output.

Canceled

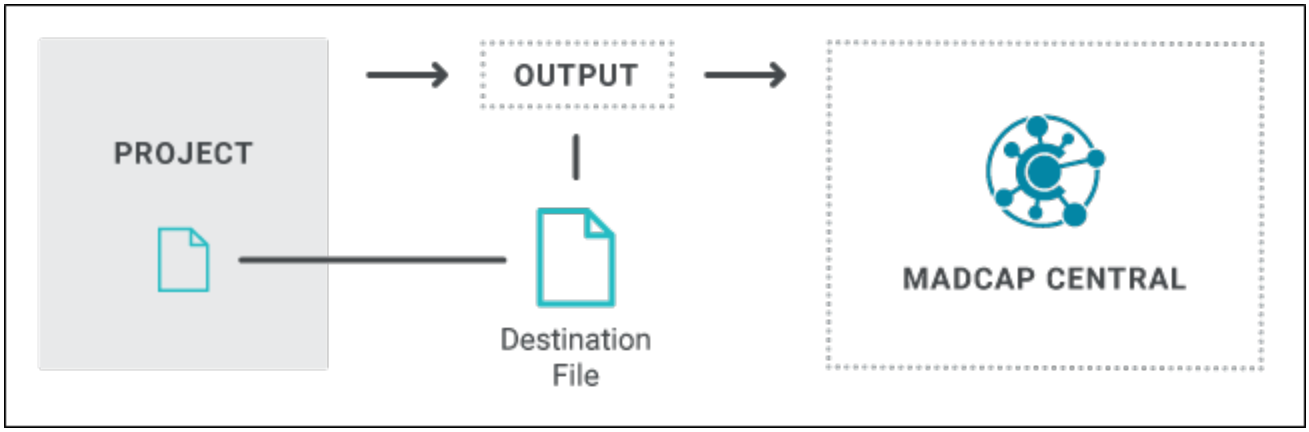
This state displays when someone cancels a build.

Failed

A build might fail for various reasons. If this happens, you can click the three dots at the left of the row and select **View Log** from the menu. This can give you an indication as to why the build was not successful. See "Opening the Build Log" on page 125.

Publishing Directly to Central

As an alternative to building a target on MadCap Central, you can publish output directly to Central from the local Flare project. This is possible via a destination file that Flare automatically creates for you. By “publish,” we mean copying your output files to Central, not making that output “live,” which would make it visible to the general public. You would still need to use Central to make that output accessible to end users.

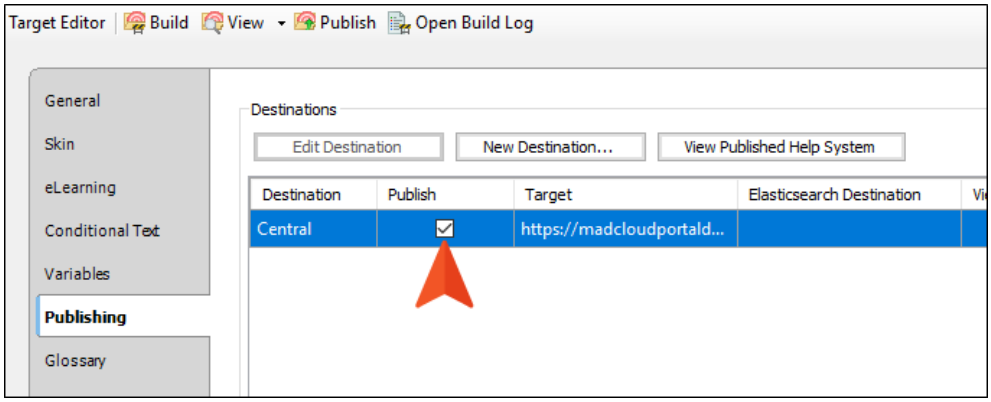


How to Publish Directly to Central

1. If it is not already done, upload a Flare project to Central.
2. In the Flare project, open a target.

> NOTE If you are using Git branching, make sure the appropriate branch is active. When you build and publish the output, it will be for that branch.

3. Select the **Publishing** tab.
4. Next to the **Central** destination, click the **Publish** check box.

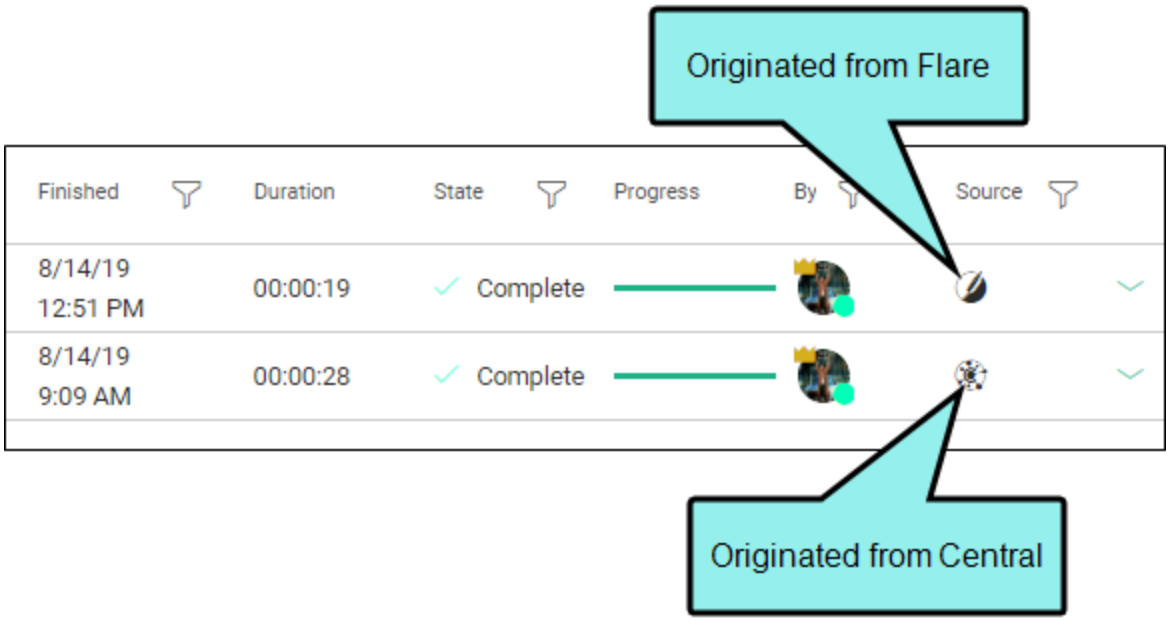


> NOTE This destination is automatically shown on the tab if the project is bound to Central. If it is not bound to Central, you will not see this destination.

5. Click to save your work.
6. At the top of the editor, click **Build**.

7. When the build finishes successfully, click **Publish**.

   After the target publishes, you can see the result in Central. A Flare icon displays next to builds that originated from Flare. A Central icon displays next to builds that originated from Central. Also, if you had a particular Git branch selected when publishing, it will be shown in the Branch column of the grid.



> **NOTE** After the project is bound to Central (along with any targets you plan to publish), it is not technically necessary to synchronize changes with Central to have the latest files included in a build that is initiated locally. However, you would need to continue synchronizing in order to see the latest files on Central when using other features (e.g., checklists, builds initiated from Central).

Setting Builds as Keep

You can mark successful builds as “keep.” You might do this to avoid accidentally deleting it, perhaps with the intention of rolling back to it at some point (i.e., setting the site associated with that build to “live”).

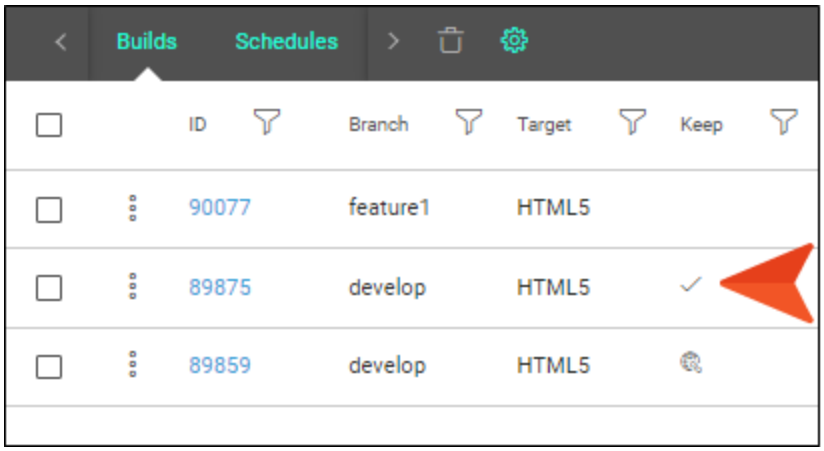
Permission Required?

For this activity, you must have the following permission setting:

For more information about permissions, see the Central online Help.

How to Set a Build as Keep

1. Select **Projects** on the left side of the interface, then click a project name to open it.
2. At the top of the page, click **Builds**.
3. In the grid, click the three dots to the left of the build that you want to set to “keep.”
4. From the menu select **Toggle Keep**.
5. In the confirmation dialog, click **OK**. An icon is shown next to the build.



What’s Noteworthy?

NOTE To remove the “keep” status of a build, select the **Toggle Keep** option again.

Viewing Output

If you are associated with a project, you can view the generated output for a target.

Permission Required?

For this activity, you must have the following permission setting:

For more information about permissions, see the Central online Help.

How to View Output

1. Select **Projects** on the left side of the interface, then click a project name to open it.
2. At the top of the page, click **Builds**.
3. In the grid, click the check box to the left of the build you want to view.
4. In the Builds toolbar, click . This button is not enabled unless you select just one build.

What's Noteworthy?

NOTE Only some output formats can be viewed immediately from Central (see "Flare Features Not Supported in MadCap Central" on page 129). If you generate an output format that cannot be viewed, you can download it to your computer, where you can view it (see "Downloading Builds" on page 127).

Deleting Builds

If you generate builds that you do not plan to keep or publish, you can delete them. This frees up space on the system.

Permission Required?

For this activity, you must have the following permission setting:

For more information about permissions, see the Central online Help.

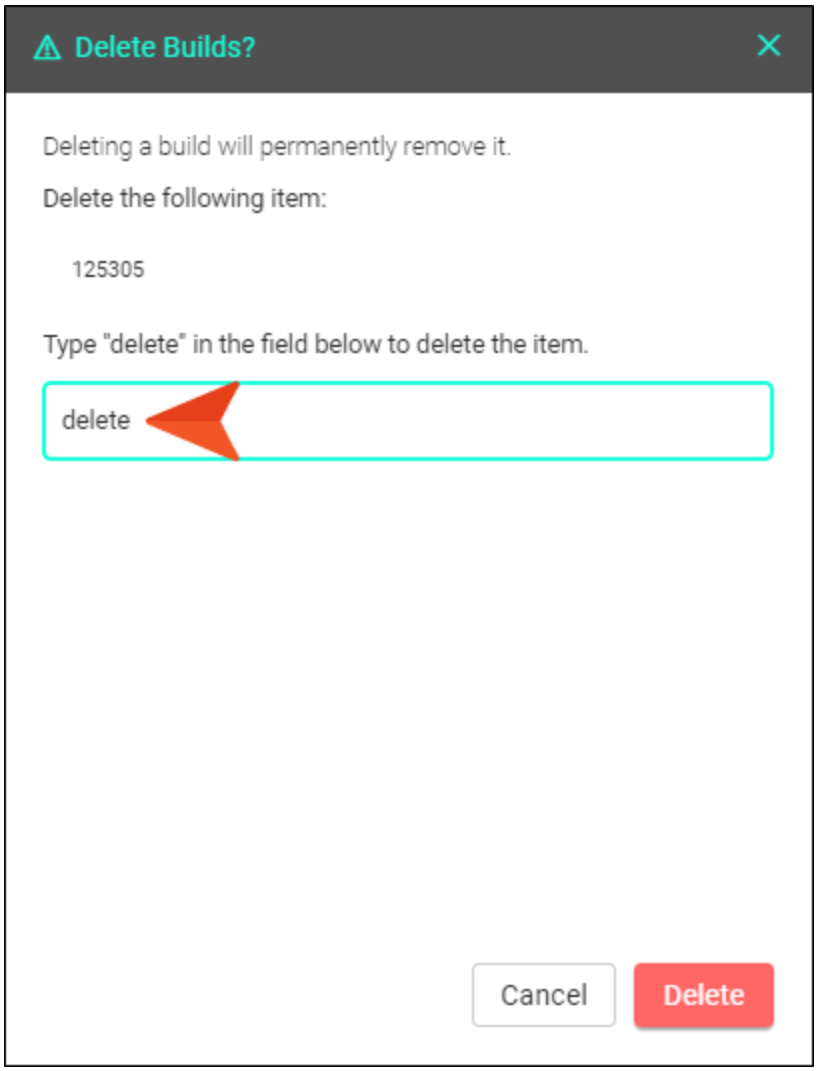
How to Delete Builds

1. Select **Projects** on the left side of the interface, then click a project name to open it.
2. At the top of the page, click **Builds**.
3. In the grid, click the check box to the left of each build you want to delete. Or click the top check box to select all builds.

   > NOTE You cannot delete builds that are set to “keep" or associated with a site.

4. In the toolbar, click .

5. In the dialog, type `delete` in the field under the listed builds.



6. Click **Delete**.

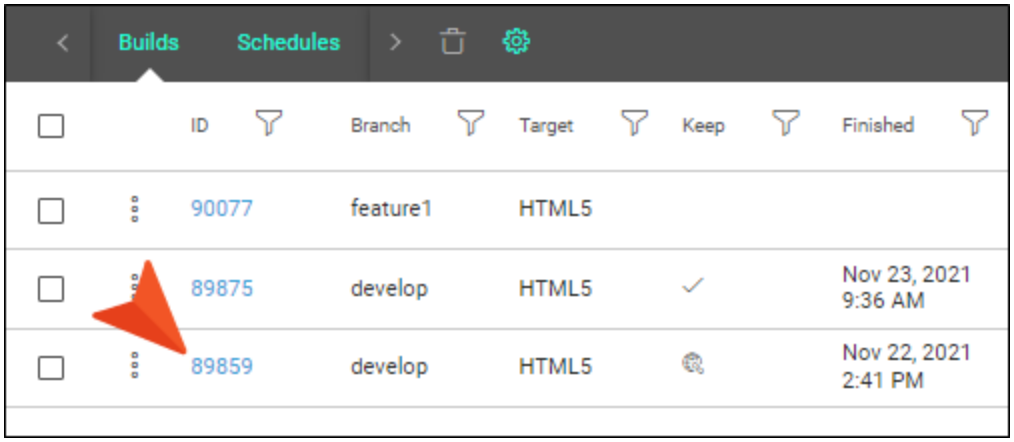
> **NOTE** You can also delete a build from the Build Profile dialog. See "Opening the Build Profile Dialog" on page 118.

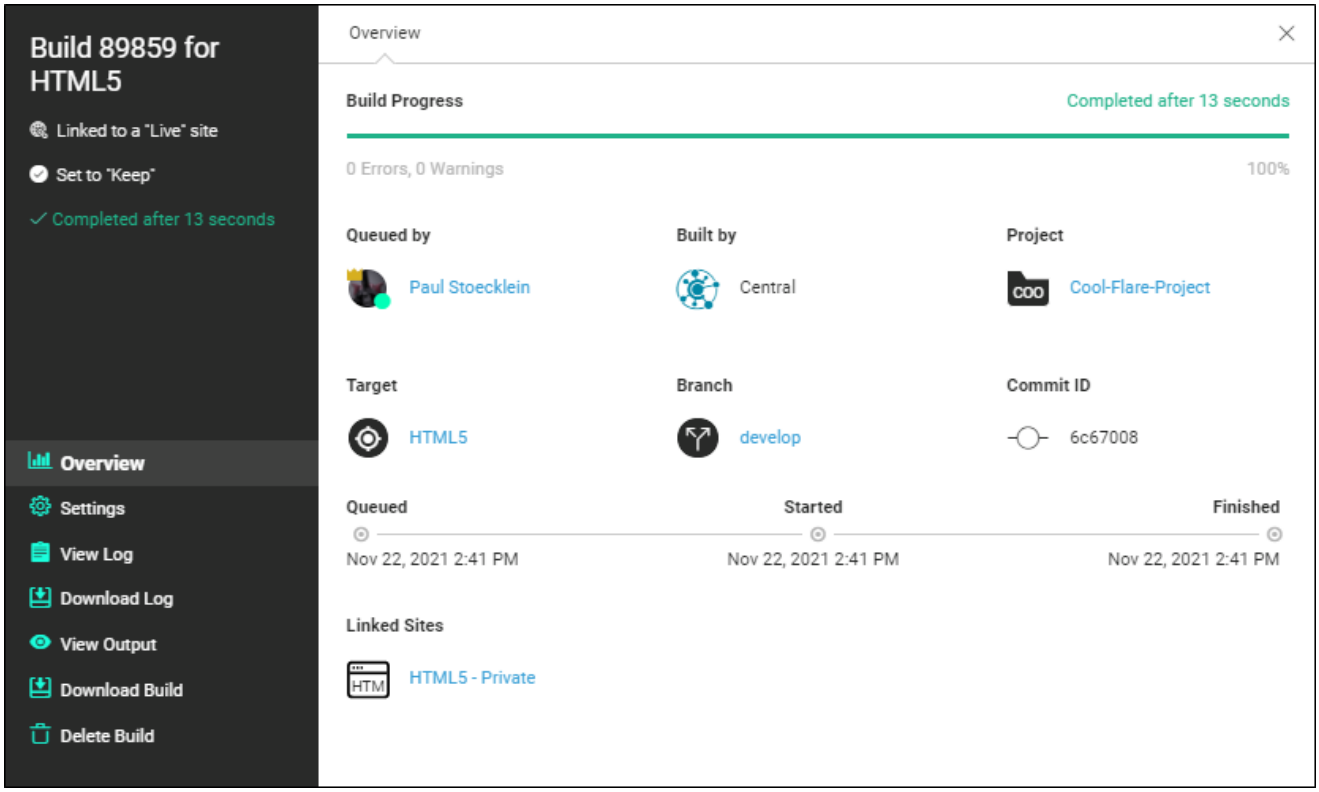
Other Activities and Information for Builds

In addition to the main activities, there are some other tasks you might perform regarding this feature.

Opening the Build Profile Dialog

On the Builds page, a column displays an auto-generated ID for each build. When you click this ID, the Build Profile dialog opens. This dialog lets you see detailed information about a particular build.

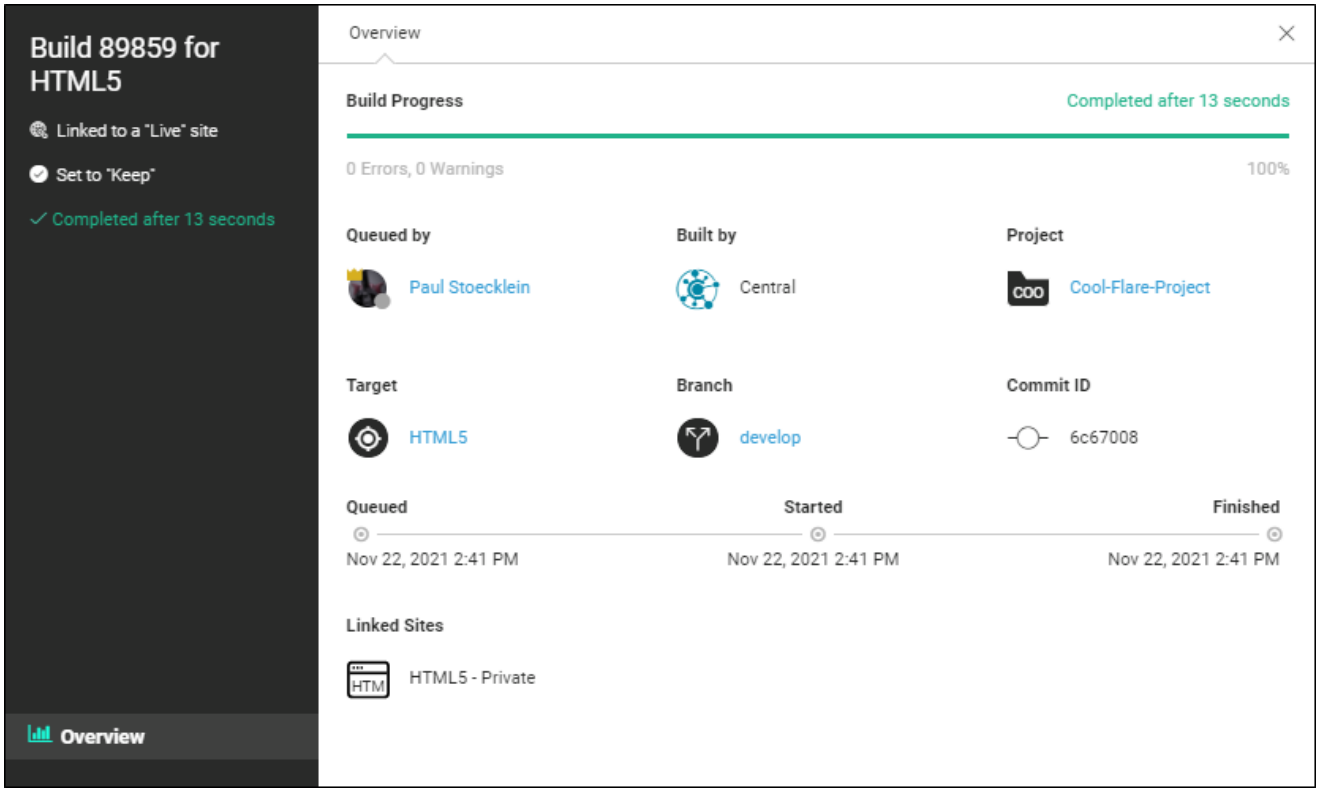




Permission Required?

To open the project in Central and access the Build Profile dialog, you must be associated with the project. However, depending on your permissions, you might see different section links on the left side of the dialog.

If you do not have any permissions for builds, the dialog will look like this (with only the Overview section):



Sections

Following are the possible sections that you might see. You can use these sections to view different information and perform various activities (as an alternative to using toolbar buttons above the Builds grid).

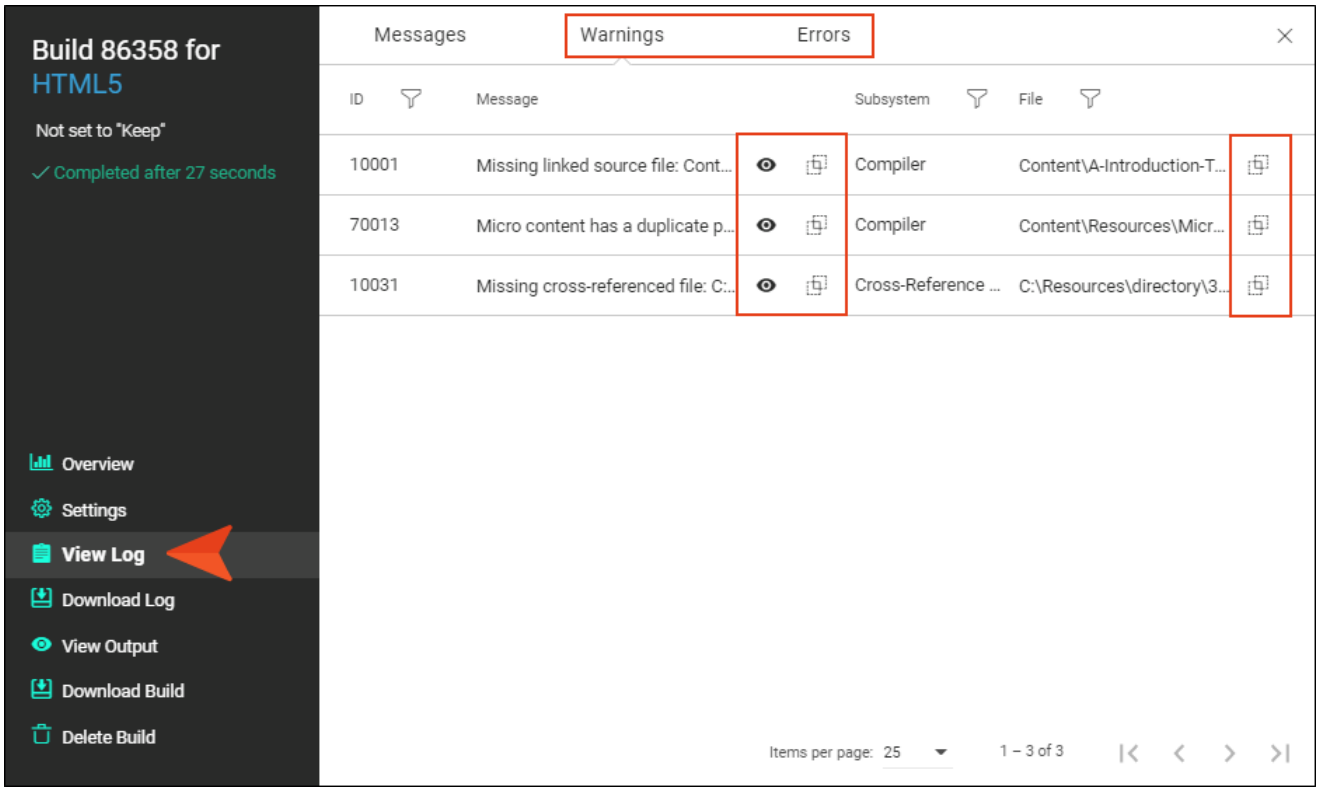
- Overview Displays the following information:
  - Build timeline and progress, including its level of success
  - Individual who generated it
  - Project
  - Target
  - Branch
  - Commit ID
  - Linked sites
- Settings Lets you mark successful builds as "keep," or remove this setting. You might set a build to "keep" to avoid accidentally deleting it, perhaps with the intention of rolling back to it at some point (i.e., setting the site associated with that build to "live"). You cannot change a build's "keep" state while it is linked to a "live" site.
- View Log Displays the log for the build, with tabs showing messages, warnings, and errors.
- Download Log Downloads the log (.mclog) file to your computer. You can then open it in a text editor such as Notepad++.
- View Output Opens the output for the build.
- Download Build Downloads the output files for the build into a ZIP file.
- Delete Build Lets you remove the build from the grid permanently. You cannot delete builds that are set to "keep" or associated with a site.

Build Log Show/Copy Options

If you open the log for a build, you might notice the following options on the Warnings and Errors tabs.

Opens a popup to show the complete message.

Copies the message or file path to the clipboard so that you can paste it anywhere you like.



Using Private URLs

For each build that you produce, a private URL is generated. This URL is useful for easily sharing generated output with individuals in your organization (e.g., for reviews and knowledge transfer). Only users registered as authors on your Central license can access this URL.

Private URLs are not to be confused with private outputs.

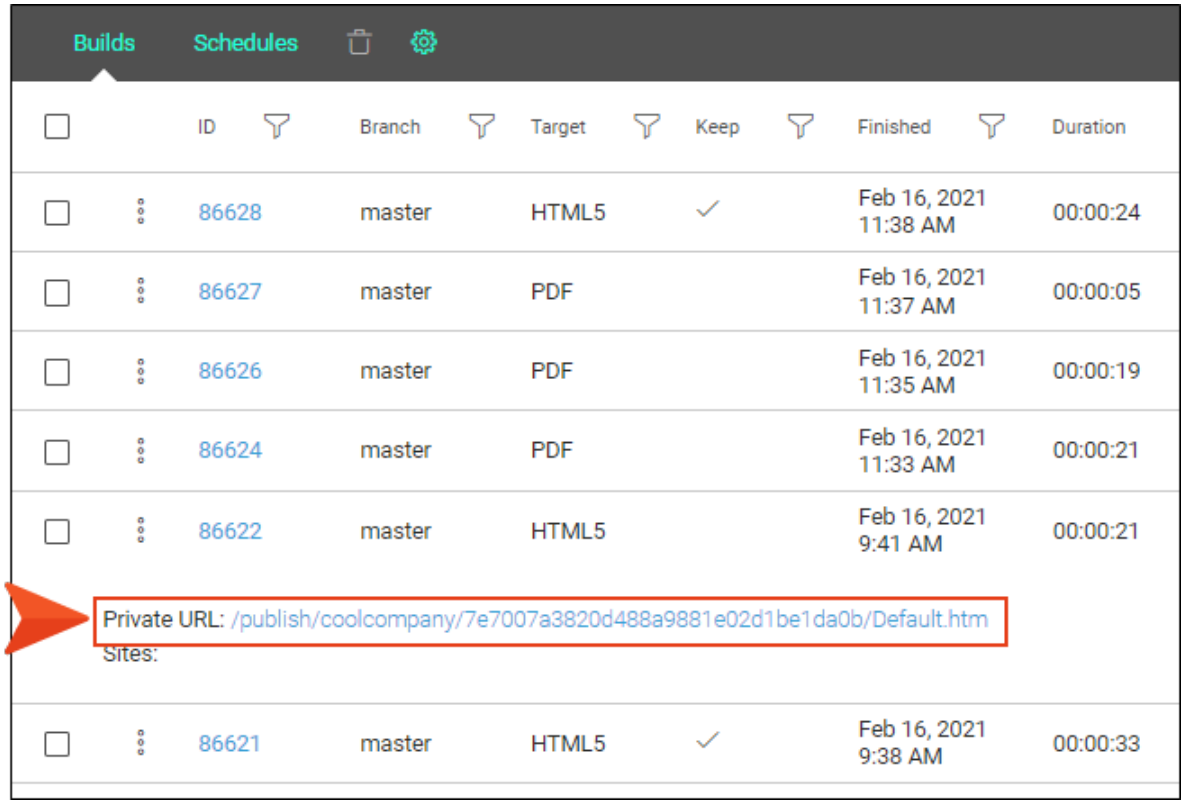
- Private URLs Only users with an author seat can view a private URL.
- Private Output Users do not always need to have a paid seat on Central (depending on whether you set the site as "live"). On a site that is both live and private, users can simply be customers of your company using the viewer user type (as well as authors and SMEs on your license).

Permission Required?

No special permission is required for this activity. All authors who are associated with the project are allowed.

How to Use a Private URL

1. Select **Projects** on the left side of the interface, then click a project name to open it.
2. At the top of the page, click **Builds**.
3. In the grid, click the build row (not the check box) that has the private URL you want to retrieve. The row expands, showing additional details.
4. From the **Private URL** field, copy the URL.



5. Provide this URL to authors on your Central license.

TIP If you create a site and set a site vanity, you can select the project and target for the output that you want to make available to other Central user's privately. When doing this, select to use the **Latest Build** instead of a specific build.

The site vanity can be used in place of the auto-generated ID that is normally part of a private URL. Anyone referencing your private URL will always see the most recent build for that target, instead of having to use a different private URL each time you generate a new build.

For example, you might have a private URL for a build, and the end of that URL looks something like this:

…/fictionsoft/cfb33266bffa45048d636878f00b1317/Default.htm

If you have a site vanity called "module1," you could replace the auto-generated ID in the private URL. That way, you could give others a URL that looks more like this:

…/fictionsoft/module1/Default.htm

If anyone uses this second URL that contains the site vanity, the most recent build in the target will always open in the browser.

NOTE If a particular site is set as "live," users do not need to be logged in to Central in order to view the associated build's private URL.

Opening the Build Log

Anytime you build a target, a log is automatically created. After opening a build log file, you can see more details of any warnings or errors you may have gotten when building the target.

In Flare, the output files and build log are retained for only the most recent build for a target; when you rebuild, the previous output files and build log are replaced with the new ones. But in Central, the output files and build log are retained for each build.

How to Open a Build Log

1. Select **Projects** on the left side of the interface, then click a project name to open it.
2. At the top of the page, click **Builds**.
3. In the grid, click the check box to the left of the build with the log you want to view.
4. In the toolbar, click . The Build Log dialog opens.

Viewing and Downloading the Log File

The log file contains the following sections, which can be accessed by clicking them on the left:

- Messages This shows compiler messages and warning text. It is essentially a running log of what is happening as the output is generated.
- Warnings This shows warnings reported by the compiler. Warnings do not prevent the output from being generated. Instead they might be issues that you need to fix (e.g., missing files) or suggestions for improving your output (e.g., accessibility settings). If you do not want to see a particular type of warning in future builds, you can open the Target Editor in the local Flare project and use the Warnings tab to choose the kinds of warnings to ignore. After this, be sure to push your changes to Central.

  IMPORTANT Use caution when ignoring compiler warnings. Only ignore warnings that you are certain you do not need to receive.

  NOTE When you generate a target from Flare, you have the option of ignoring warnings globally (using the Options dialog) or only for a specific target (using the Target Editor). However, when generating a target from Central, you must set the warnings to ignore in the Target Editor; Central will not use any global settings from the Options dialog in Flare.

- Errors This shows errors, which are critical problems that stop the project from building output.

If you want a local copy of the log, click Download Log.

What's Noteworthy?

NOTE You can also view and download a build log from the Build Profile dialog. See "Opening the Build Profile Dialog" on page 118.

Downloading Builds

After generating a target, you can download the output to your computer. This is an optional step, in case you want to have a local copy of the build (e.g., you want to view your Microsoft HTML Help output, which is an output format that you cannot view from Central).

Permission Required?

For this activity, you must have the following permission setting:

For more information about permissions, see the Central online Help.

How to Download a Build

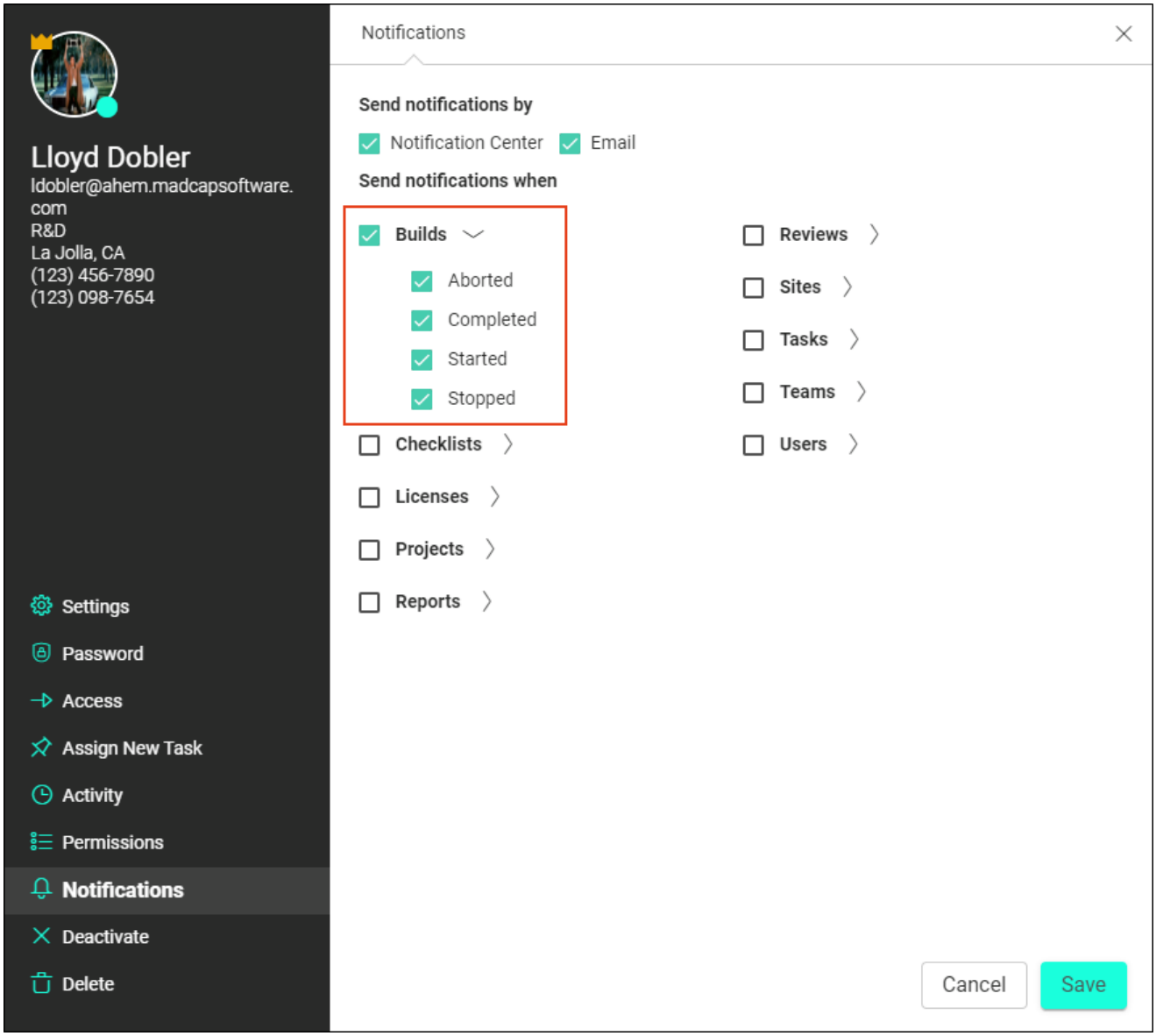
1. Select **Projects** on the left side of the interface, then click a project name to open it.
2. At the top of the page, click **Builds**.
3. In the grid, click the check box to the left of the build that you want to download.
4. In the toolbar, click . The output files are placed in a numbered zip file and downloaded to your Downloads folder.

> NOTE Due to issues with Microsoft HTML Help, the CHM file that is downloaded will initially be blocked. To unblock the CHM file, remove it from the ZIP file. Then right-click the CHM file and select **Properties**. On the **General** tab in the dialog, select **Unblock**.

> NOTE If you attempt to open a build and Central cannot open the file in the web browser (e.g. a CHM file), Central will download the file.

Setting Notifications for Builds

In your account settings, you can choose when to be notified for specific activities. The user who is notified depends on the activity. For more information, see the online Help.



Flare Features Not Supported in MadCap Central

There are certain features in Flare that are not fully supported when you generate targets from MadCap Central. However, a few of these features are supported when publishing directly to Central from a local Flare project. See "Publishing Directly to Central" on page 109.

Automatically Get Latest Files

If you are working in a dual-bound scenario and you select the option in the target to automatically get the latest version of files before generating, this option will not work when you build from MadCap Central. Instead, you must first synchronize your files with source control in Flare and then push those changes to Central.

Batch Targets

Batch targets cannot be generated from MadCap Central.

However, you can generate a batch target when publishing directly to Central from a local Flare project.

Custom Build Log Folder

In the Options dialog, Flare has a feature that lets you choose a custom folder to hold the log file after generating output. This feature is not supported when building from MadCap Central. Instead, the log file is stored at the root level of the output folder.

Custom Fonts

If you are using custom fonts in your Flare project, they are not supported when you build output from MadCap Central.

Custom Output Folder

Flare targets let you choose a custom location for your output folder. However, this is not supported when building from MadCap Central. You must use the default location.

Date/Time Variables

Target generation occurs on a server with Central, so date/time variables are not supported.

Elasticsearch

Elasticsearch is one of the search engine options that you have in Flare (along with MadCap Search and Google Search). Currently, using Elasticsearch in your HTML5 output is not supported in targets generated from Central.

However, Elasticsearch is supported when publishing directly to Central from a local Flare project.

Global Ignored Warnings

You can tell Flare to ignore certain warnings so that they do not display as such in the Builds window pane when generating output. This can be done globally for all targets in all projects. You can also override those global settings for a specific target in a project. However, if you generate output from Central, global settings to ignore warnings are not supported; instead, you must ignore warnings in the target.

Linked Mimic Movies and Projects

When building from MadCap Central, external links to Mimic movies and projects will not work. However, embedded movies in a Flare project are supported.

MadCap Pulse

MadCap Pulse is not supported in output that is generated from Central. In order to use Pulse with your output, it must be hosted outside of Central.

Pre- and Post-Build Events

If you create pre- or post-build events in your Flare target, they will not function when building from MadCap Central.

However, pre- and post-build events are supported when publishing directly to Central from a local Flare project.

Publishing to Outside Servers

Flare lets you create destination files for the purpose of publishing to external servers. This feature is not supported when building from MadCap Central. Instead, live builds are always published in Central.

Viewing Some Output Formats

The following types of output can be generated from MadCap Central, but they cannot be viewed (output files are downloaded to your computer):

- DITA
- Eclipse Help
- EPUB
- Microsoft HTML Help

  > NOTE Due to issues with Microsoft HTML Help, the CHM file that is downloaded will initially be blocked. To unblock the CHM file, remove it from the ZIP file. Then right-click the CHM file and select **Properties**. On the **General** tab in the dialog, select **Unblock**.

- Microsoft Word
- Server-Based Outputs:
  - HTML5 Plus
  - WebHelp Plus

  > NOTE If you want to produce server-based output from MadCap Central, you will need to download the output from Central after it finishes building. You can then publish the output to your server.

CHAPTER 7

Checklists

You can create checklists to keep track of work related to your uploaded Flare projects.

Checklists might have to do with specific files (e.g., topics) in your project. These are called “Project Files” checklists. You can create custom columns for whatever types of activity you want to track for each file, and you can use a note column for specific information about each row.

Alternatively, you might create checklists for random things you need to accomplish, such as a product release “To Do” list. These are called “Generic” checklists, and they let you manually name each column and row.

You can set the appropriate status on each item as you work. At the top of the interface you can select to show charts and percentages as you progress through the checklist.

This chapter discusses the following:

Main Activities for Checklists

Some activities are particularly common and important when it comes to this feature.

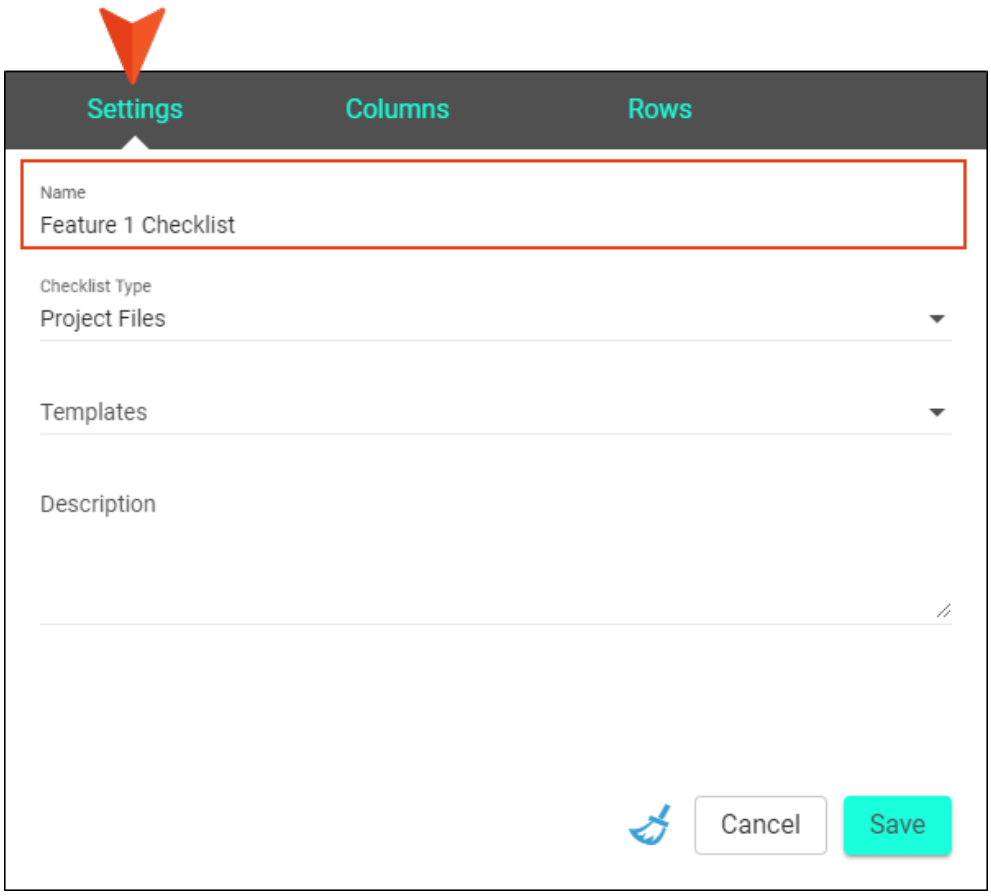
- "Creating Checklists" on the next page
- "Editing Checklists" on page 160
- "Deleting Checklists" on page 161
- "Setting Notifications for Checklists" on page 162

Creating Checklists

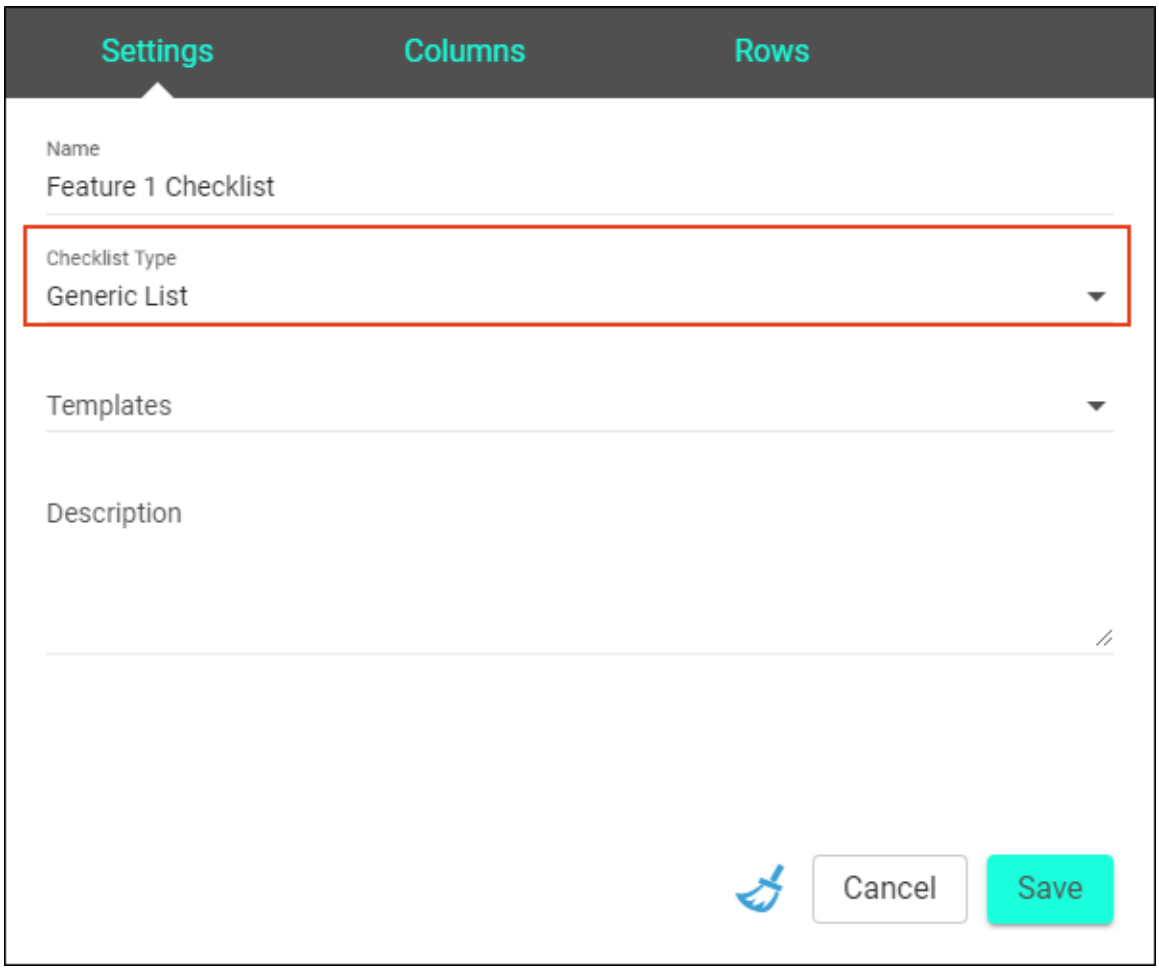
You can create checklists to keep track of work related to your uploaded Flare projects.

How to Create a Checklist

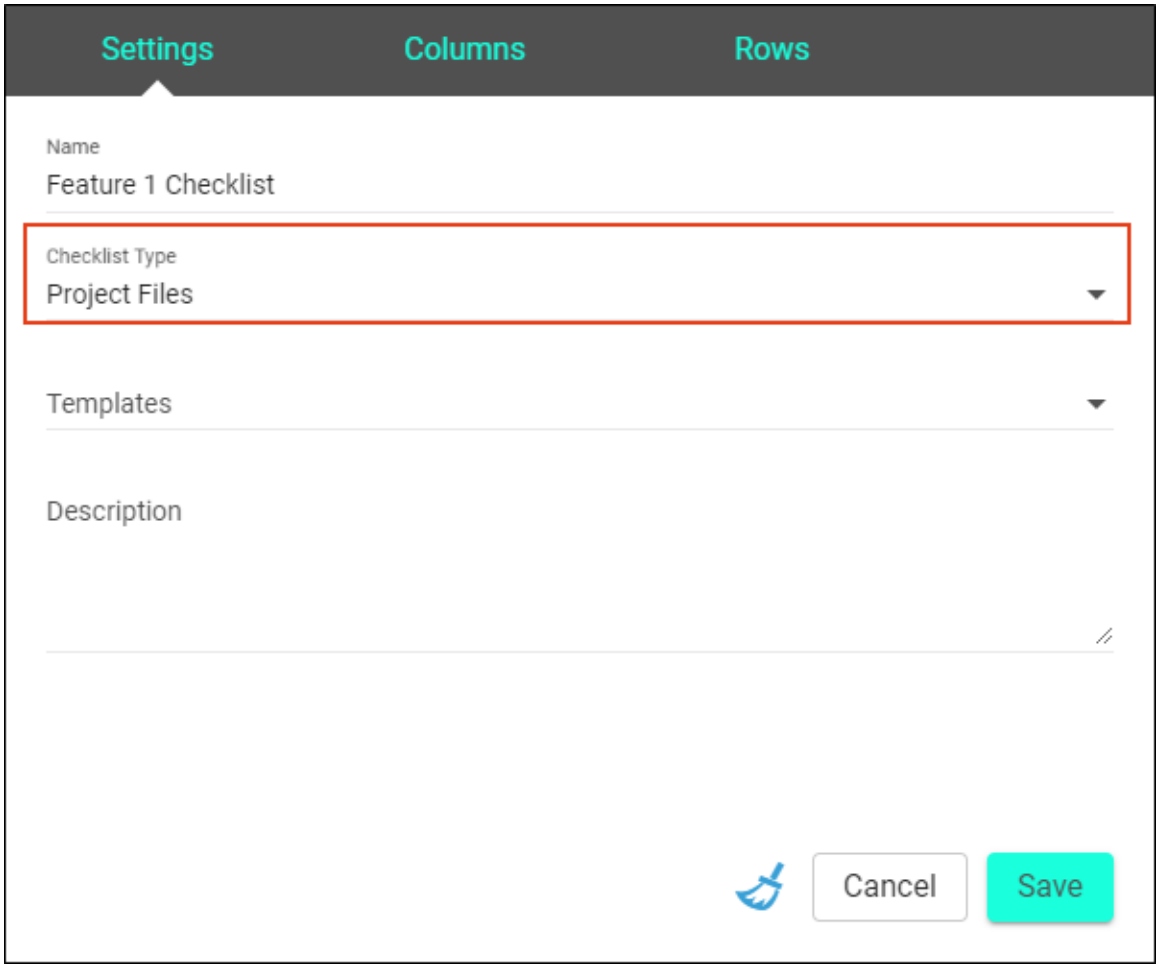
1. Select **Projects** on the left side of the interface, then click a project name to open it.
2. At the top of the interface, click **Checklists**.
3. In the toolbar click . The Create Checklist dialog opens.
4. On the **Settings** tab, complete the **Name** field.



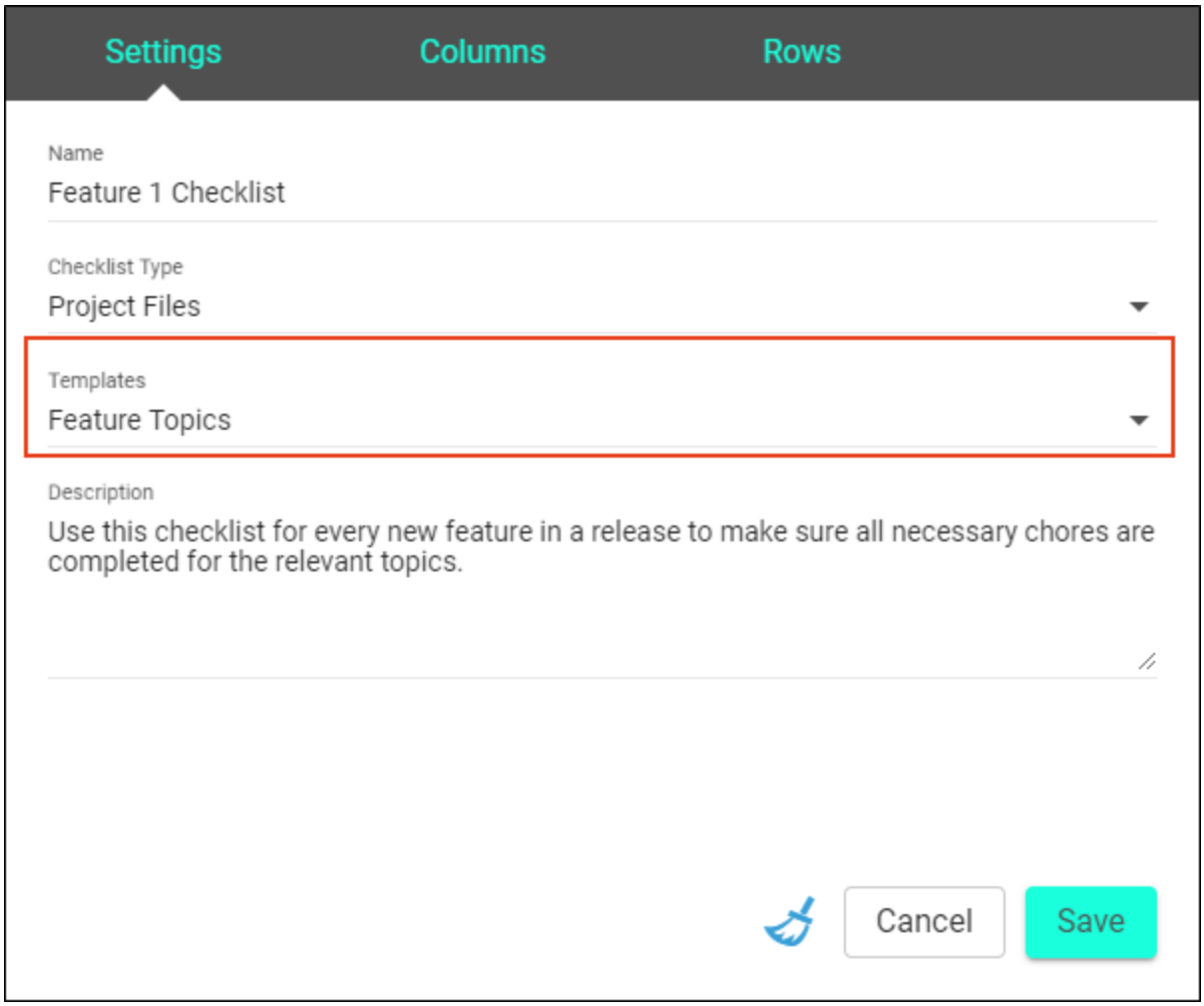
5. From the **Checklist Type** field, you can select a generic or project files checklist type. However, if you plan to use a template (see next step), you can skip this step because the template will populate this field (and others) automatically.
    - **Generic List** This lets you add a checklist where each row is manually created. The checklist can pertain to anything you want.



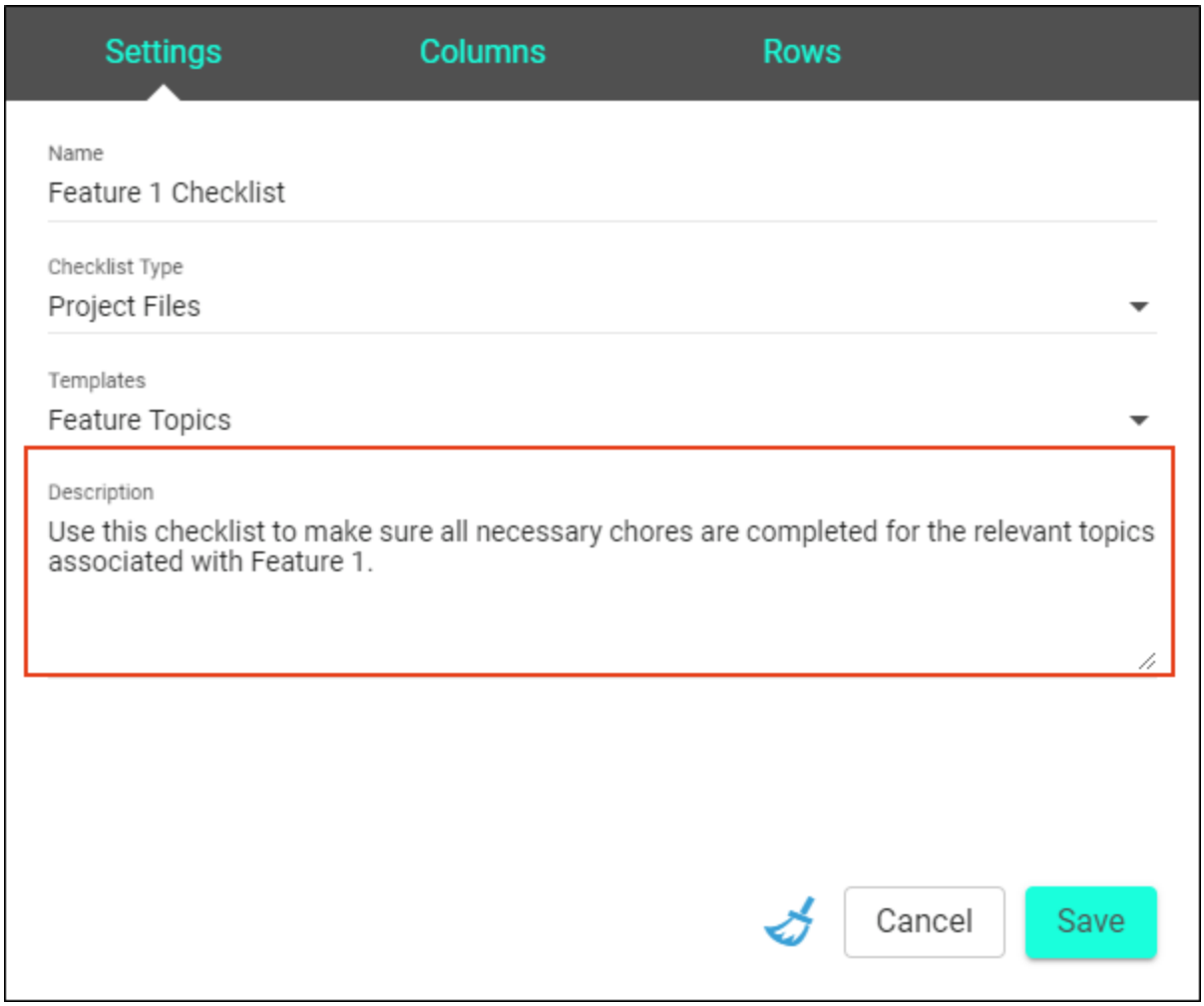
- Project Files This lets you add a checklist where the rows are based on files from your Flare project. Most commonly, these kinds of checklists will be used to track the progress of topics as you edit them, but if you like, you can create checklists that include any other kinds of files from the project (e.g., template pages, images, TOCs, targets, skins).



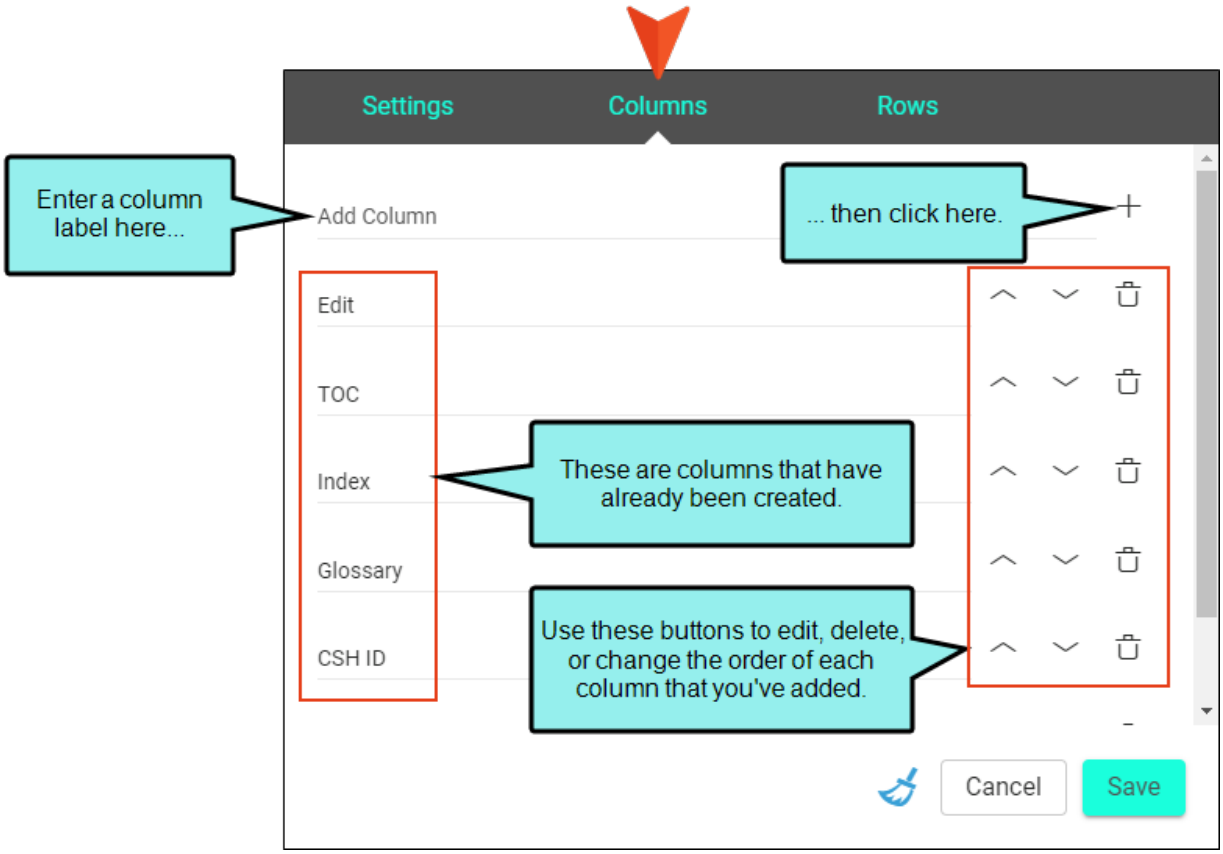
6. (Optional) From the **Templates** field, you can choose a template that you've previously created (see "Creating Checklist Templates" on page 164). Information from the template (i.e., type, description, columns, rows, notes) will populate the corresponding fields in this dialog. After the fields are populated, you can edit them if necessary.



7. (Optional) You can complete the **Description** field if you want a more detailed explanation of the checklist for yourself and other writers on your team.



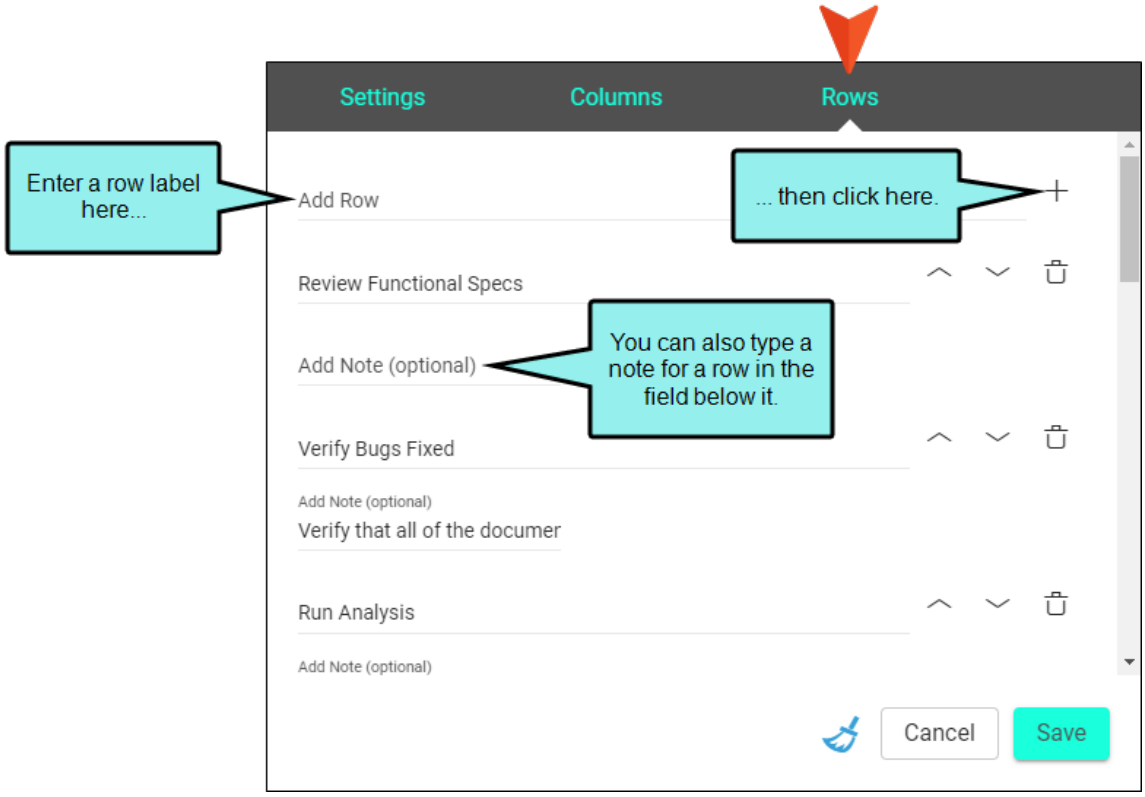
8. Select the **Columns** tab. Then create columns by typing a label in the text box and pressing **ENTER** or clicking ⊞. Repeat this for each column you want to add. You can use the up and down arrows to change the order of the columns (the column at the top of the dialog will appear at the far left in the checklist, and the column at the bottom will appear at the far right).



9. Select the **Rows** tab to create rows for the checklist. This is done differently, depending on the type of checklist you are creating.

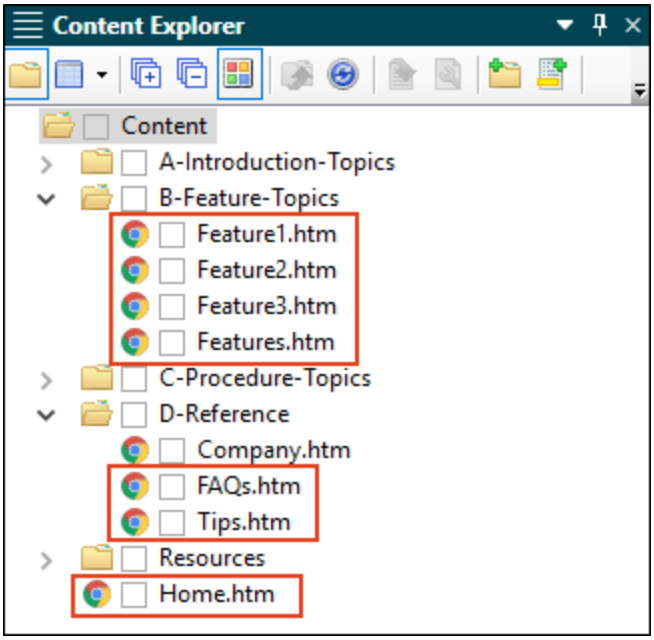
   - **Generic List** In the text box, type the label for a row and click ⊞. Repeat this for each row you want to add. You can use the up and down arrows to change the order of the rows.

     You can also complete the **Add Note** field for any row.

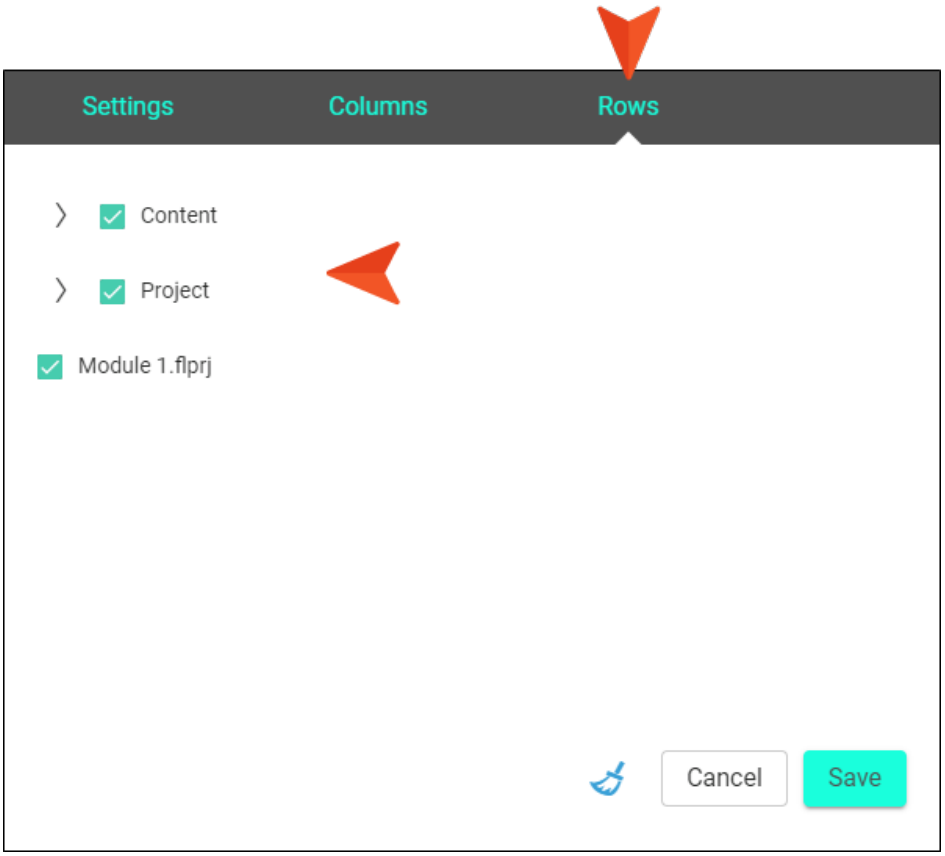


   - **Project Files** Your project folders and files are shown on the Rows tab, with check boxes next to each item. Any item with a check mark will be included in the checklist. It is likely that you will only want to include certain files in the checklist. Therefore, remove the check marks for any files you want to exclude. If you want to include a small number of items in a folder, the easiest thing to do is to begin by removing the check mark from the folder. Then navigate to the items you want to include and select them.

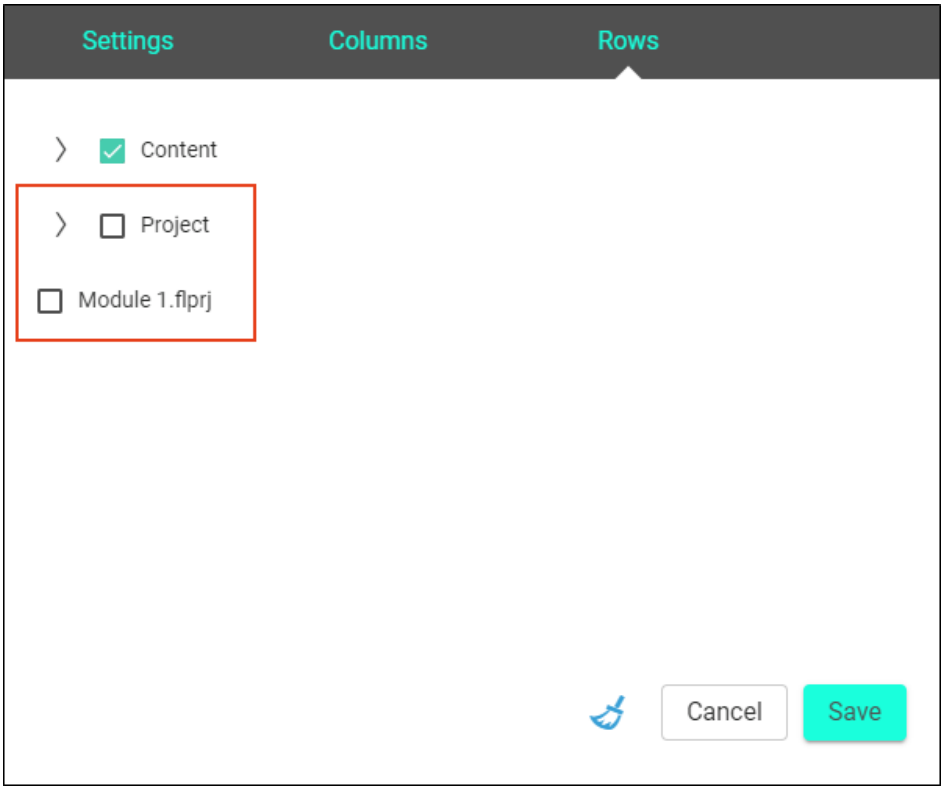
EXAMPLE You have a Flare project where the Content Explorer looks as follows, and you want to create a checklist that includes only the indicated topics.



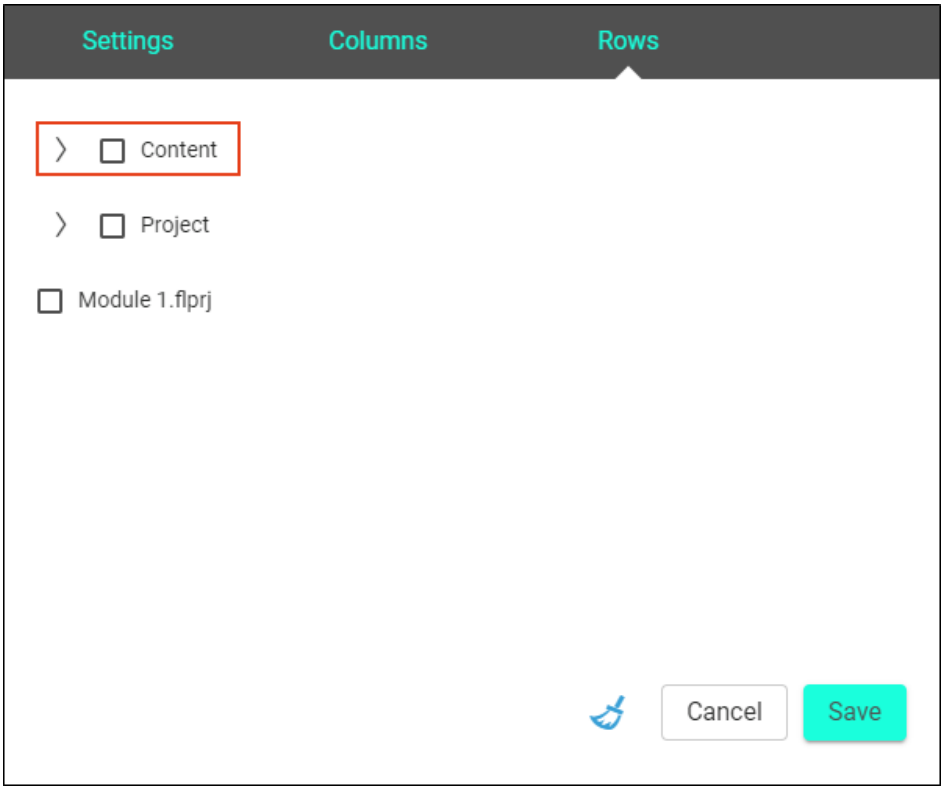
When you create the checklist, you will see the following on the Rows tab:



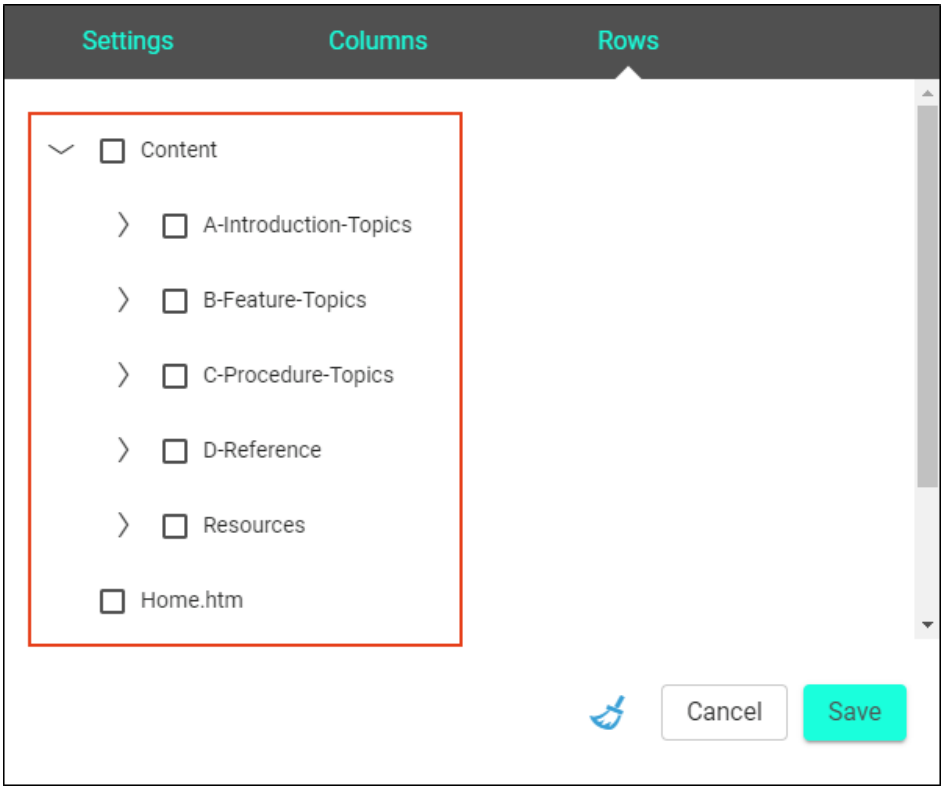
You don't want any of the files from the Project Organizer to be part of the checklist, and you don't want the main project file included either. So you can remove both of those check marks.



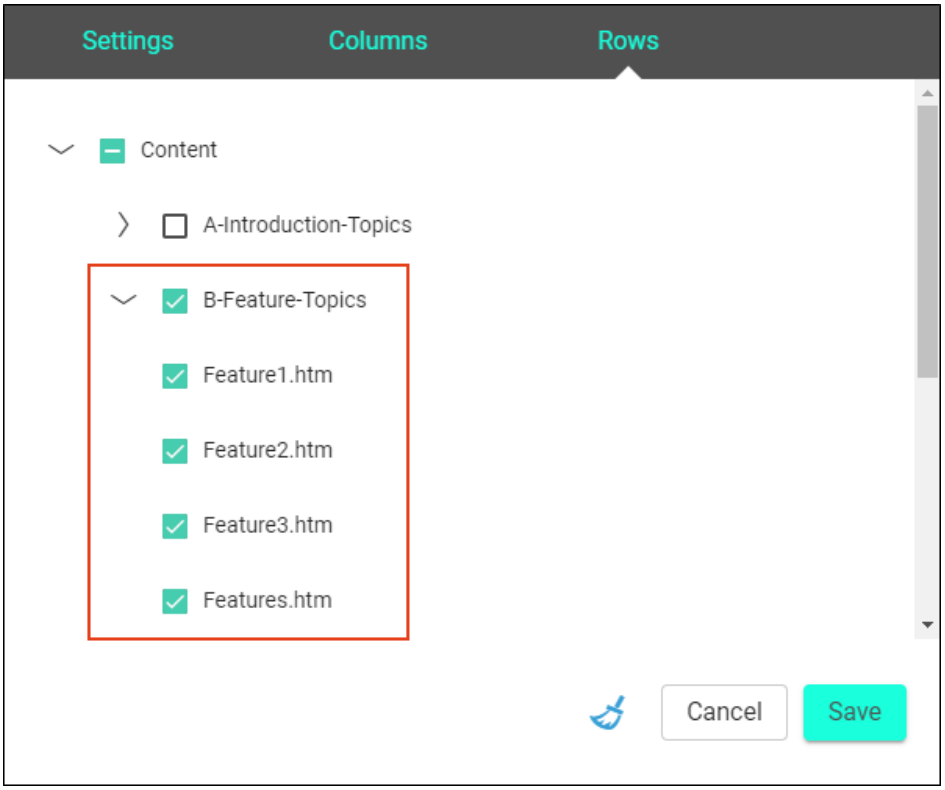
The files to be included in the checklist are topics, which are found in the Content folder. However, most of the topics and other files from the Content folder will not be part of the checklist. So the easiest thing is to initially remove the check mark from that folder as well.



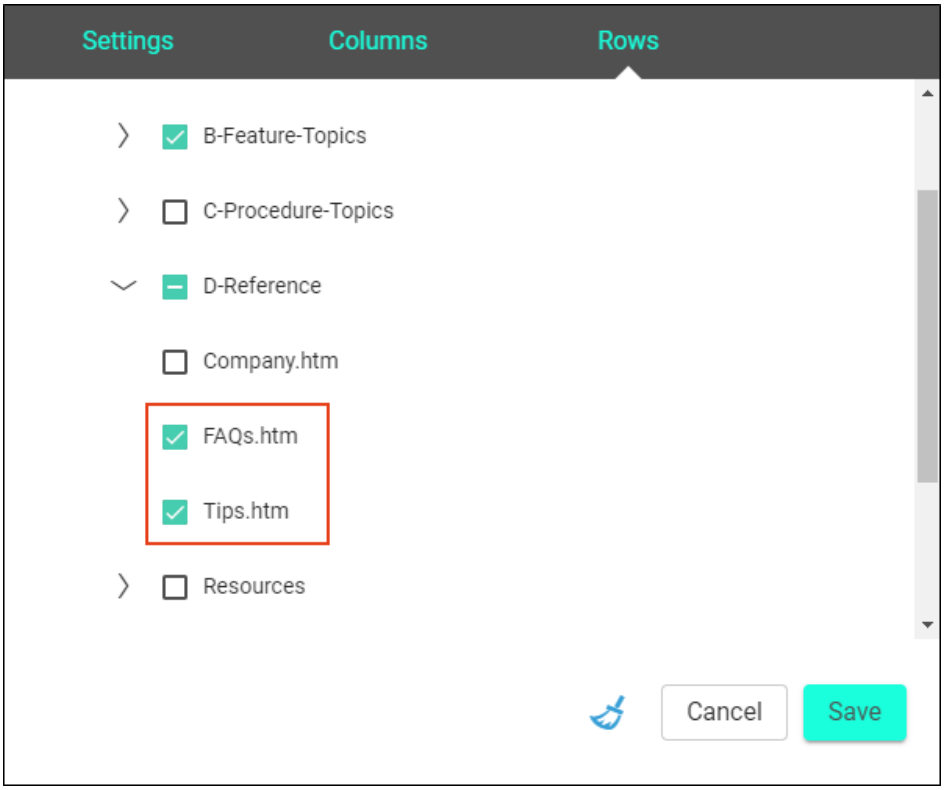
To choose specific topics in the Content folder, click the arrow to the left to expand it. You will see all of the subfolders and files within it.



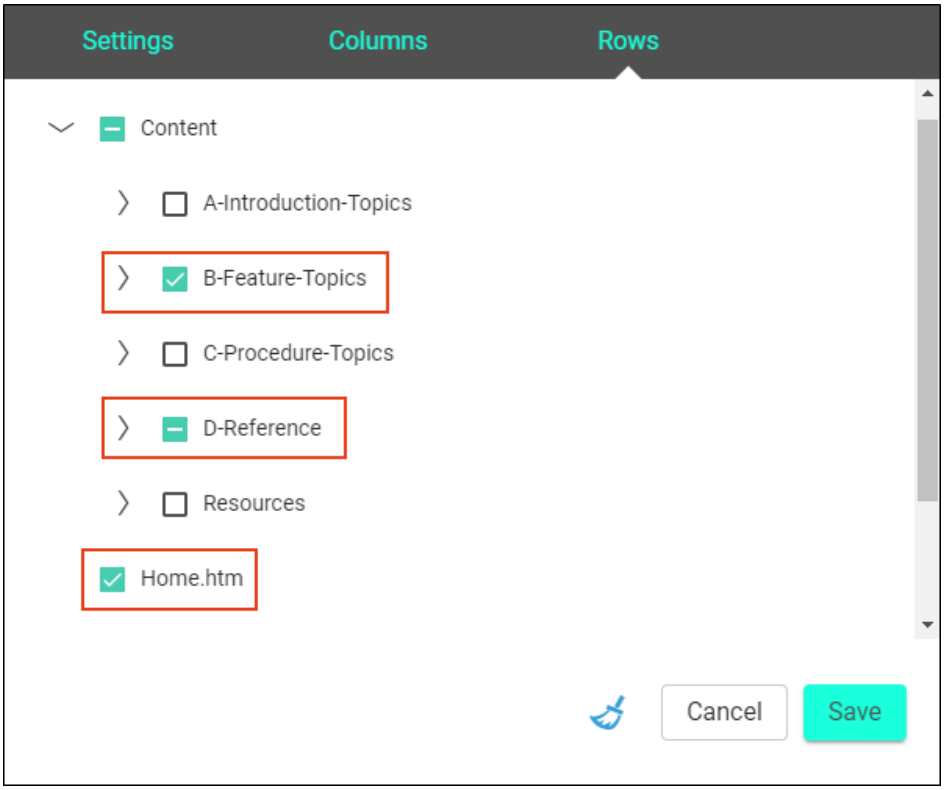
Since you want all of the topics in “B-Feature-Topics” to be included, you can just click the check box next to that subfolder. If you expand that subfolder, you will see that all of the topics within it are automatically selected.



You only want two of the topics in the “D-Reference” subfolder to be included, so expand that folder and select those topics.

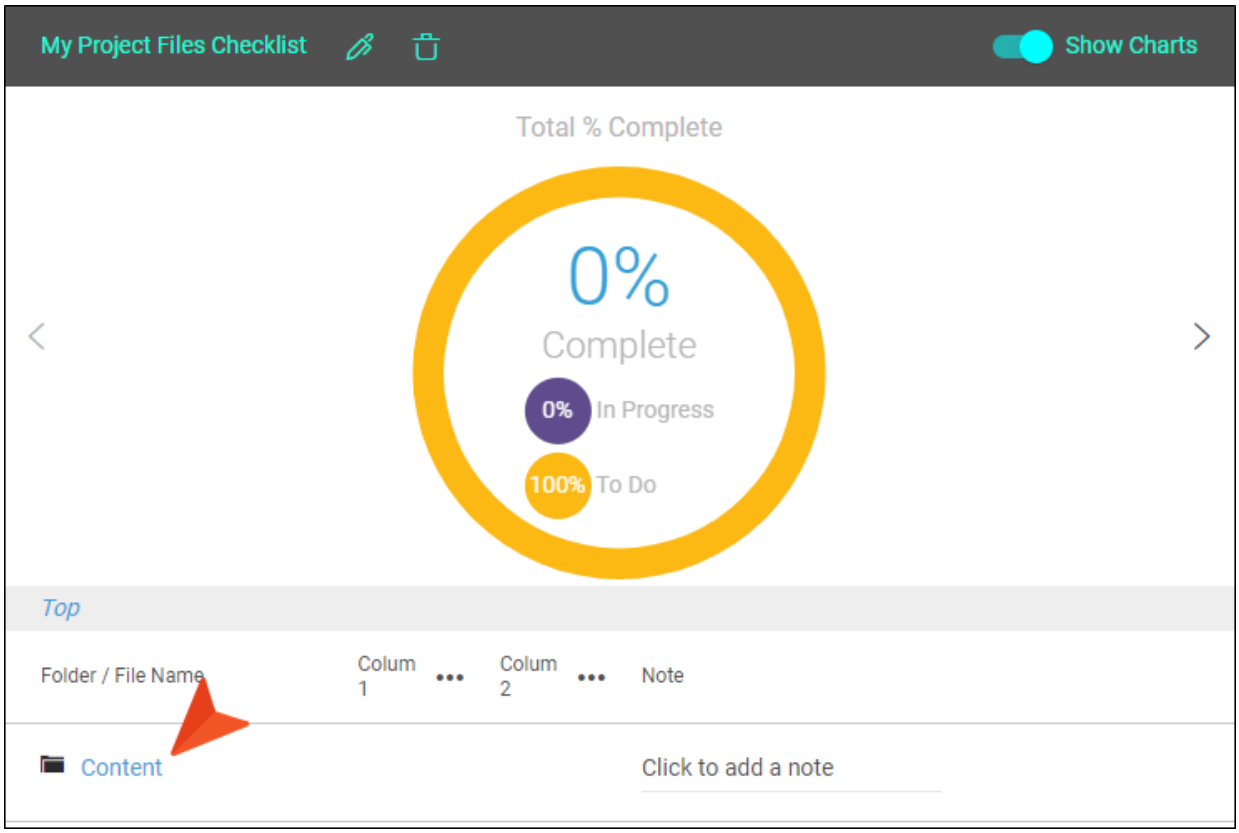


And finally, you want the Home topic at the root level of the Content folder to be included, so select it. The dialog should now look as follows when all subfolders are collapsed. Notice that the “D-Reference” subfolder has a small dash instead of a check mark; that’s because only some of the topics within it are selected.

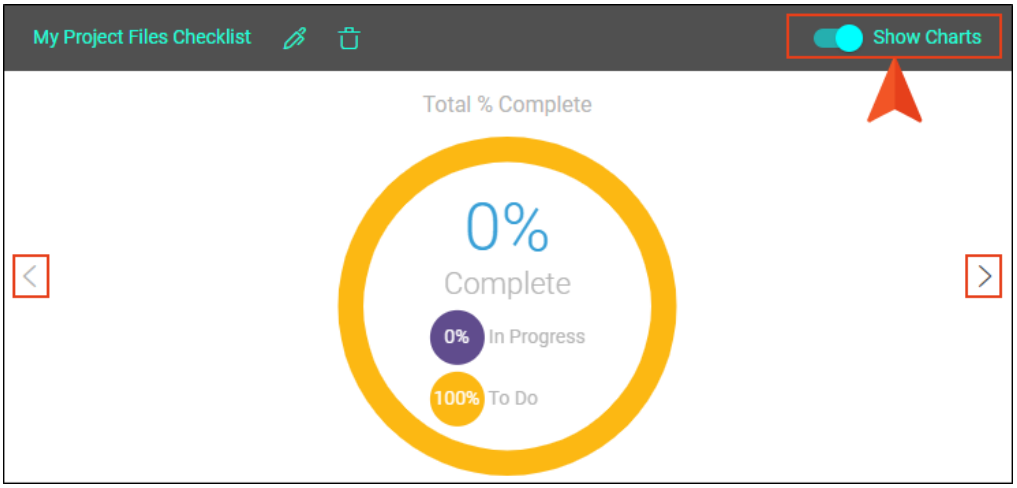


10. Click **Save**. The checklist is created and its name appears on the left side of the interface. The checklist graph and items appear to the right.

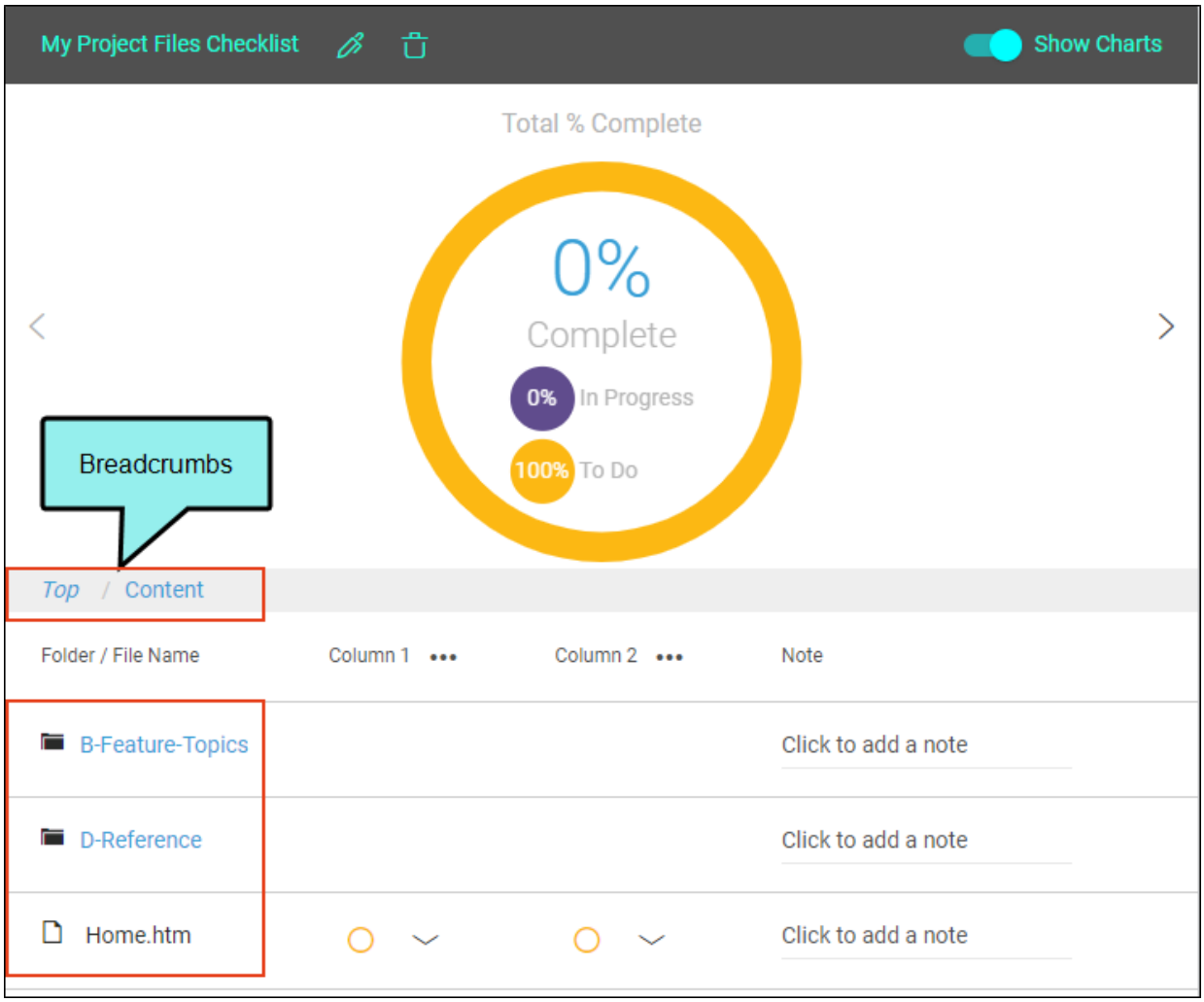
11. If you have created a checklist based on project files, you will see the first folder level (e.g., Content).



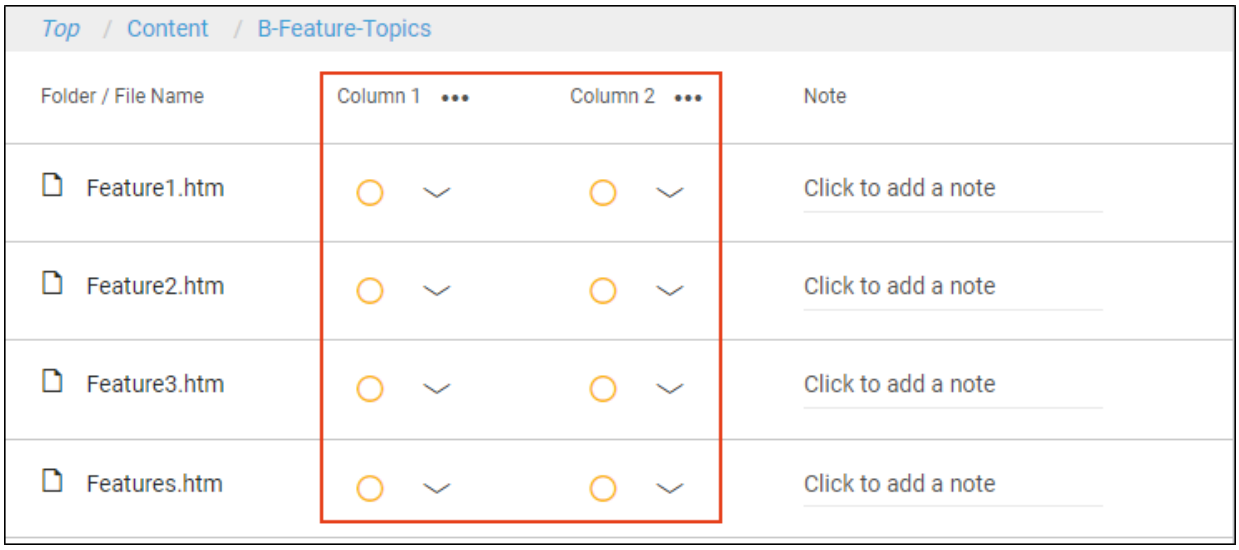
NOTE Since the Show Charts option is selected by default, the first folder level will display below the checklist graph. With Show Charts activated, you can click the arrow on the side of the current chart to see a different chart (e.g., Total % Complete, Checklist Item % Complete, and Checklist Completion Table). You can hide the visual display by clicking the Show Charts option.



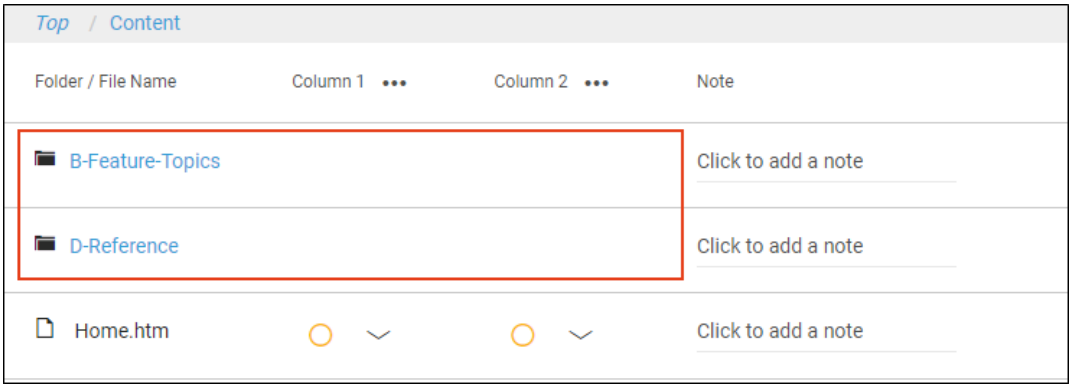
To navigate to a particular subfolder or set of files, click that folder. You will then see the content within that folder, as well as a breadcrumbs trail.



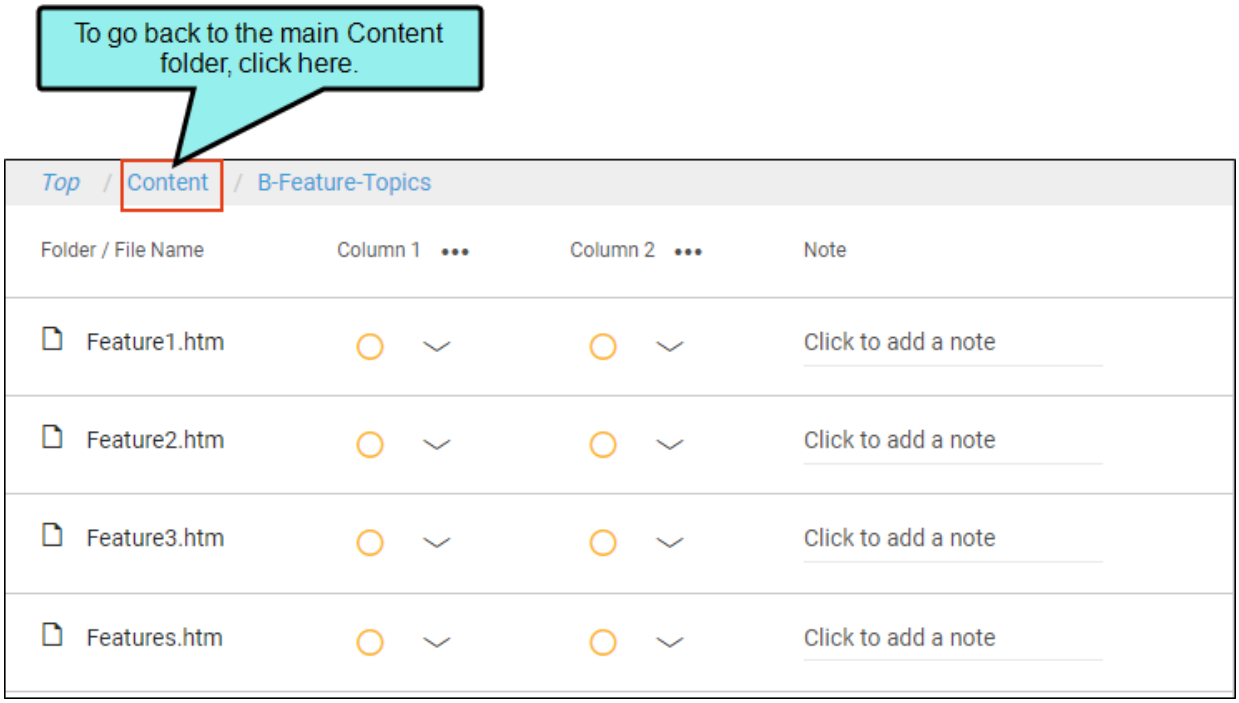
When you open a folder that contains files, you will see a circle for each column. These circles are where you set the status of each item.



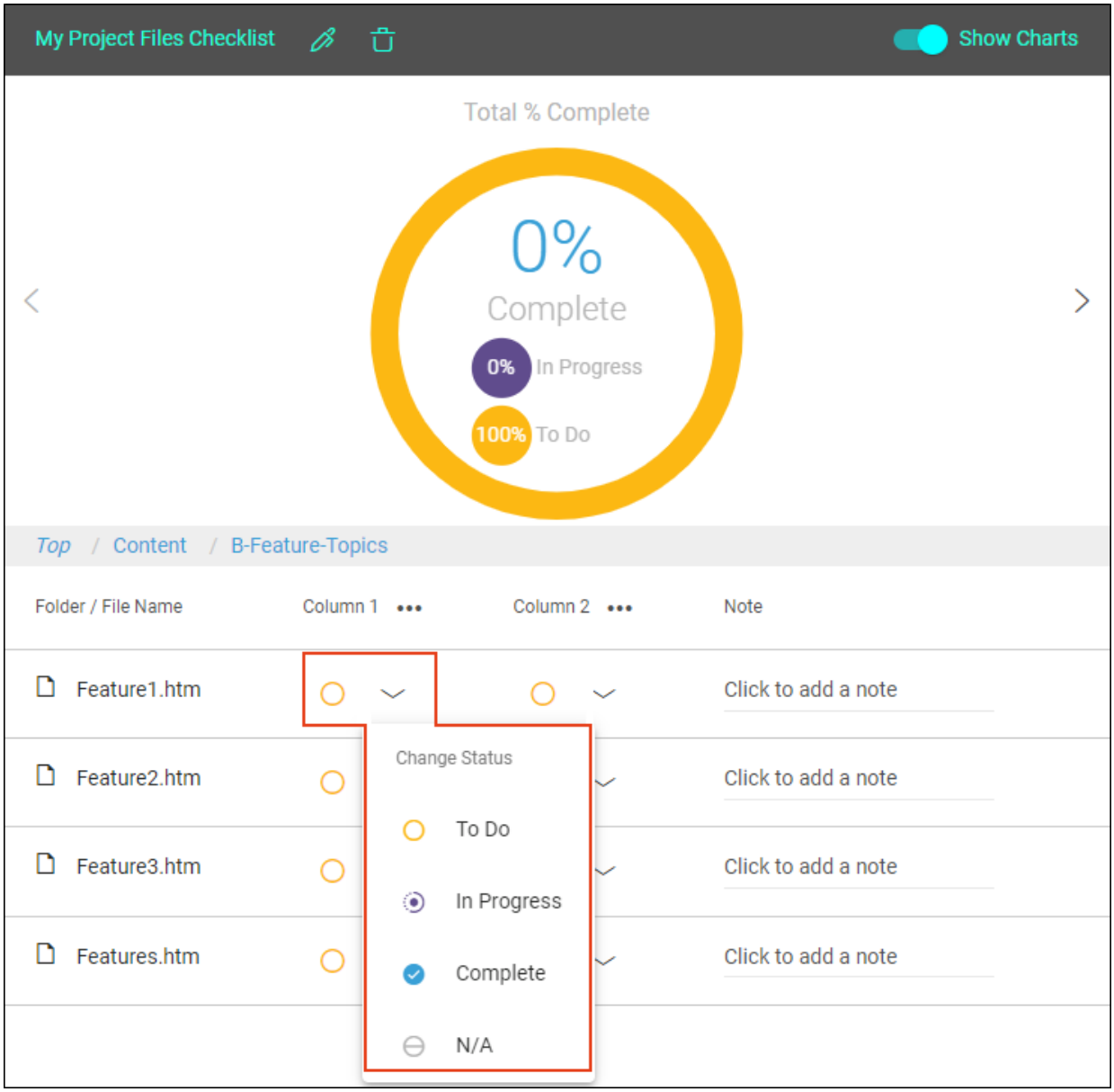
NOTE You will not see the status circles for rows represented by folders. You can set the status for files only.



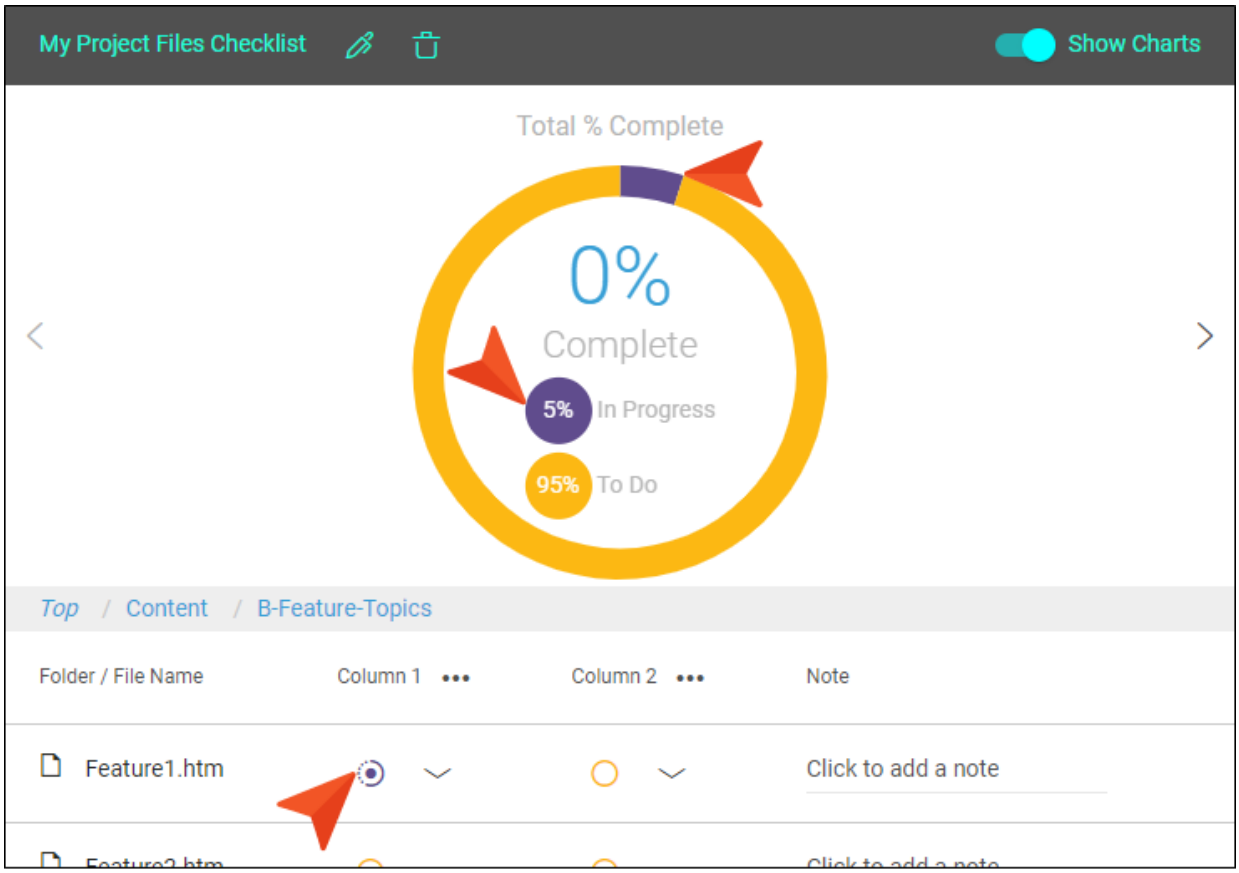
If you want to navigate back to a particular folder or subfolder, click it in the breadcrumbs.



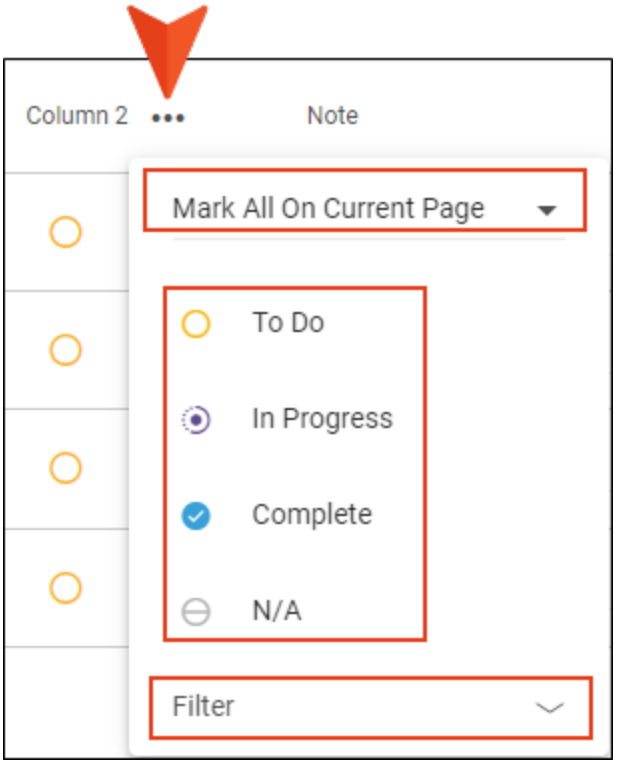
12. Click a circle and it will cycle through all the statuses each time you click. Alternatively, you can click the down arrow next to the circle and choose one of the statuses from the context menu.
    - To Do
    - In Progress
    - Complete
    - N/A Select this status if a particular column is not applicable for that row.


My Project Files Checklist
Show Charts
Total % Complete
0%
Complete
0% In Progress
100% To Do
Top / Content / B-Feature-Topics
Folder / File Name
Column 1
Column 2
Note
Feature1.htm
Feature2.htm
Feature3.htm
Features.htm
Click to add a note
Change Status
To Do
In Progress
Complete
N/A

Depending on your selection, the appearance of the circle changes, and the graph is updated.



TIP You can click the ellipsis in a column heading to set all of the items in that column to the same status. Click the drop-down at the top of the context menu to apply the settings to the current page of items, or to all pages in the checklist.

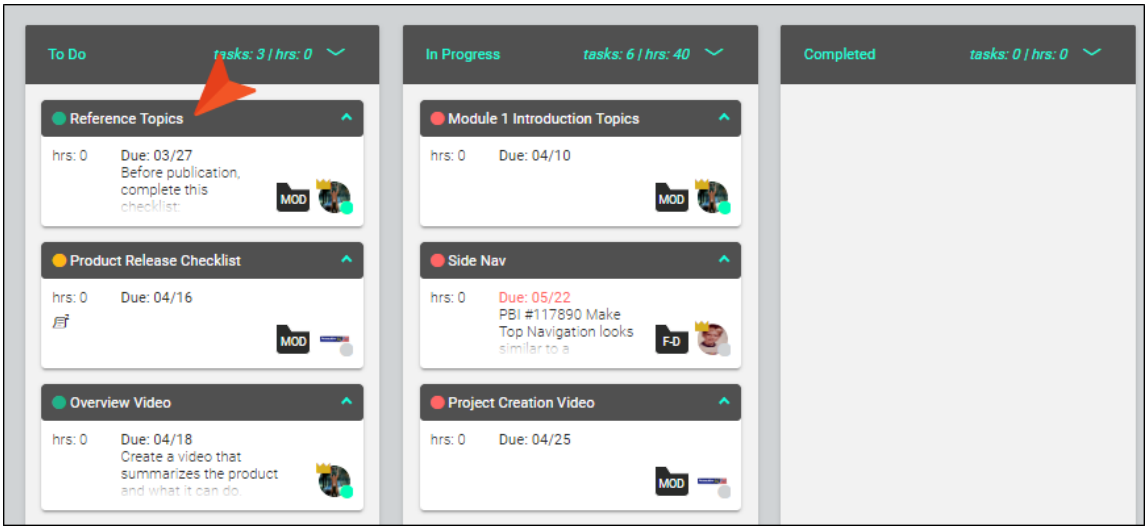


You can also click the Filter drop-down in the context menu to either select to filter only on a specific status, or clear a status type from the checklist.

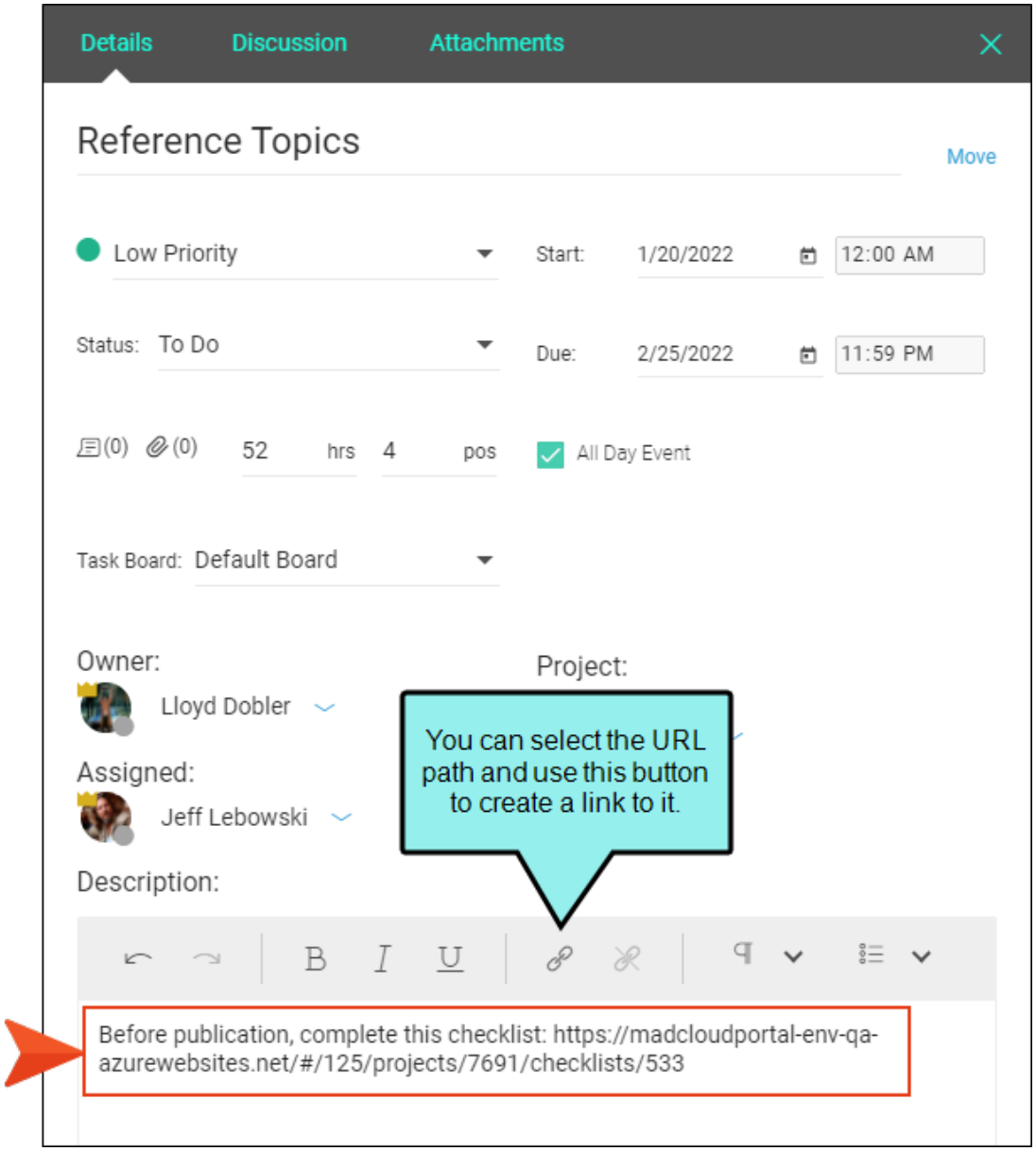
13. (Optional) If you want to add a comment for a particular row, you can click in the **Note** cell and type it.

TIP Although there is not currently any software integration between project checklists and tasks, you might find it helpful to create your own associations between the two as part of your overall workflow.

Let's say you have created a task to write a particular group of reference topics.



When creating or editing this task, instead of listing all of the things you need to do, you can paste a copy of the checklist URL in your task description and create a link to it.



Once you finish the checklist, you can return to the Tasks page to move the task card to the Completed milestone.

Editing Checklists

After creating a checklist, you can edit it to make changes or additions.

How to Edit a Checklist

1. Select **Projects** on the left side of the interface, then click a project name to open it.
2. At the top of the interface, click **Checklists**.
3. On the left side of the page, select the checklist you want to edit. The checklist grid and chart are displayed to the right.
4. In the toolbar click .
5. You can use the Settings tab to change the name, type, or description of the checklist. You can use the other tabs to make any row or column changes for the checklist. For more details about each option, see "Creating Checklists" on page 135.
6. (Optional) You can click **Save as Template**, which will make the checklist content available when you create other checklists in the future. See "Creating Checklist Templates" on page 164.
7. When you are finished, click **Save**.

Deleting Checklists

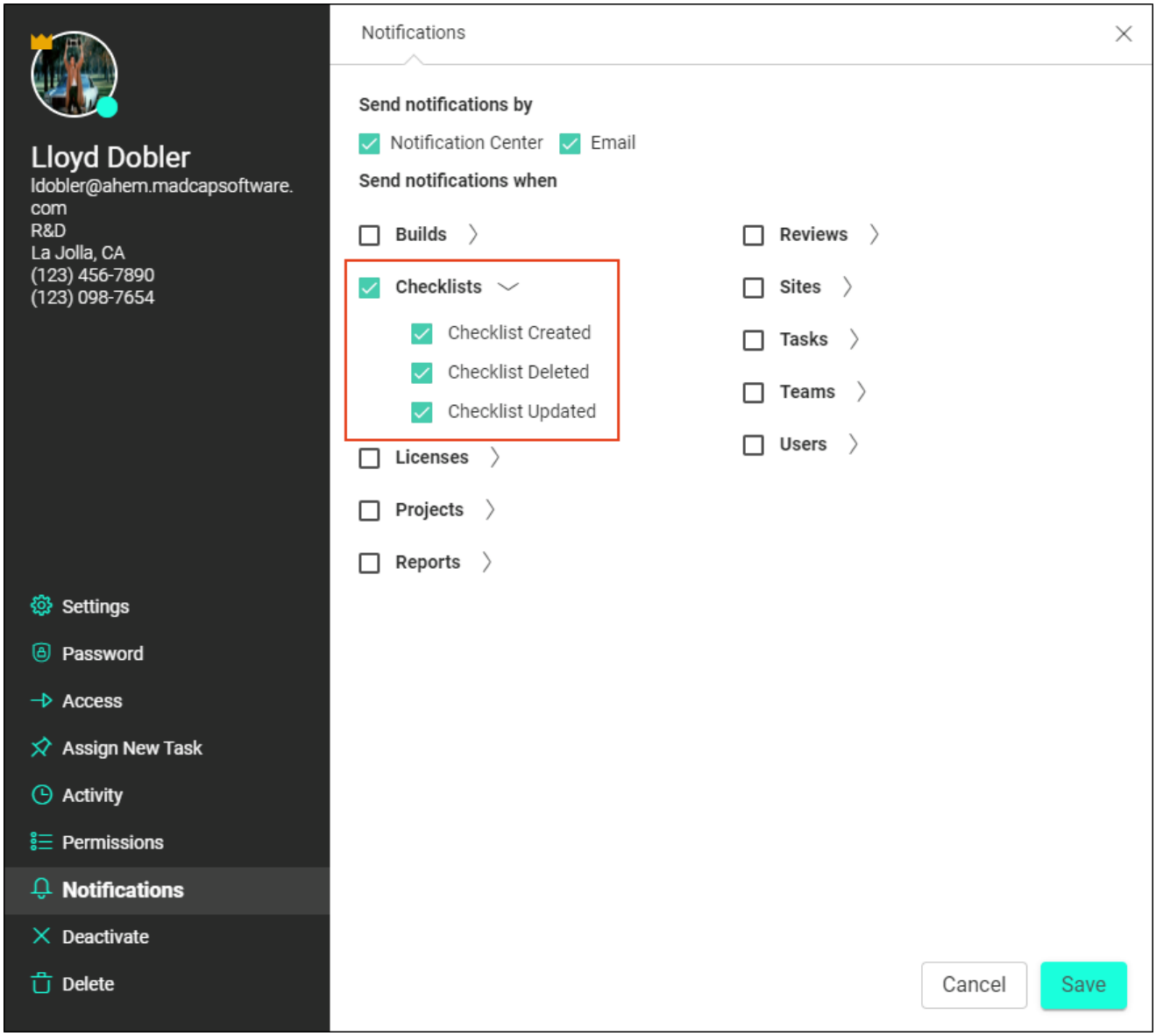
Checklists can be deleted from the project.

How to Delete a Checklist

1. Select **Projects** on the left side of the interface, then click a project name to open it.
2. At the top of the interface, click **Checklists**.
3. On the left side of the page, select the checklist you want to delete.
4. In the toolbar click .
5. In the dialog that opens, type the name of the checklist in the field and click **Delete**.

Setting Notifications for Checklists

In your account settings, you can choose when to be notified for specific activities. The user who is notified depends on the activity. For more information, see the online Help.



Checklist Templates

To be more productive, you can create and use templates when working with checklists.

Creating Checklist Templates

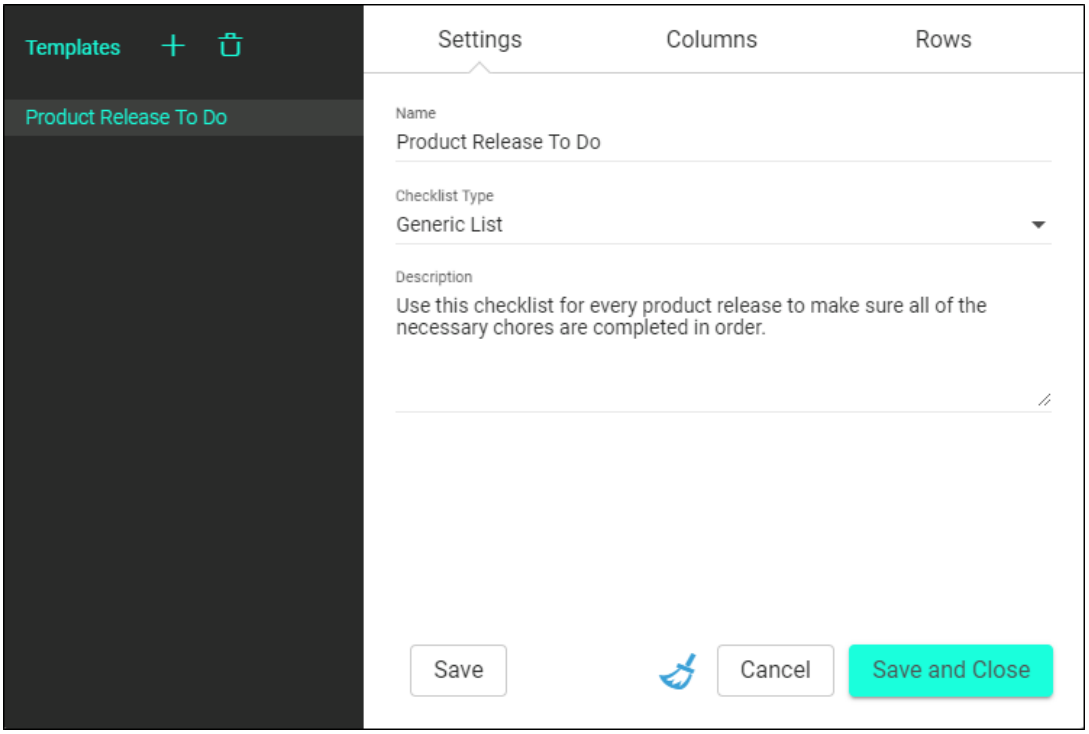
You can create checklist templates to save time when adding new checklists. When you create a new checklist, you can base it on any of your saved templates. This is quite useful when you are creating many checklists that share the same type, description, columns, rows, or notes. Rather than typing all the information manually, the template populates the relevant fields automatically.

Checklist templates are available across all projects that you upload to Central.

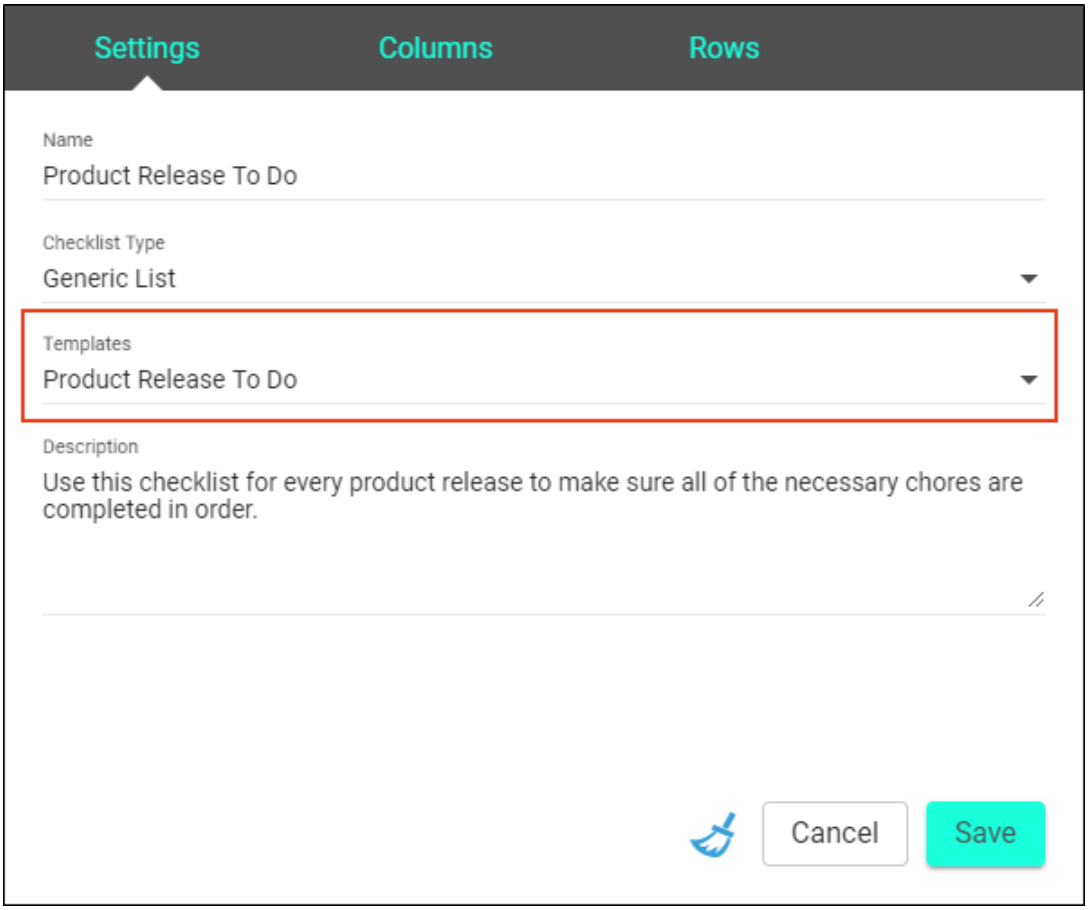
EXAMPLE You have a product release "To Do" checklist that you use every time you publish updated documentation for each of your company's products. The checklist has just one column to indicate whether a particular task has been completed. However, there are about a dozen rows listing things you need to accomplish as part of the product release. Also, some of the rows have detailed notes that provide more explanation of what needs to be done.

Items	Complete •••	Note
Research		Click to add a note
Review Functional Specs		Click to add a note
Verify Fix Bugs		Finish assigned doc bugs.
Run Annotation Report		Click to add a note
Run Analysis		Scan for critical issues.
Search Output &Test		Click to add a note
Check Troubleshooting Top...		Click to add a note
Replace Images		Click to add a note

If you were to completely re-create that checklist from scratch each time you had a product release, you would spend a lot of time typing (or copying and pasting) the same information repeatedly. Therefore, you create a template that includes the necessary checklist type, description, columns, rows, and notes.



Then, whenever you need to create a new product release checklist, you select your template.



How to Create a Checklist Template

1. Select **Projects** on the left side of the interface, then click a project name to open it.
2. At the top of the interface, click **Checklists**.
3. In the toolbar click .
4. Complete the **Name** field.
5. From the **Checklist Type** field, you can select a project files or generic checklist type.
   - **Project Files** This lets you add a checklist where the rows are based on files from your Flare project. Most commonly, these kinds of checklists will be used to track the progress of topics as you edit them, but if you like, you can create checklists that include any other kinds of files from the project (e.g., template pages, images, TOCs, targets, skins).
   - **Generic List** This lets you add a checklist where each row is manually created. The checklist can pertain to anything you want.
6. (Optional) You can complete the **Description** field if you want a more detailed explanation of the checklist for yourself and other writers on your team.
7. Select the **Columns** tab. Then create columns by typing a label in the text box and pressing **ENTER** or clicking . Repeat this for each column you want to add. You can use the up and down arrows to change the order of the columns (the column at the top of the dialog will appear at the far left in the checklist, and the column at the bottom will appear at the far right).
8. If you selected **Generic List**, complete the **Rows** tab in the same way that you created the columns. The only difference is that you can also complete an **Add Note** field for each row.

   > NOTE The Rows tab is not available if you selected Project Files. That's because the files will be different with each project where you might be using the project checklist.

9. Click either **Save** (to save the template and keep the dialog open) or **Save and Close** (to save the template and close the dialog). The template name appears on the left side of the interface. It will also now be available in the Create Checklist dialog.

How to Save an Existing Checklist as a Template

1. Select **Projects** on the left side of the interface, then click a project name to open it.
2. At the top of the interface, click **Checklists**.
3. On the left side of the page, select the checklist you want to save as a template. The checklist grid and chart are displayed to the right.
4. In the toolbar click . The Edit Checklist dialog opens.
5. If necessary, make any changes to the checklist that you want. For more details about each option, see "Creating Checklists" on page 135.
6. Click **Save as Template**.

Other Template Options

When working in the Template Manager, there are a few more things you can do.

	Lets you create a new template
	Deletes the selected template
	Clears all fields in the dialog

Editing Checklist Templates

After creating a checklist template, you can edit it to make changes or additions. Keep in mind that existing checklists that were previously created based on the template will not be changed as a result of the template edits.

How to Edit a Checklist Template

1. Select **Projects** on the left side of the interface, then click a project name to open it.
2. At the top of the interface, click **Checklists**.
3. In the toolbar click .
4. On the left side of the dialog, select the checklist template you want to edit. The fields are populated to the right.
5. Use the tabs and fields to make any changes that you want. For more details about each option, see "Creating Checklists" on page 135.
6. Click either **Save** (to save the template and keep the dialog open) or **Save and Close** (to save the template and close the dialog).

Deleting Checklist Templates

Checklists templates can be deleted, in which case they will no longer be available for selection when creating new checklists. Keep in mind that existing checklists that were previously created based on the template will not be affected as a result of the template deletion.

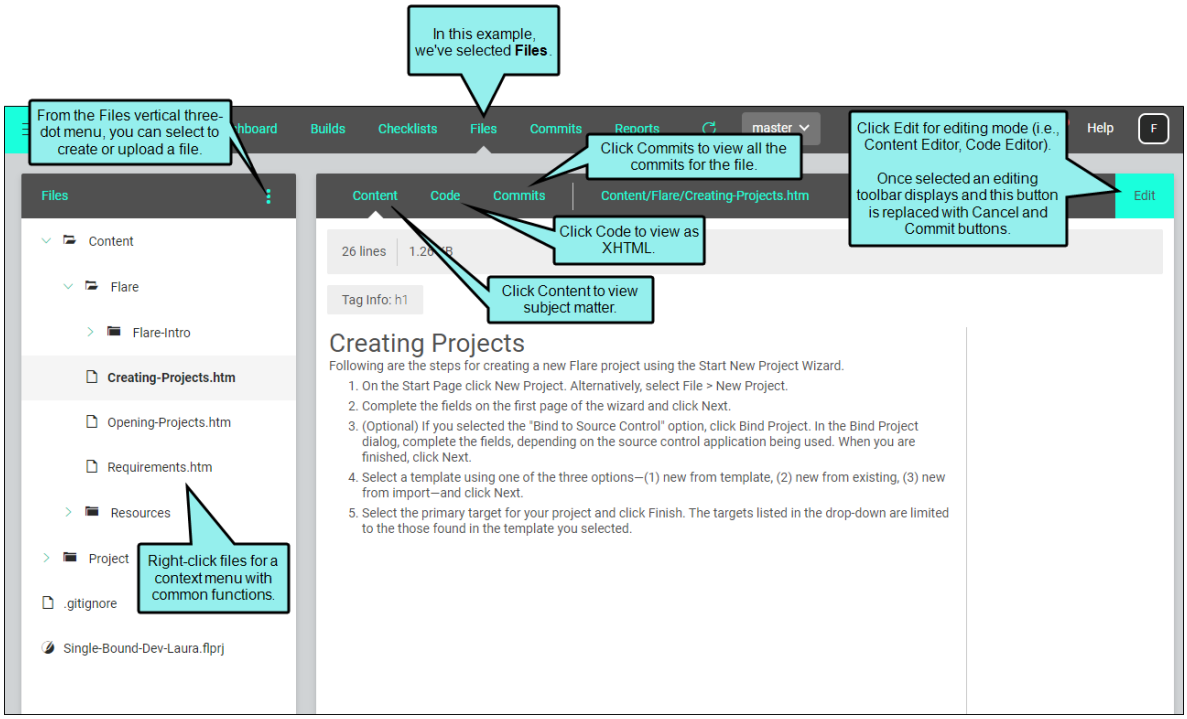
How to Delete a Checklist Template

1. Select **Projects** on the left side of the interface, then click a project name to open it.
2. At the top of the interface, click **Checklists**.
3. In the toolbar click .
4. On the left side of the dialog, select the checklist template you want to delete.
5. Click .
6. In the dialog that opens, enter the name of the template in the field and click **Delete**.

CHAPTER 8

Files

Using the Files page you can view, edit, and author files for any project that is uploaded to Central.



This chapter discusses the following:

Authoring

MadCap Central's cloud environment empowers you to author Madcap Flare project files directly without having to use Flare's desktop application. You can add new files, edit existing content, upload files, and even use integrated ChatGPT). The Central interface includes an easy-to-use editor for content viewing and editing, and if permission is granted, you can also edit in a code editor.

For more information and details, see the online Help or the *Central Authoring Guide*.

EXAMPLE Your documentation team has created an online Help system using MadCap Flare. Your larger organization uses MadCap Central as its platform to manage the content. A director, who is not trained in using Flare, views some content from the project in Central and sees an ideal place to insert a relevant new topic. Without having to go through a review process or track down a writer to do the work, the director uses Central to quickly create and add a topic to the project. The change is committed to the project. Anyone who works in the project in Central will see the change, and Flare users will see updates once the remote and local repositories are synchronized.

Permission Required?

Authoring is available to users with the Author status. By default, users with Author status have the following permissions set:

- ☑ Create/Edit Files

  If this is deselected, then viewing files in a read-only mode is allowed. On the left side of the page, the Files vertical three-dot menu is not available.

- ☑ Edit Code

  If this is deselected, the XHTML in the Code view is read-only.

  Editing code is regarded as a capability for an advanced user. If not done properly, the code can become malformed quickly. Administrators can prevent users from editing the code by deselecting the Edit Code permission.

In addition, AI Assist involves the following permissions:

- ☑ Server Management

  This is required to integrate a ChatGPT account with a Central license in the license settings.

- ☑ Edit Files With AI Assist

  This is required to use AI Assist (and therefore ChatGPT) when modifying topics and snippets.

  > NOTE Even if this permission is enabled, ChatGPT does not scan anything on your computer. The only information ChatGPT can acquire from you is what you enter manually into the prompt when using AI Assist. If your company has strict policies against AI or ChatGPT, simply do not use it.

For more information about permissions, see the Central online Help.

NOTE For the authoring feature to work properly, your project must be single-bound to Central as the primary source control provider. The authoring feature does not support dual-bound projects.

NOTE Since Central is a remote repository, those who use Flare after changes are made in Central, need to synchronize their remote and local repositories.

- **Central side** Content is authored and committed to the project in Central.
- **Flare side** To interact with updated content in the Flare application, use source control to pull changes from the remote repository and sync it to the local repository.

During the synchronization process, keep the following in mind.

- If two authors are editing the same file, at the same time, but one is working in Flare and the other is working in Central, then there may be conflicts when trying to synchronize the repositories. These conflicts can be resolved using the source control features in Flare.
- If two authors are both working in Central and are simultaneously editing the same file, then the first person to commit the file has precedence. When the second person commits their file, they will be alerted to the issue and prompted to refresh to get the latest files.

Renaming Project Files

Instead of opening a project in Flare to rename files, you can do that directly in Central.

Permission Required?

Authoring is available to users with the Author status. By default, users with Author status have the following permissions set:

- ☑ Create/Edit Files

  If this is deselected, then viewing files in a read-only mode is allowed. On the left side of the page, the Files vertical three-dot menu is not available.

- ☑ Edit Code

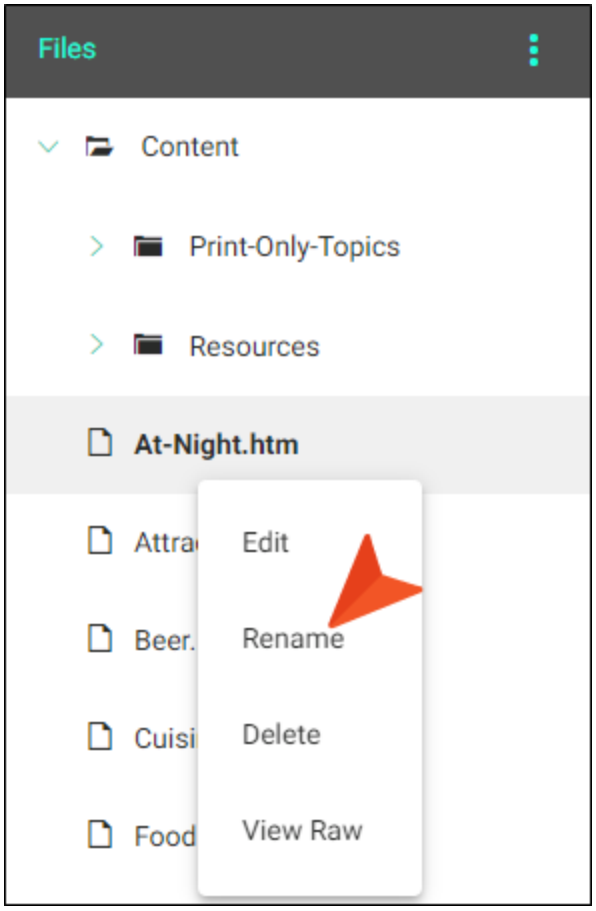
  If this is deselected, the XHTML in the Code view is read-only.

  Editing code is regarded as a capability for an advanced user. If not done properly, the code can become malformed quickly. Administrators can prevent users from editing the code by deselecting the Edit Code permission.

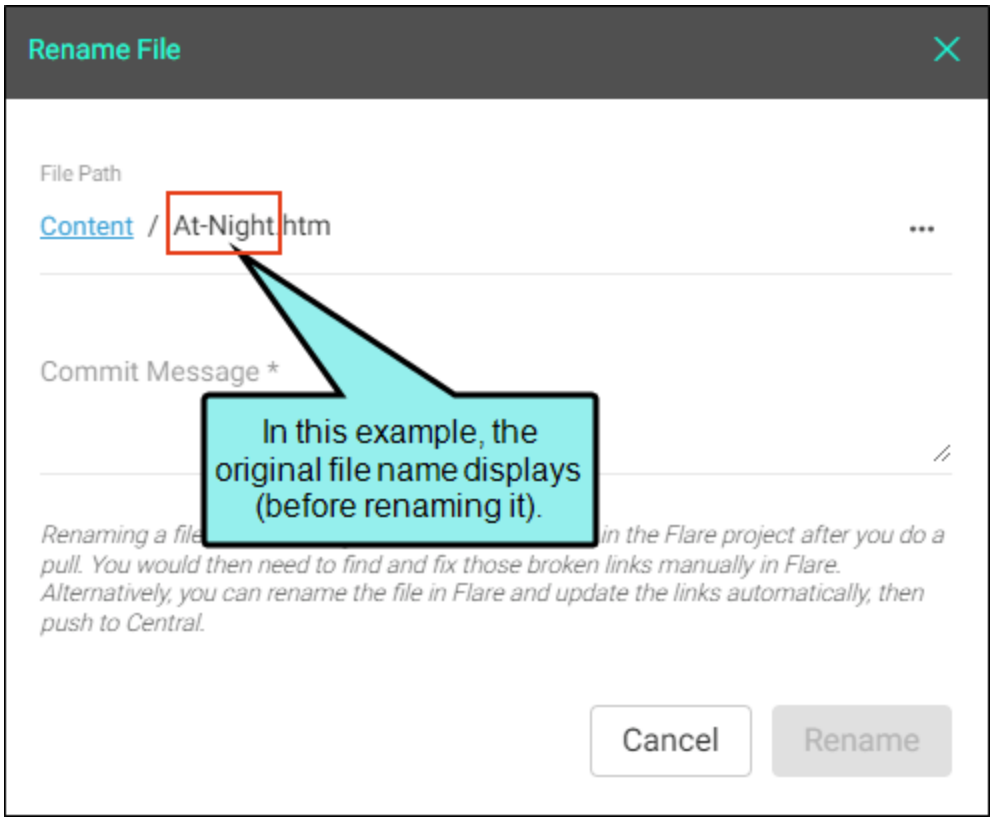
For more information about permissions, see the Central online Help.

How to Rename Files in Central

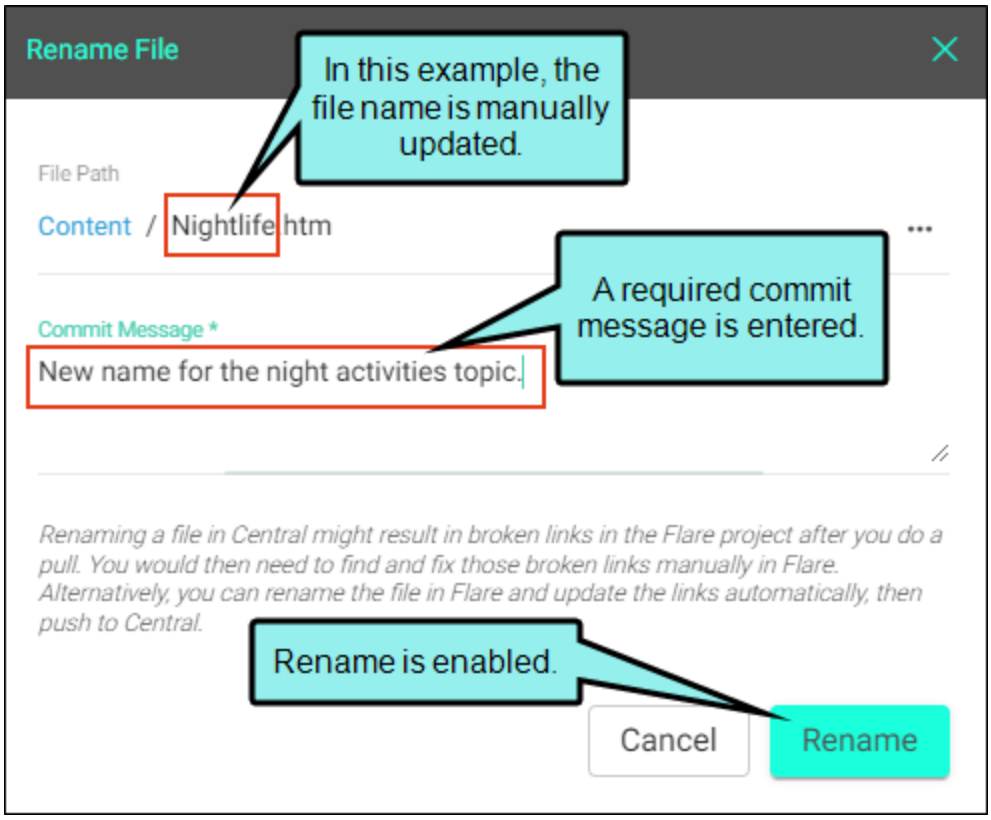
1. On the left side of the Central interface, click **Projects**.
2. Click **Files** at the top of the screen.
3. From the left side of the Files page, you can right-click a file, and select **Rename**.



4. In the Rename File dialog, at the end of the **File Path** field, enter a new name for the file.



5. In the **Commit Message** field, enter a (required) comment for the commit.



6. Select **Rename**.

> **WARNING** Renaming a file in Central might result in broken links in the Flare project after you do a pull. You would then need to find and fix those broken links manually in Flare. Alternatively, you can rename the file in Flare and update the links automatically, then push to Central.

Viewing Project Files

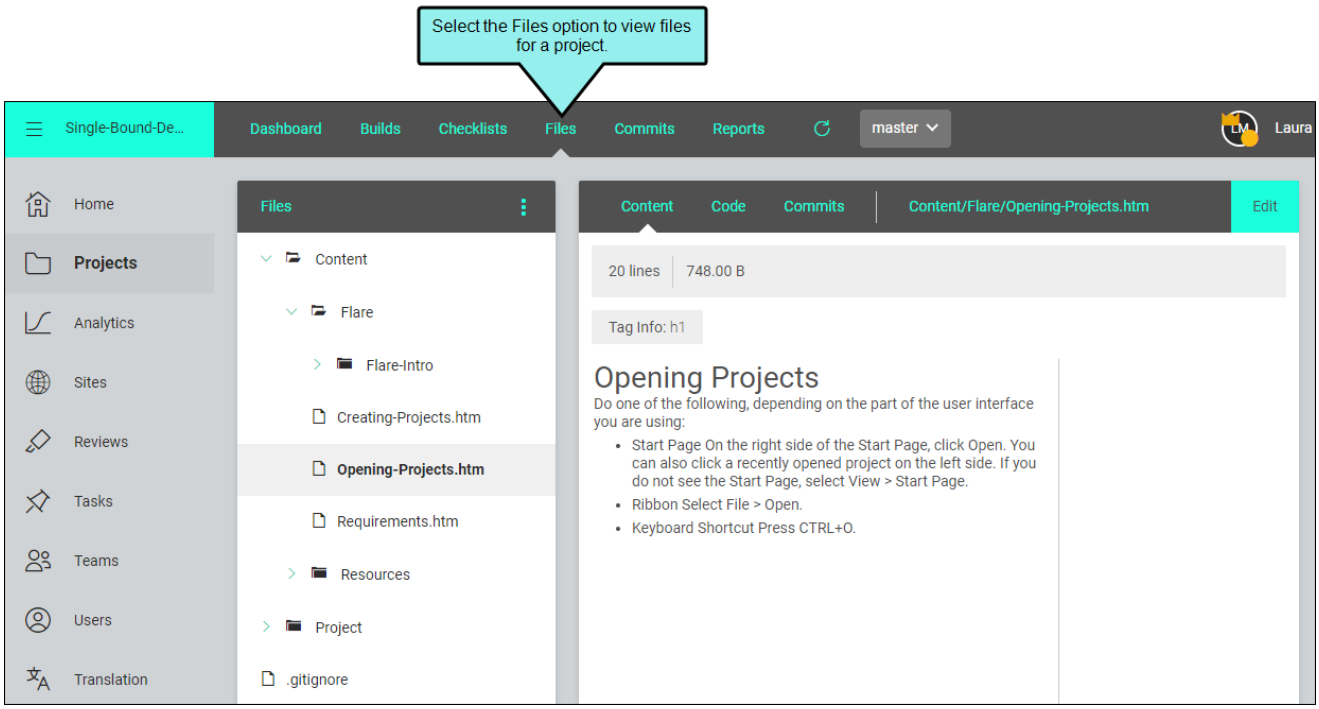
After opening a project in Central, you can view files associated with a project.

Permission Required?

No special permission is required for this activity. All authors who are associated with the project are allowed.

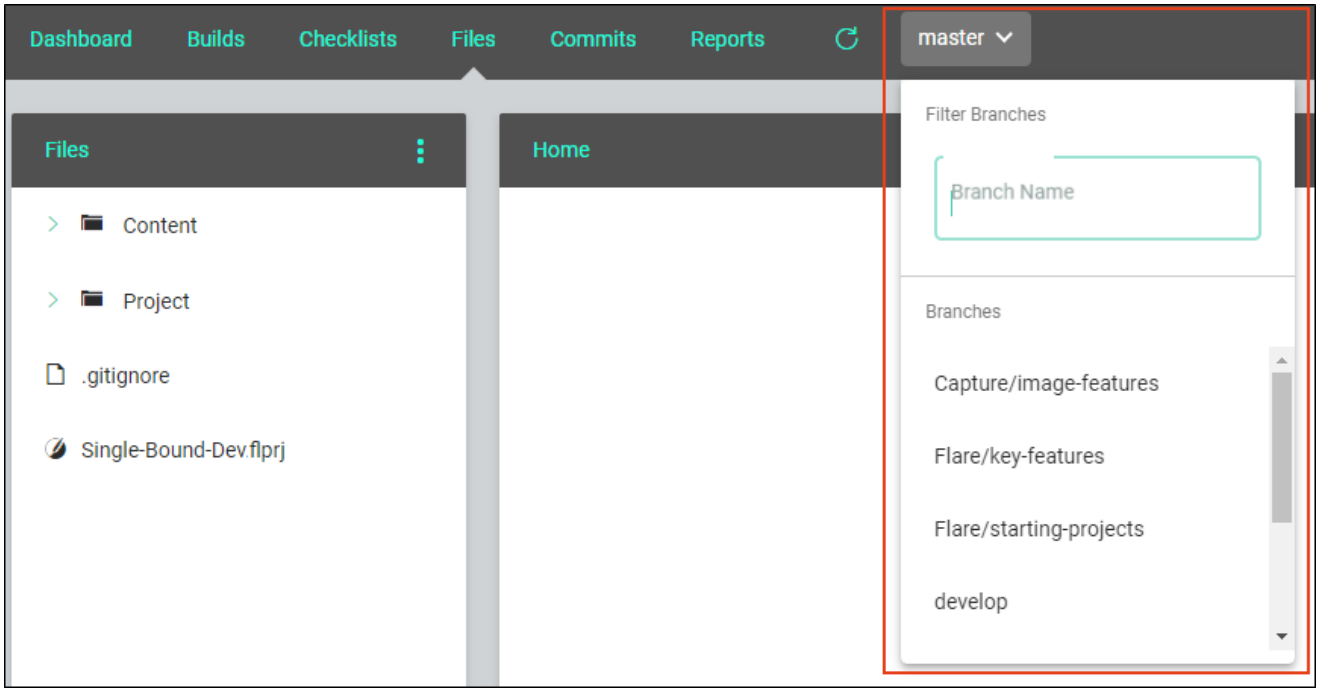
Viewing Files Page

After you open a project in Central, you can click the **Files** option at the top. This lets you see all files in your project in a few different ways.



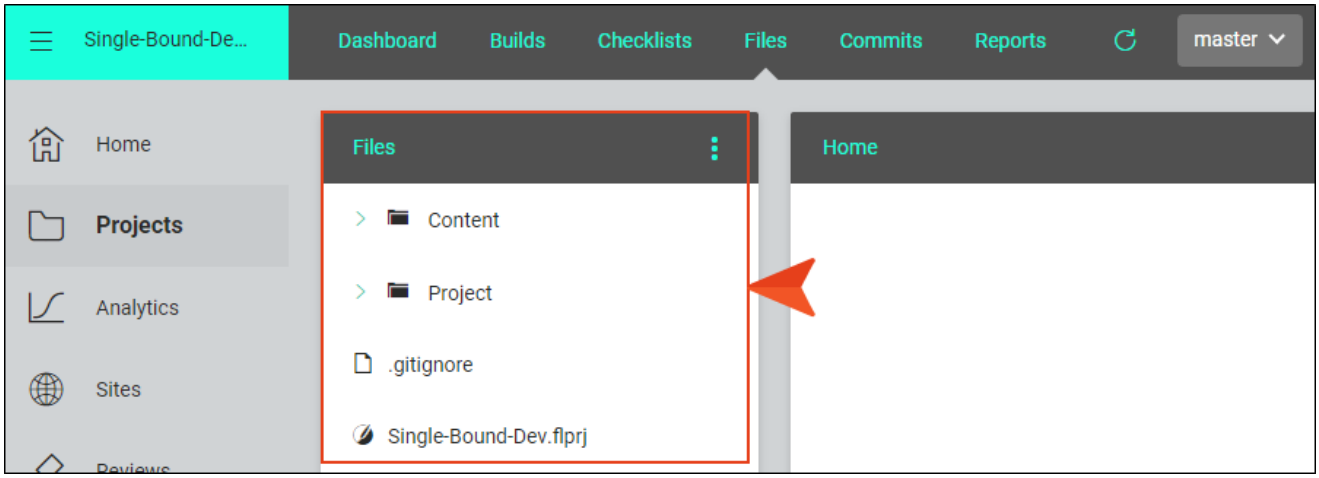
Switching Branches

If you are using branching, you can click the drop-down at the top of the page and switch from one branch to another. The files specific to that branch display accordingly on the page. A filter field at the top of the drop-down lets you limit the branches shown.

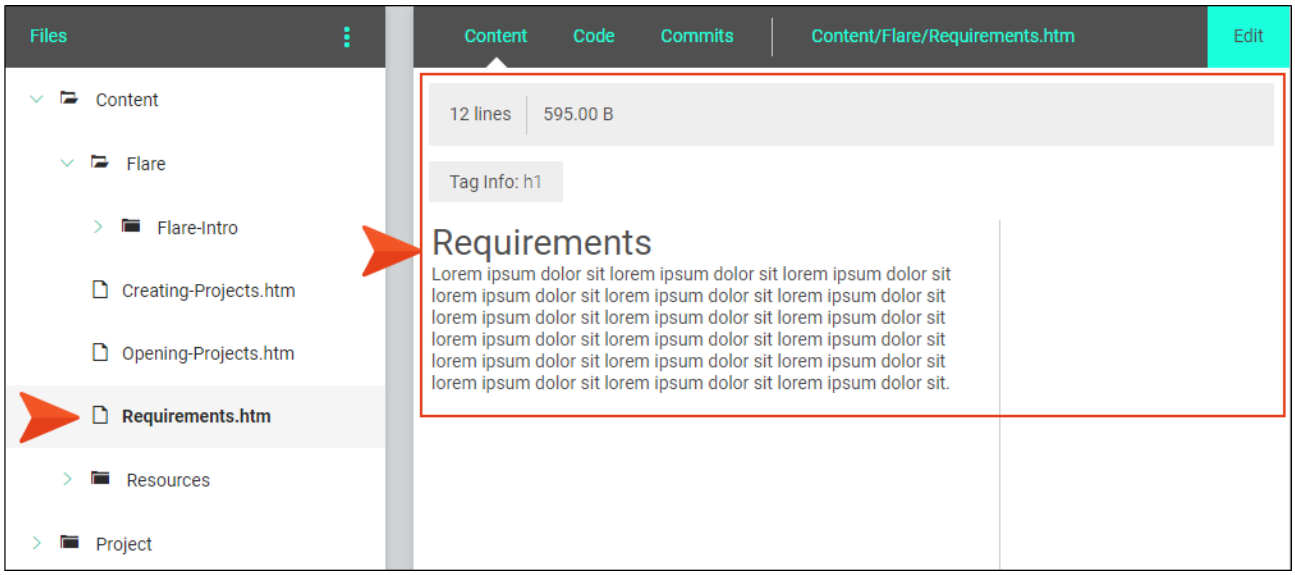


Viewing Individual Files

The project files are shown on the left side of the page. You can expand or collapse the folders to locate a specific file.



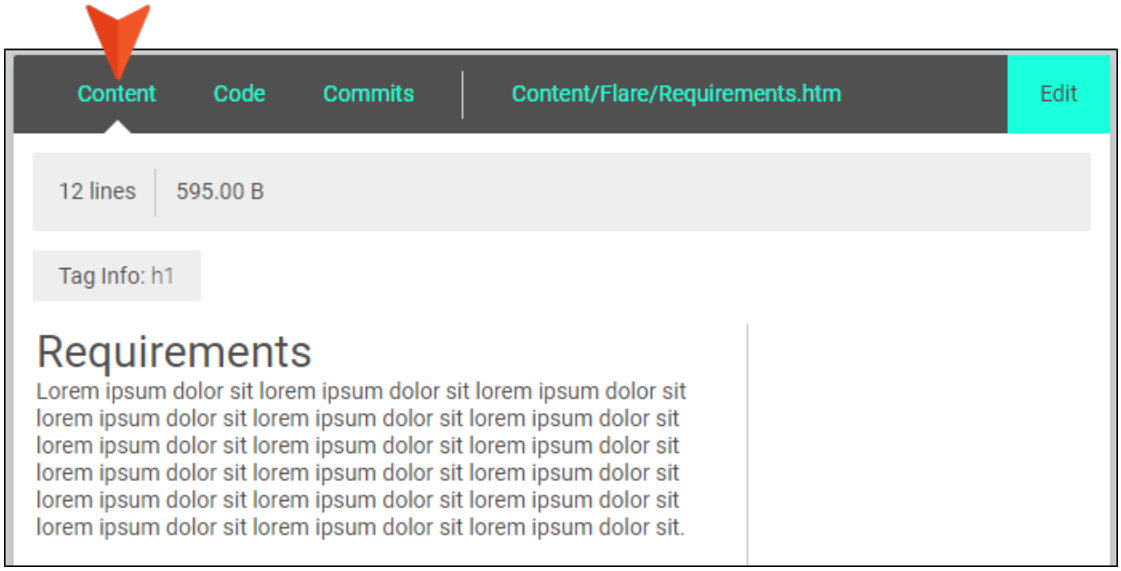
If you select a file, it opens as a preview in the editor to the right.



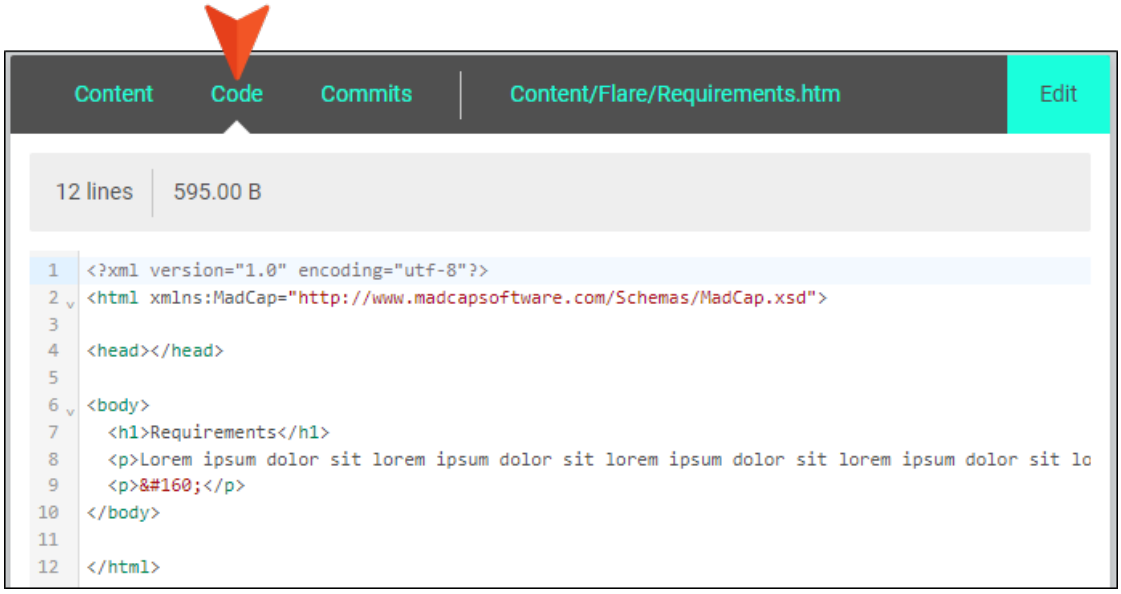
The file will not look exactly the way it does in the Flare project. Content files (e.g., topics and snippets) display in much the same way they do when viewing them in the editor on the Reviews page. Other files might only display the code.

Depending on the type of file you open, you can switch between Content, Code, and Commits views.

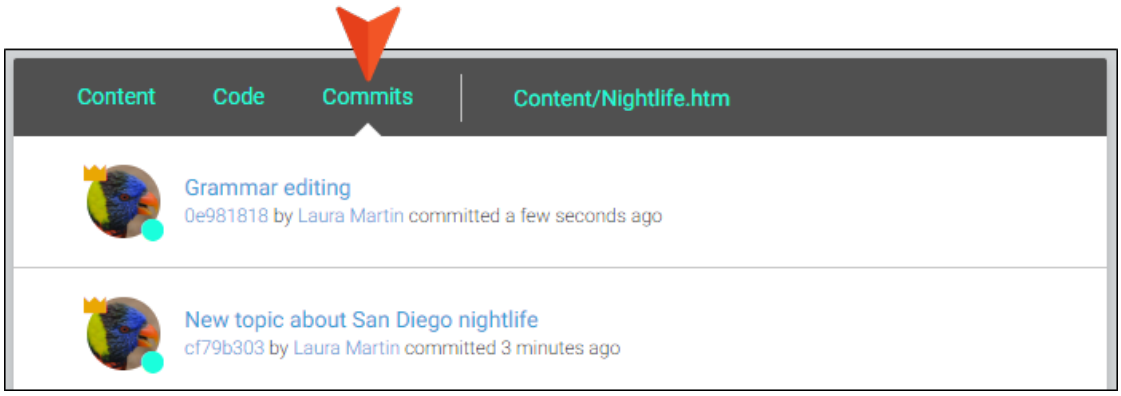
Here is the Content view:



Here is the Code view:

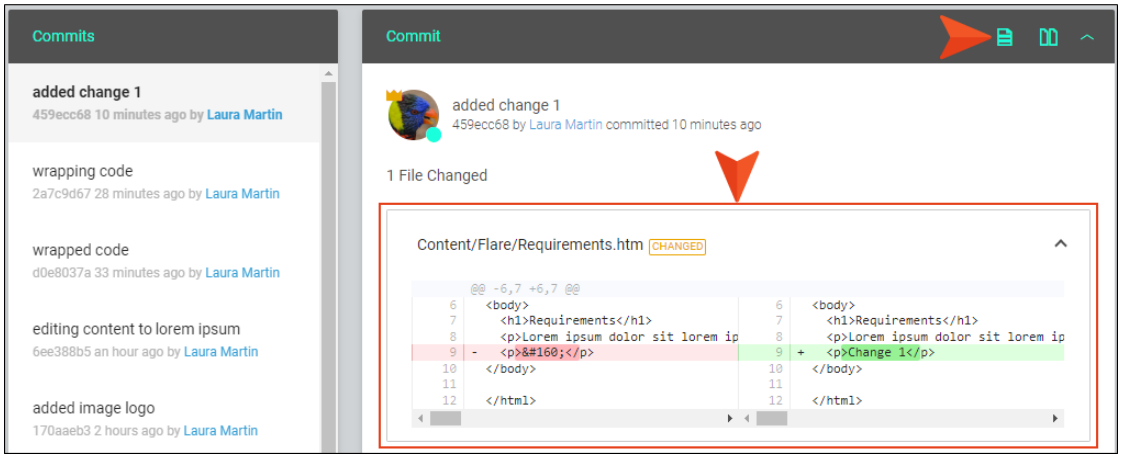


Here is the Commits view:



The Commits view lists all of the changes over time for the file, with the most recent commit at the top.

You can click a row to focus on the code for the file. Buttons let you see the old and new versions in a Unified view or Split view.



CHAPTER 9

Commits

After opening a project in Central, you can see your commits associated with a project. You can also switch between branches (if you are using branching) to see the commits for each.

If permissions are set in Central that allow for authoring project files (i.e., add, edit, upload files), you can update files directly in Central. Changes that you want to keep need to be committed to the project. This is done through the Files page and Central's editor.

This chapter discusses the following:

Permission Required?

No special permission is required for this activity. All authors who are associated with the project are allowed.

What are Commits?

A commit is a collection of changes that are pushed to a project's repository. You can commit files locally using Flare's application, or you can commit files working remotely through Central's editor.

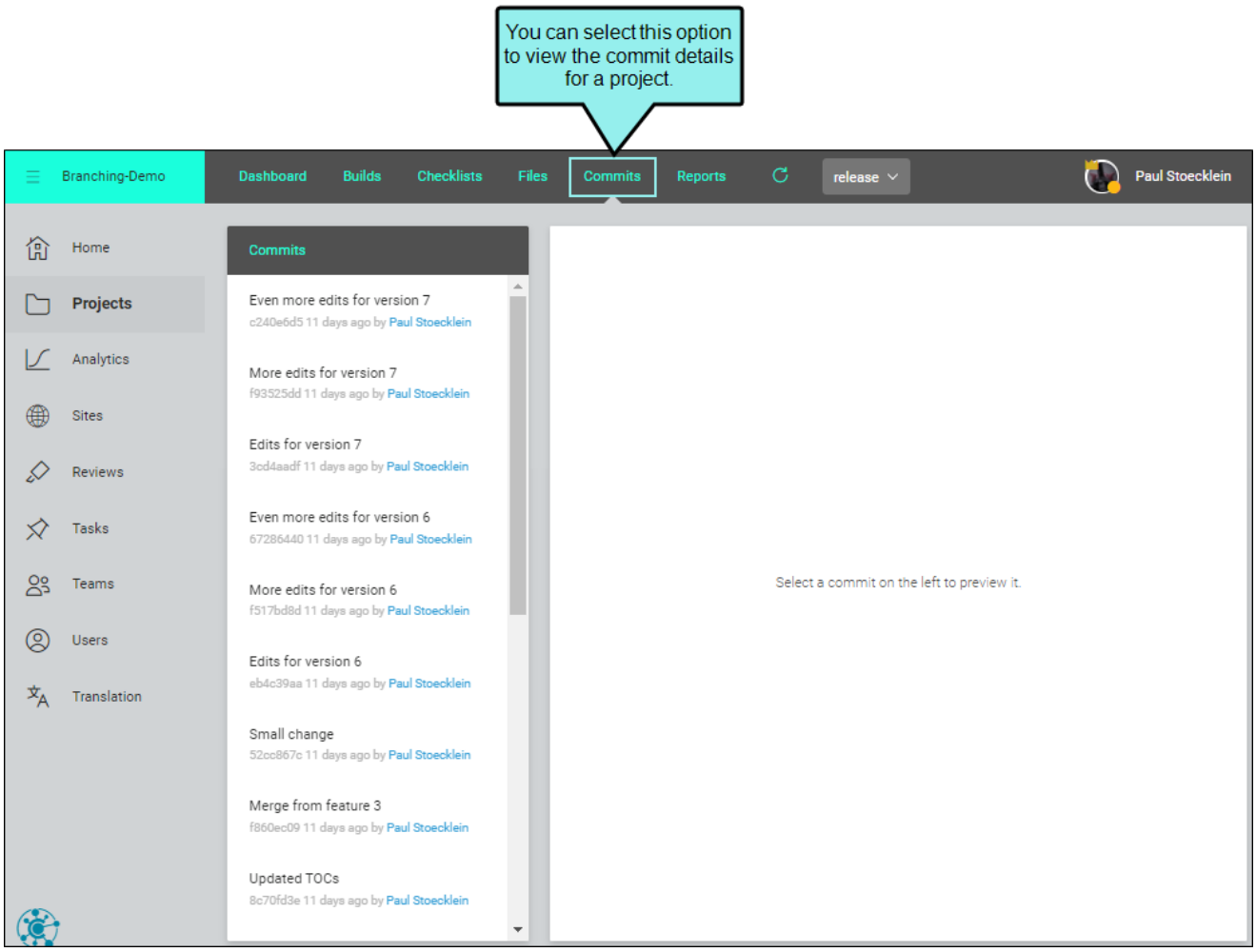
Whether you are working in Central or Flare, adding comments to each commit is required. Comments are an easy way to see what types of changes are included in that commit. It is up to you to decide which changes are included in a commit before you continue making further changes that go into the next commit after that.

- Central side A commit can be viewed in Projects using the Files or Commits page.
- Flare side Flare commits need to be pushed up to Central to keep the files synchronized. Once the files are synchronized, you can view a commit in Central.

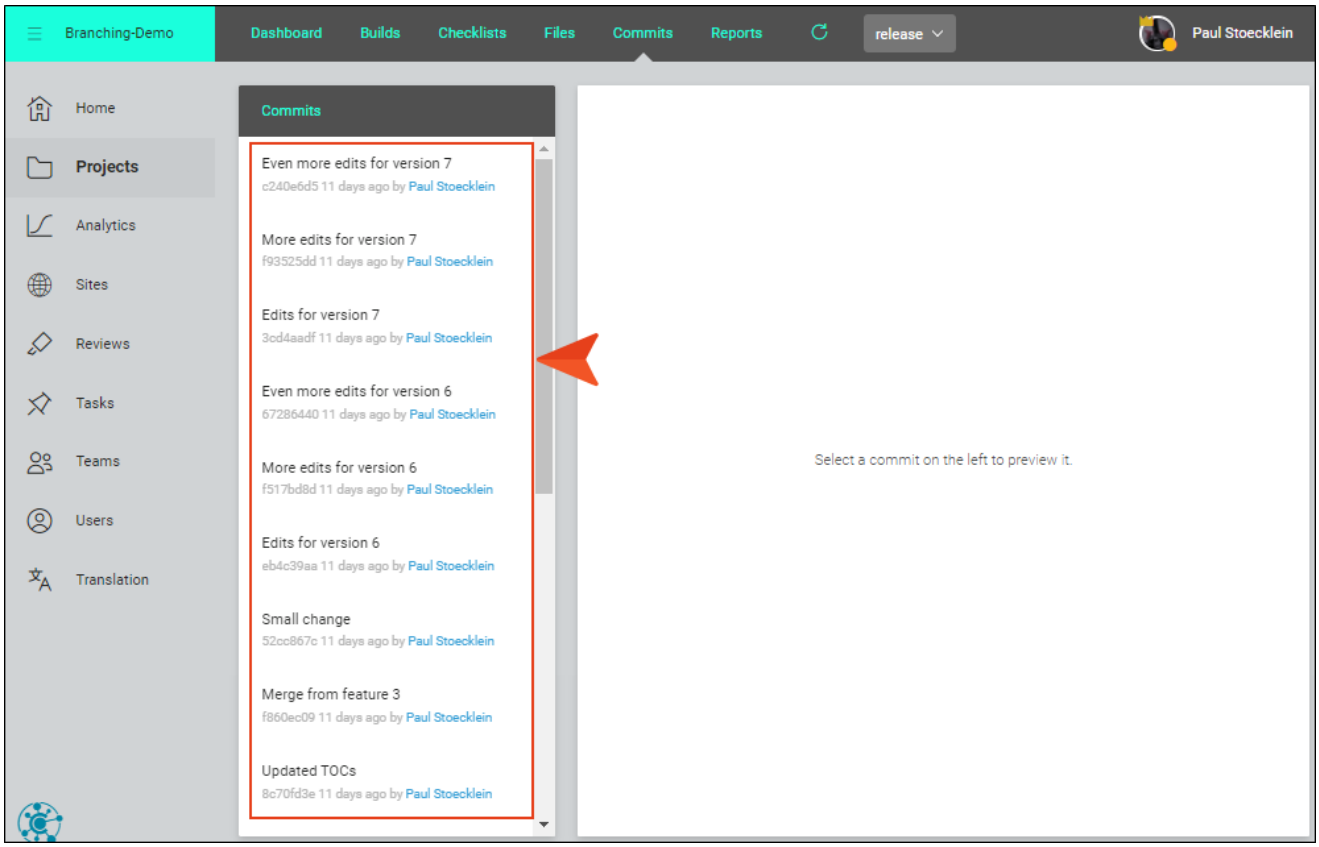
Viewing Commits Page

If you want to view details about your commits on Central, do one of the following from Projects.

- Click **Commits** at the top. The left side of the page lists the commits associated to all project files. If you click the row for the committed file, the Commit view opens to the right with changed items.
- Click **Files** at the top. When you select a file it displays in the editor to the right. Click the **Commits** button in the editor. This lists all the commits pertaining to the selected file. If you click the commit item for a file, it displays the Commits page, and the Commit view opens where you can see exactly what changed.

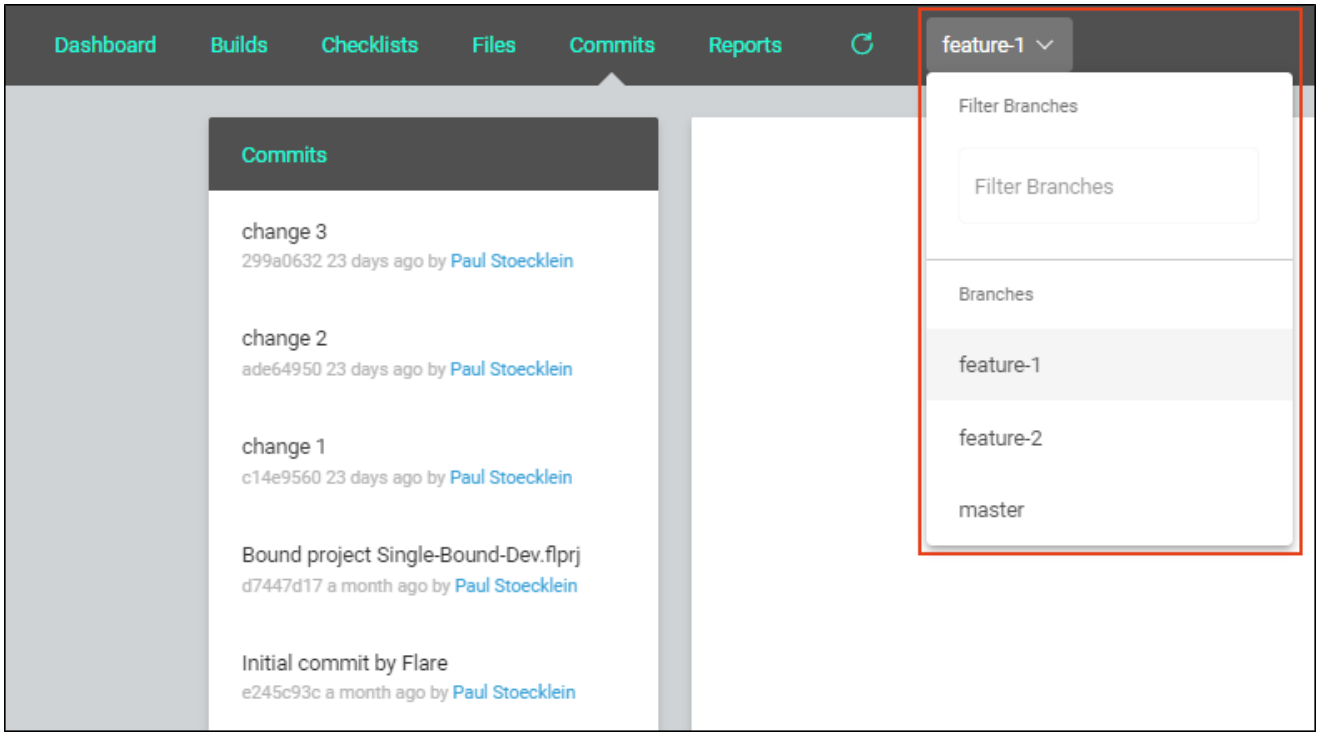


All of the commits for the project are listed, with the most recent one at the top.



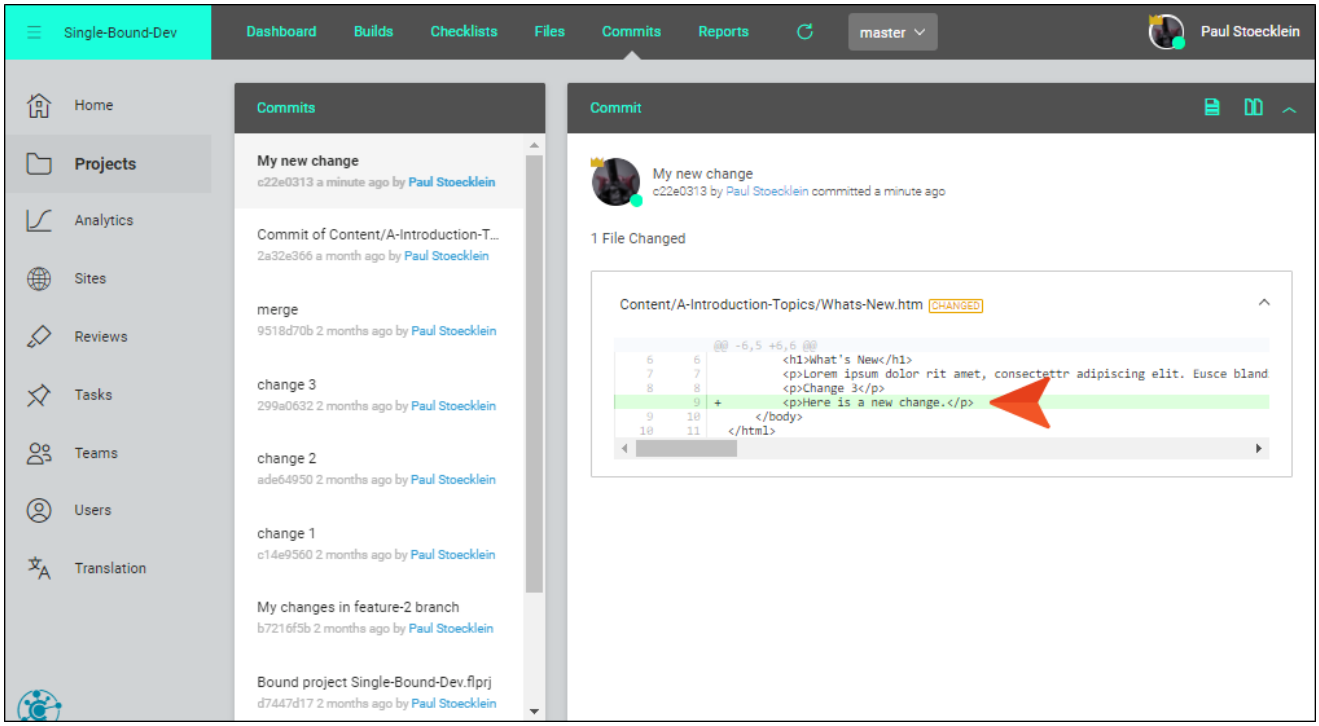
Switching Branches

If you are using branching, you can click the drop-down at the top of the page and switch from one branch to another. The commits specific to that branch will display accordingly on the page. A filter field at the top of the drop-down lets you limit the branches shown.



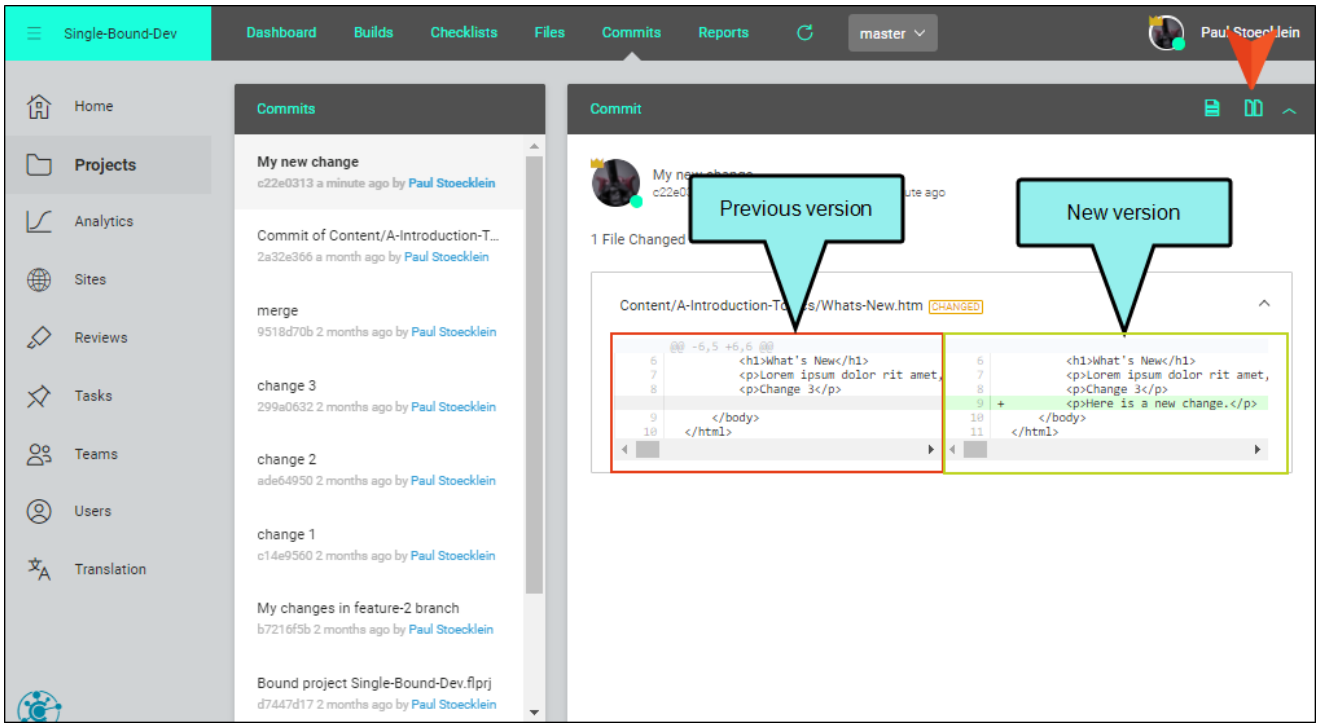
Viewing Individual Commits

If you click any of the commit items, the Commits view opens allowing you to see what changed. You will often see two versions of the files involved in that commit. The background color shows changes. For example, more recent changes have a green background.



You can use the following buttons to switch between a Unified view and a Split view:

In a Split view, the differences look like this:

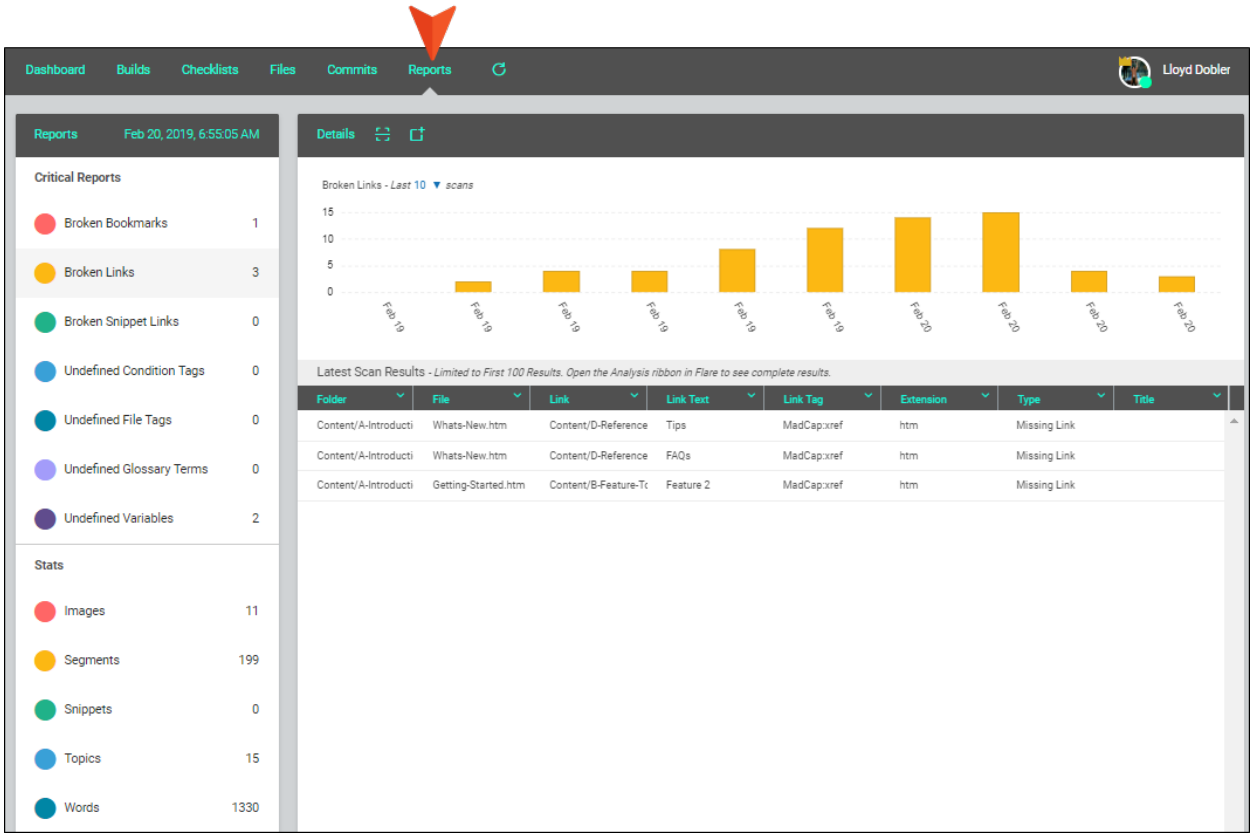


Sometimes a commit might show code on only one side of the page or the other. This usually happens when you have added or deleted a file in the project.

CHAPTER 10

Reports and Statistics

After opening a project in Central, you can click **Reports** at the top of the interface to see various types of reports and statistics. When you select a report or statistical category on the left, a chart and grid are populated on the right with details. At this time, reports support only the master branch, not other branches that you might have created.



This chapter discusses the following:

Opening the Reports Page

You can access the Reports page after opening a project in Central, or by using a widget.

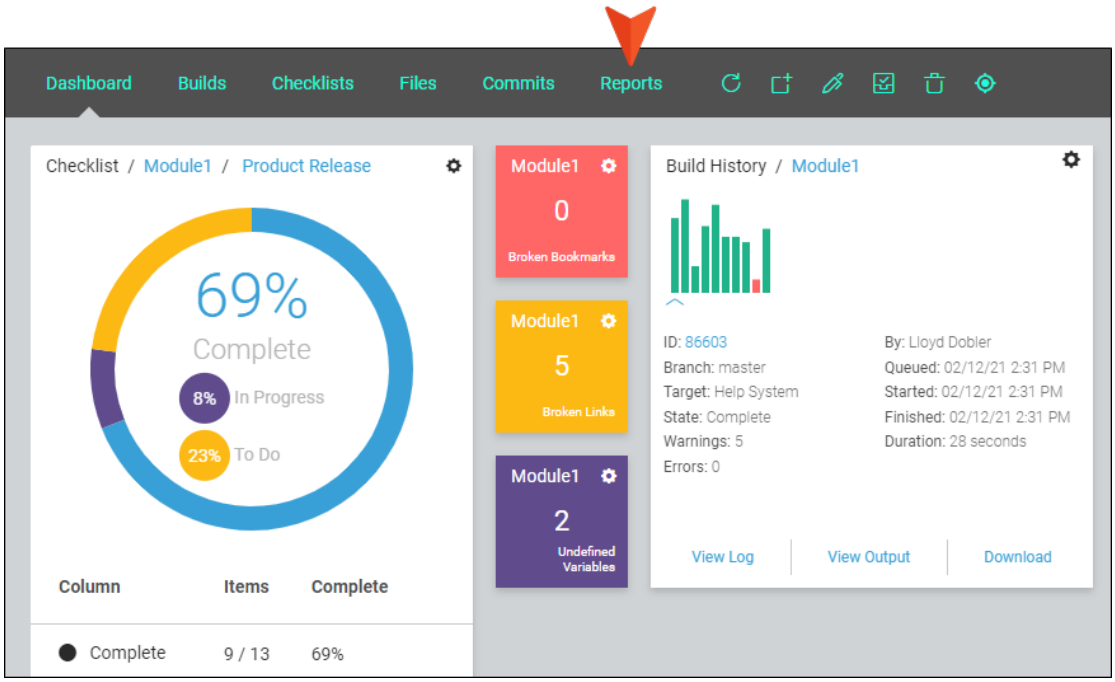
Permission Required?

No special permission is required for this activity. All authors who are associated with the project are allowed.

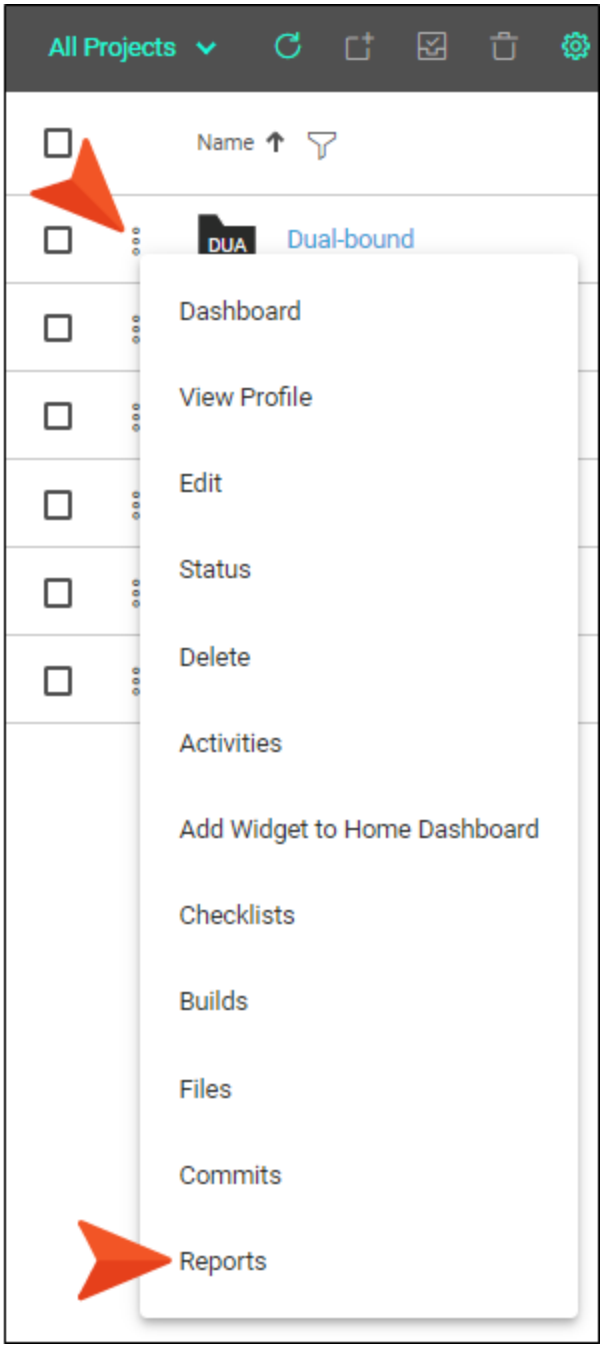
Methods for Opening the Reports Page

There are multiple ways to open the Reports page to view it:

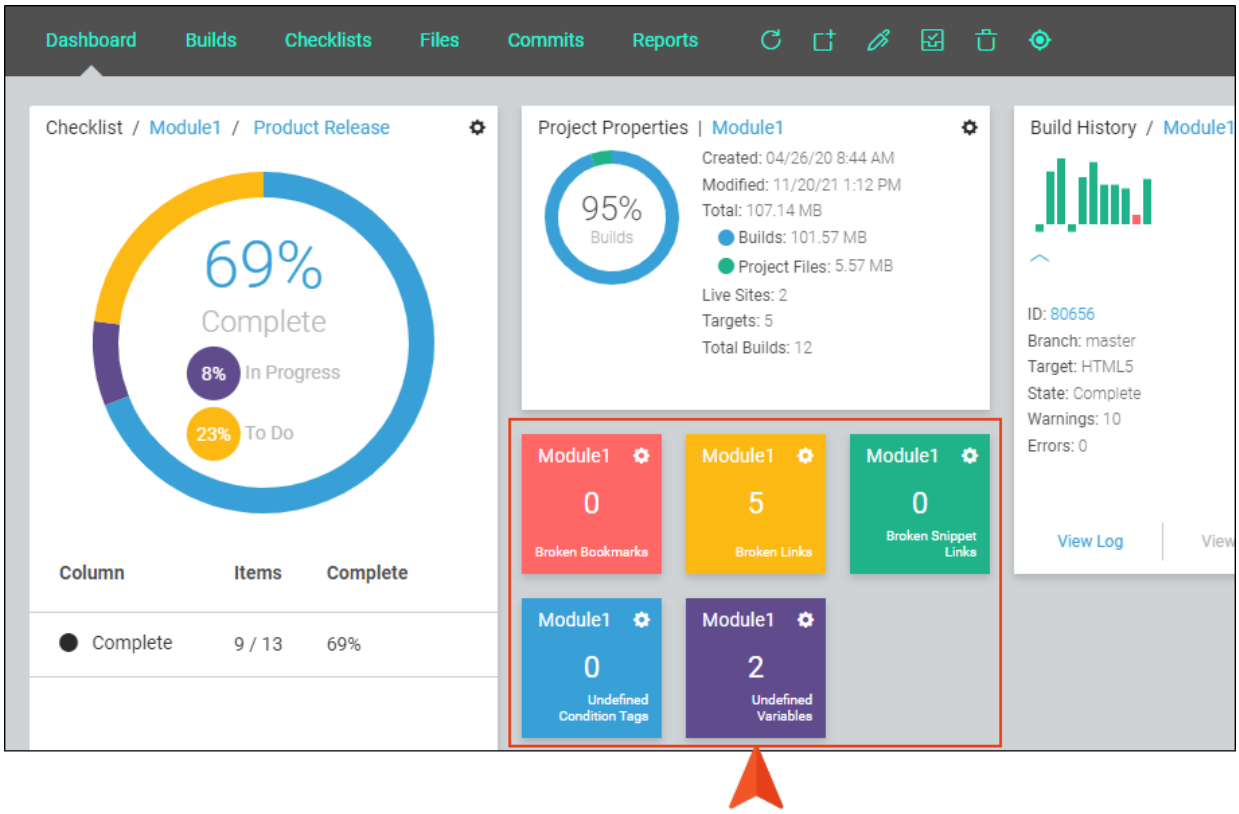
- **Reports Option on Projects Page** After opening a project in Central, you can click **Reports** at the top.



- **Project Grid Context Menu** On the Projects page, you can click the three dots on the left side of the row and select **Reports**.



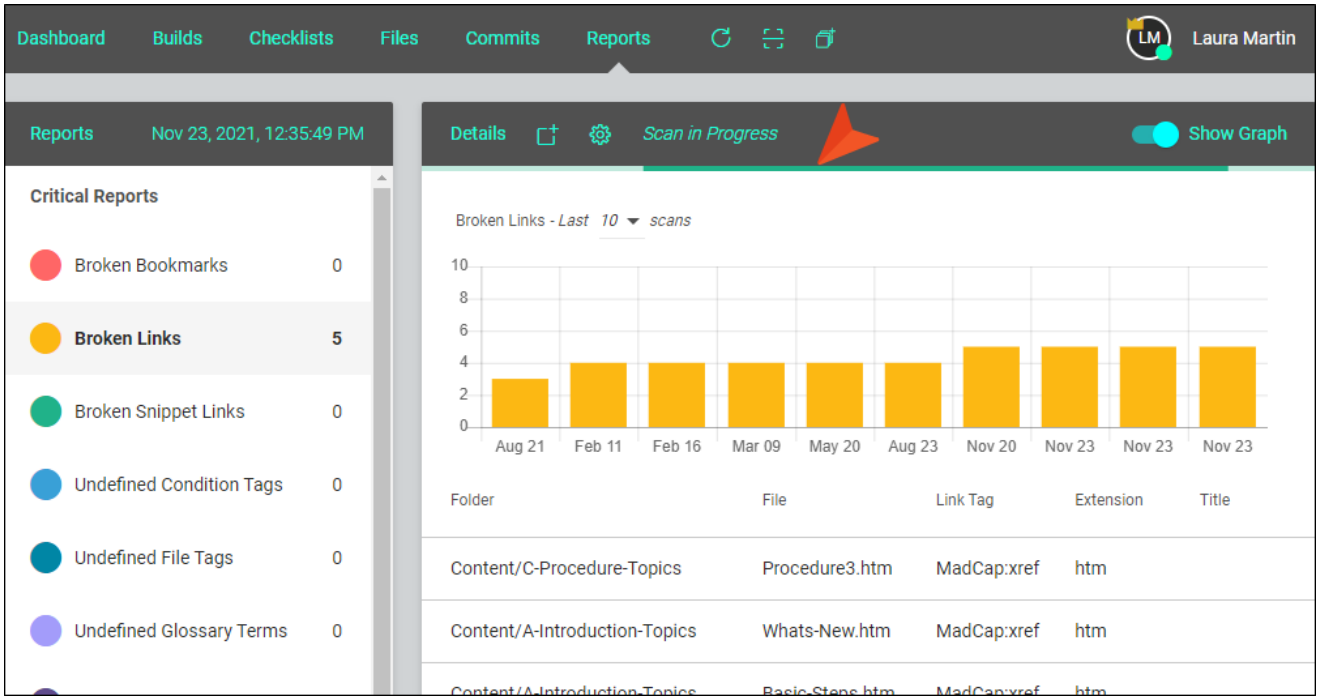
- **Report Widgets** If you have added Report widgets to a dashboard, you can click any of the square objects related to a specific type of report.



NOTE A refresh button is available in the toolbar on the Reports page. This option is not necessary to populate the chart and grid. However, the project activity sometimes does not automatically refresh after a scan; therefore, you can click the refresh button to update it.

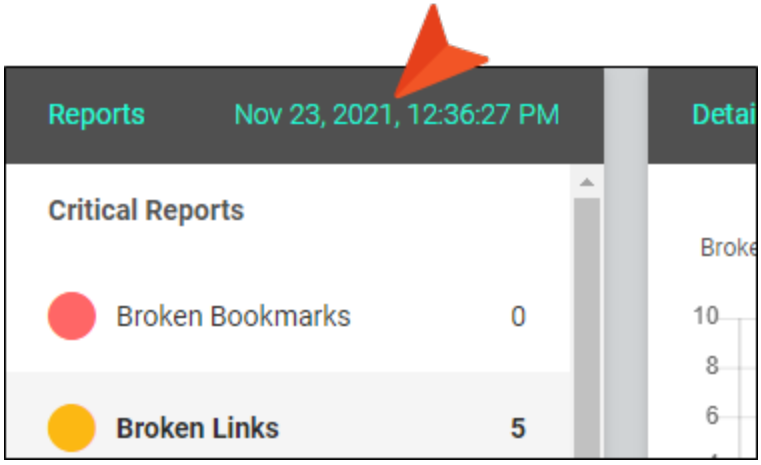
Automatic or Manual Scans

Whenever you push changes for a project to Central, a new scan will be automatically initiated. While the scan is in process, a green progress bar will move across the top.

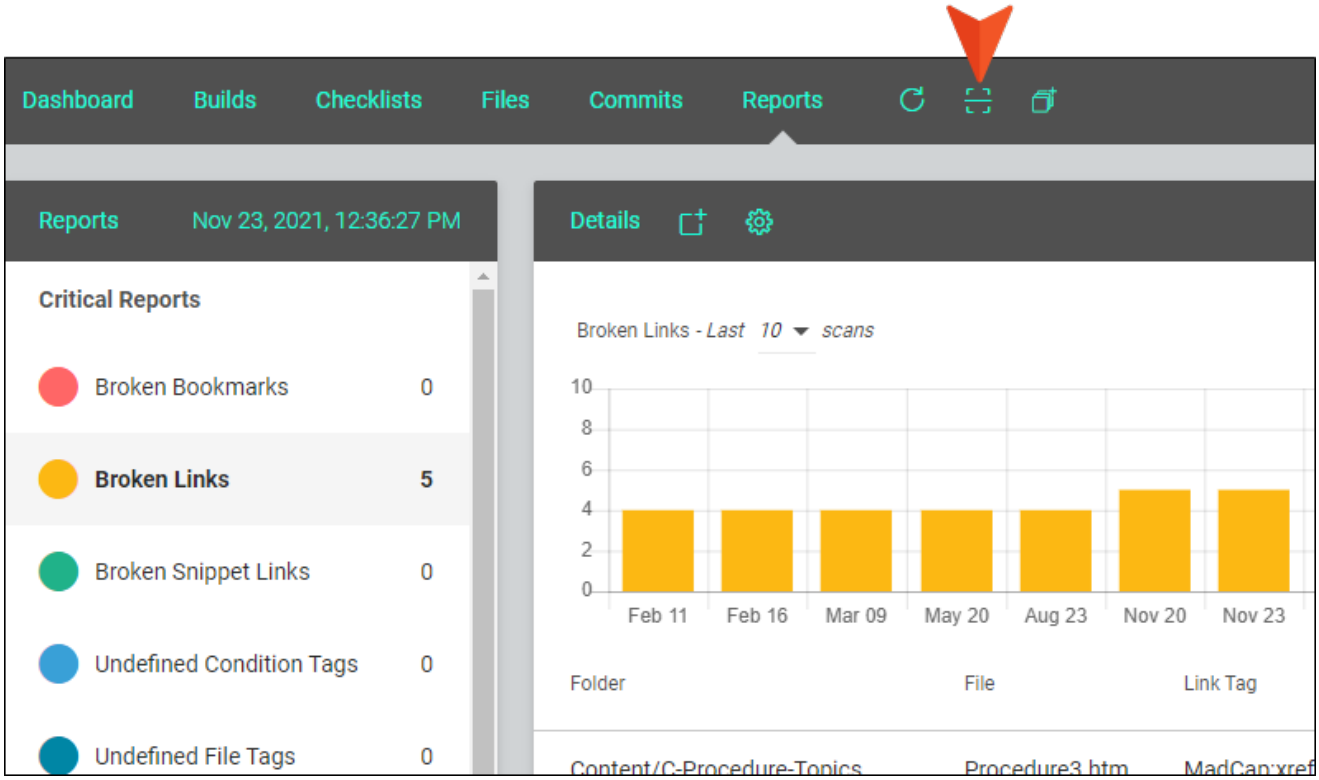


When you no longer see the green bar, the scan is finished. Larger projects will take longer to scan.

The last time the project was scanned is indicated in the upper-left part of the page, next to the Reports pane heading.



You can also click in the main toolbar to initiate a scan manually whenever you like. However, make sure you first have the Scan permission to do this.



NOTE This feature is currently limited to projects smaller than 1 GB. Projects larger than that will not be scanned for reports or statistics.

NOTE If you already had a Flare project uploaded before this feature was introduced, you will not initially see data on the Reports page. You must either perform a manual scan or push changes for that project to Central first. After that, the Reports page will populate.

Selecting and Viewing Reports and Statistics

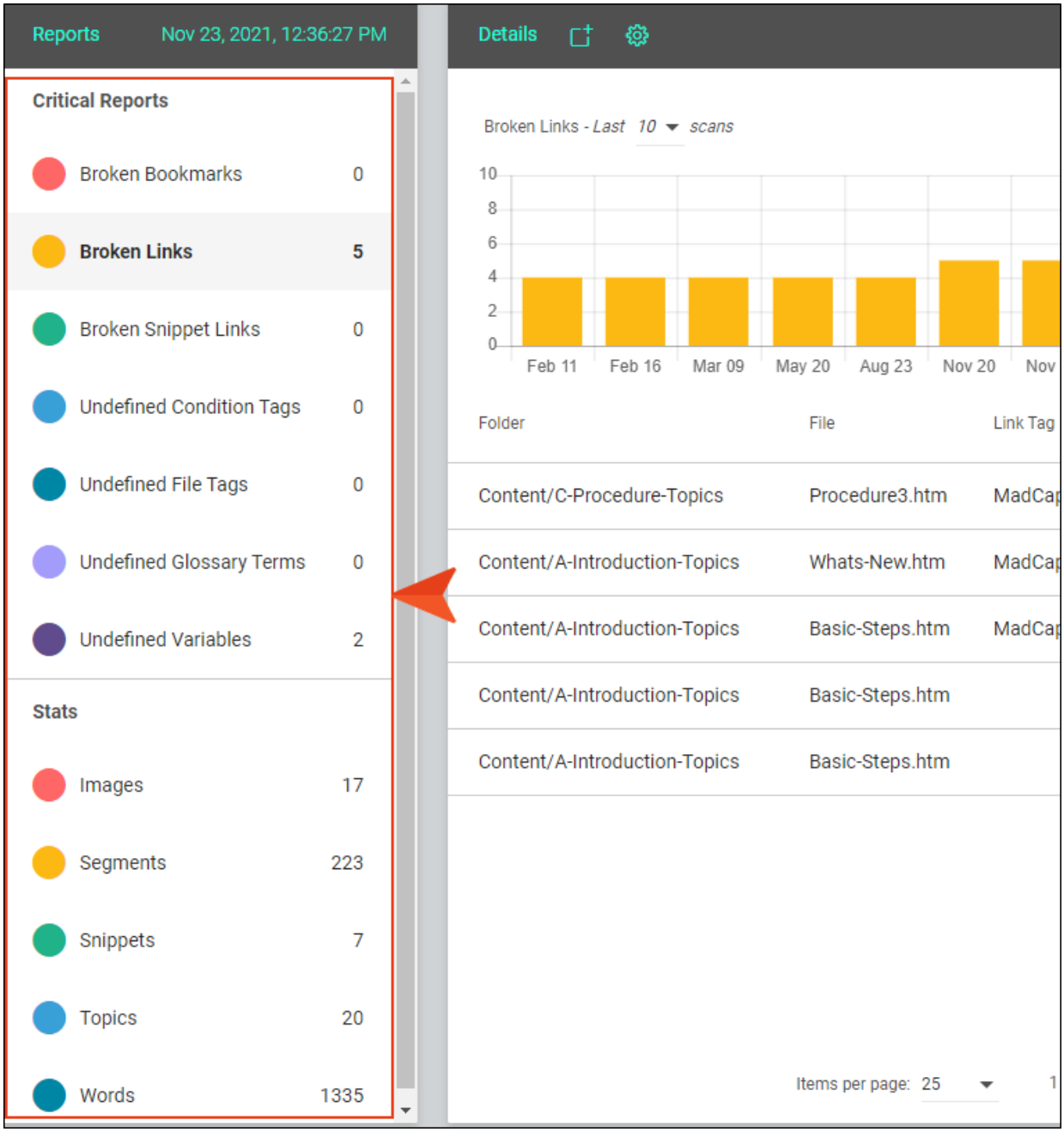
From the Reports page, you can open various kinds of reports and statistics to view details.

Permission Required?

No special permission is required for this activity. All authors who are associated with the project are allowed.

How to Select and View Reports and Statistics

1. After opening the Reports page, notice the various types of reports and statistics on the left side of the page.



CRITICAL REPORTS

The following reports are provided. These are all reports that are considered "critical," in that your output will contain missing or incorrect information if you do not fix the issues.

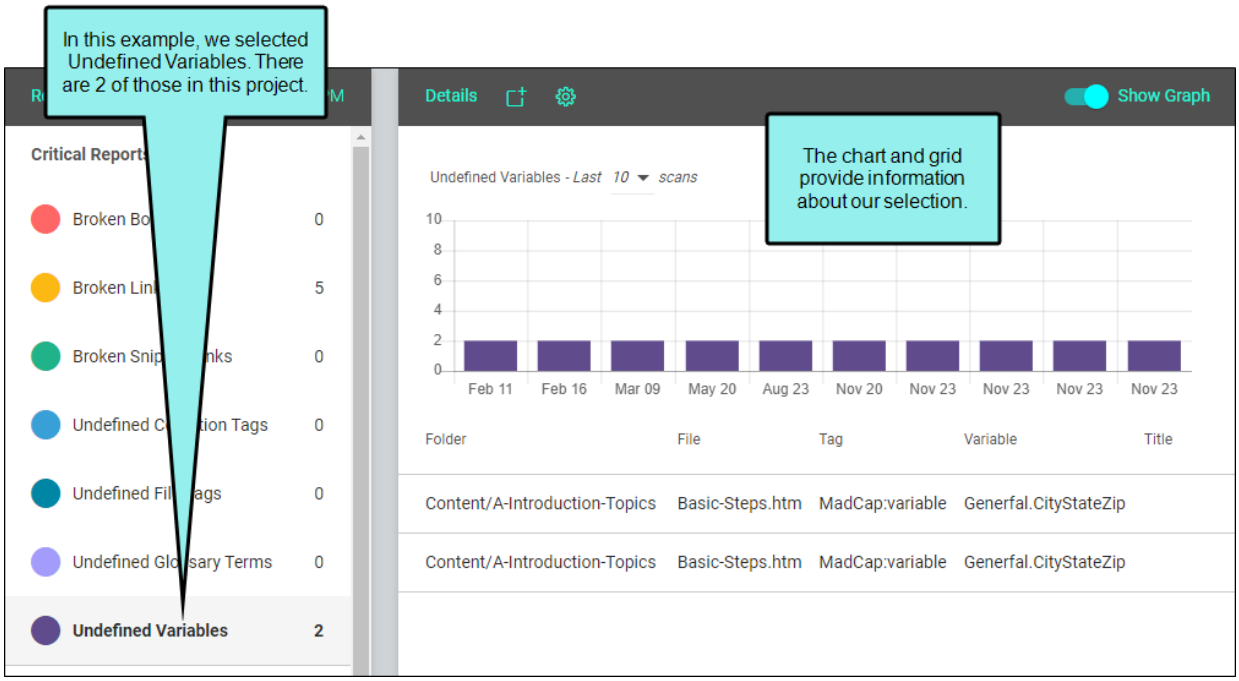
- Broken Bookmarks These are links to bookmarks that are broken. In Flare, you need to fix the issue (e.g., remove or replace the bookmark link).
- Broken Links These are links—such as cross-references—that are broken. In Flare, you need to fix the issue (e.g., remove or replace the link).
- Broken Snippet Links These are links to snippets that are broken. In Flare, you need to fix the issue (e.g., remove or replace the snippet).
- Undefined Condition Tags These are conditions that have been applied in the project, but the name or location of those conditions cannot be found. This might occur, for example, if you have applied a condition in a topic and then later change the name of the tag or condition tag set. The old tag or set name is still used at the location where it was previously applied in the topic. In Flare, you need to fix the issue (e.g., rename the condition tag or set, remove the tag where it was applied).
- Undefined File Tags These are file tags that have been applied in the project, but the name or location of those file tags cannot be found. This might occur, for example, if you have associated a file tag with a topic and then later change the name of the tag or file tag set. The old tag or set name is still used for the topic. In Flare, you need to fix the issue (e.g., rename the file tag or set, remove the tag where it was applied).
- Undefined Glossary Term Links These are glossary term links in your content that are broken. This might occur, for example, if you have created a new term in a glossary and then inserted that term into a topic as a glossary term link. Later, you remove that term from the glossary. You now have a glossary term link in a topic, but the term no longer exists in the glossary. In Flare, you need to either remove the glossary term link from the content, or you need to add the term to the glossary again.
- Undefined Variables These are variables that have been inserted in the project, but the name or location of those variables cannot be found. This might occur, for example, if you have inserted a variable in a topic and then later change the name of the variable or variable set. The old variable or set name is still used at the location where it was previously inserted in the topic. In Flare, you need to fix the issue (e.g., rename the variable or set, remove the variable where it was applied).

STATS

In addition, you can view statistics (i.e., the count) for the following in each project:

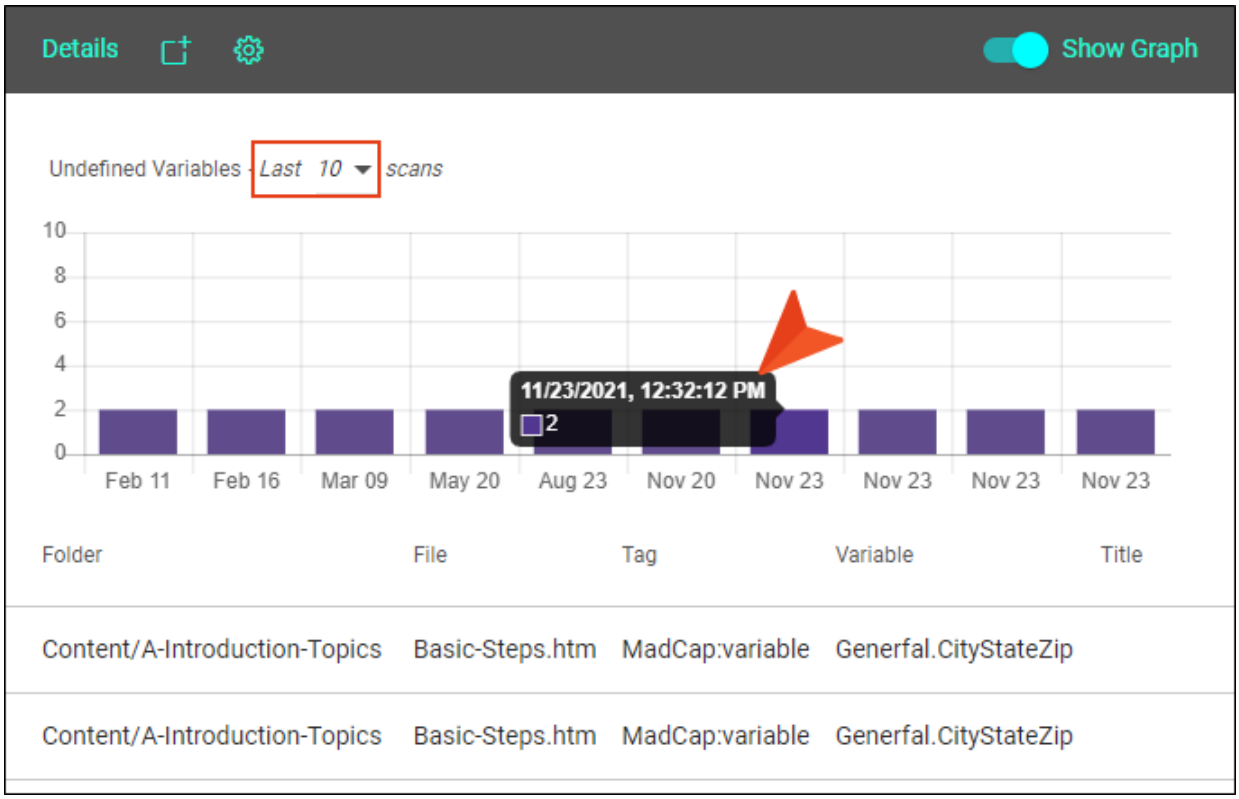
- Images
- Segments (i.e., phrases or sentences)
- Snippets
- Topics
- Words

2. To see data for a particular report or statistical category, click it. The area to the right is then populated with a chart and grid.



The chart shows a count history of the item, with each bar representing a different push of changes to Central. You might see many bars with the same date, because you might have pushed changes to Central multiple times throughout the day.

A drop-down lets you choose how many of the most recent scans to show in the chart (e.g., 1, 10, or 25). If you hover over a bar, the date, time, and count shows for a particular scan of the selected item.



Unlike the chart, the grid below it does not show a history of pushes. Instead, it shows results for the most recent scan and displays all of the instances of the selected item in the project. For example, if you selected the Broken Links report, each row of the grid will show a broken link that was found in the latest scan of the project. The Folder and File columns on the left tell you where the issue was found. The rest of the columns are different for each type of report and provide information to track down the problem.

In addition to the folder and file location, the grid for broken links shows information such as each broken link path, the link text, and the type of issue encountered.

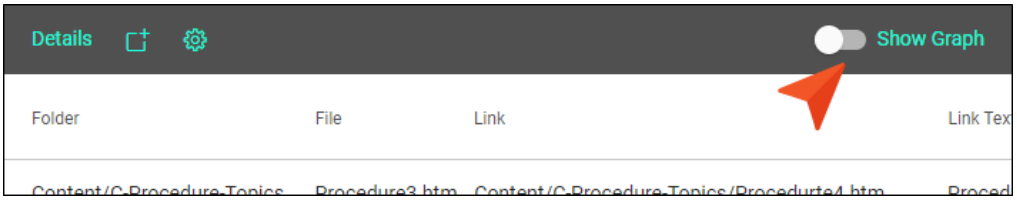
Folder	File	Link	Link Text	Link Tag	Extension	Type	Title
Content/C-Procedure-Topics	Procedure3.htm	Content/C-Procedure-Topics/Procedurte4.htm	Procedure 4	MadCap:xref	htm	Missing Link	
Content/A-Introduction-Topics	Whats-New.htm	Content/B-Feature-Topics/Featurfes.htm	Features	MadCap:xref	htm	Missing Link	
Content/A-Introduction-Topics	Basic-Steps.htm	Content/A-Introduction-Topics/Getting-Sftarted.htm	Getting Started	MadCap:xref	htm	Missing Link	

Hover to the right of any column heading for an arrow to display. Click the arrow to sort the information according to that column.

Folder	File ↑	Link	Link Text	Link Tag	Extension	Type	Title
Content/C-Procedure-Topics	Procedure3.htm	Content/C-Procedure-Topics/Procedurte4.htm	Procedure 4	MadCap:xref	htm	Missing Link	
Content/A-Introduction-Topics	Whats-New.htm	Content/B-Feature-Topics/Featurfes.htm	Features	MadCap:xref	htm	Missing Link	
Content/A-Introduction-Topics	Basic-Steps.htm	Content/A-Introduction-Topics/Getting-Sftarted.htm	Getting Started	MadCap:xref	htm	Missing Link	

When you select a statistical category, the grid will show a count history for all of the categories. However, the chart above represents the count history for only the selected item.

NOTE The chart and grid show by default for a particular report. You can hide the visual display of the chart by clicking the Show Charts option.



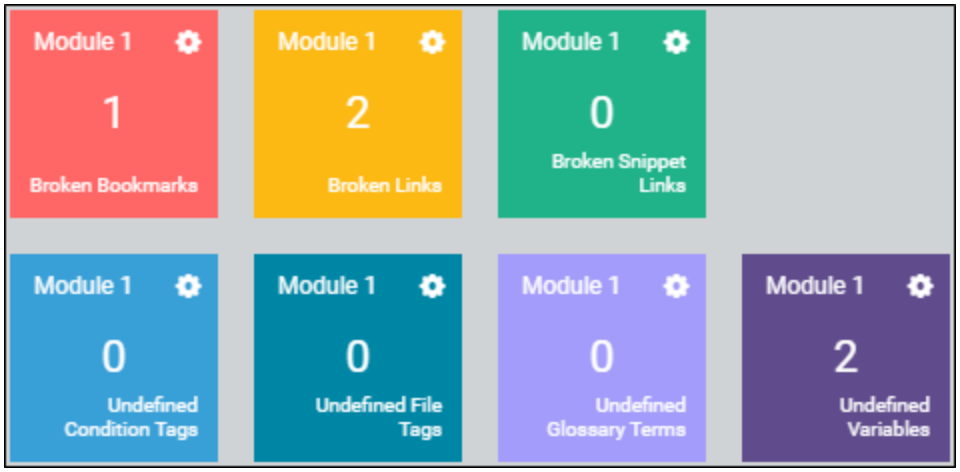
When a Scan Shows Critical Issues

When you view a report and notice critical issues, you should do the following:

1. Use the report to identify the location in Flare where the issue exists.
2. Open Flare and fix the problem.
3. Commit and push your changes to Central.
4. Open the Reports page again and ensure that no more critical issues are present. (Alternatively, you can use the Analysis ribbon in Flare to see the updated reports.)

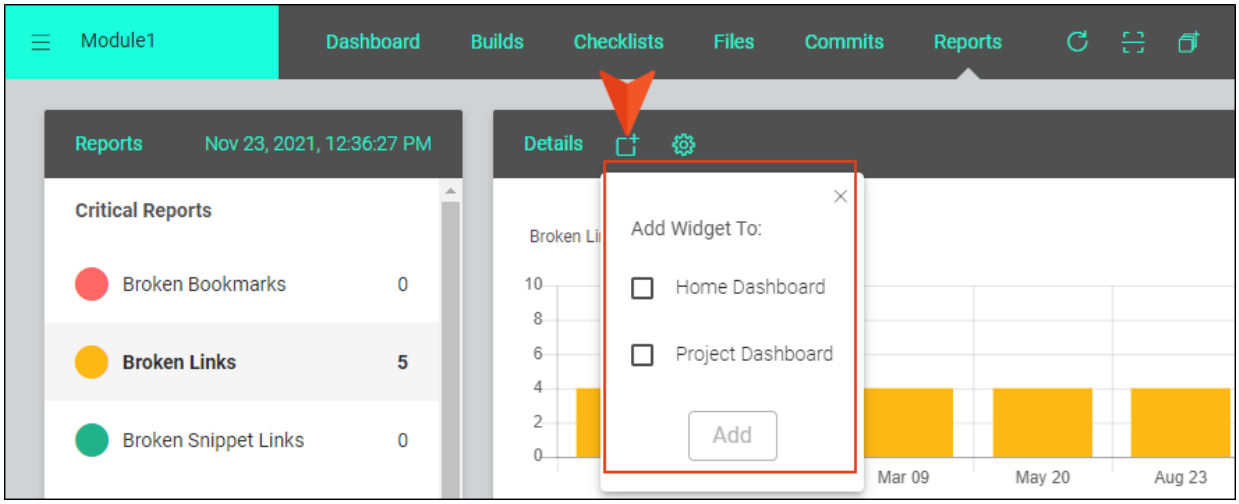
Report Widgets

You can add Report widgets to your Home or Project dashboards, based on any of the critical reports. Each widget is a small square with a number that indicates the number of issues found in the project.

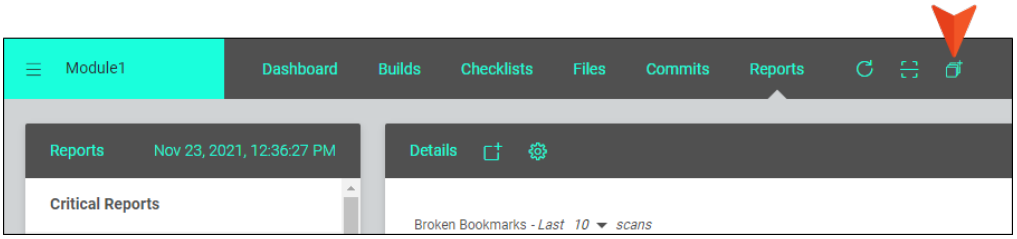


How to Add Report Widgets

1. Open the Reports view, and in the toolbar click . From the context menu, you can select to add the widget for the active report to the Home or Project dashboard(s).



Alternatively, you can click this same button in the toolbar of the Home or Projects dashboard.

> TIP In the main toolbar you can click to add multiple report widgets to the Home or Project dashboard(s).
>
> 
>
>
> It is recommended to use this option in the Reports view because it lets you add widgets for multiple kinds of critical reports at the same time. On the other hand, if you are on a dashboard, you will need to repeat the steps for adding each type of report, plus you will need to filter each blank widget afterward to choose a report for it.

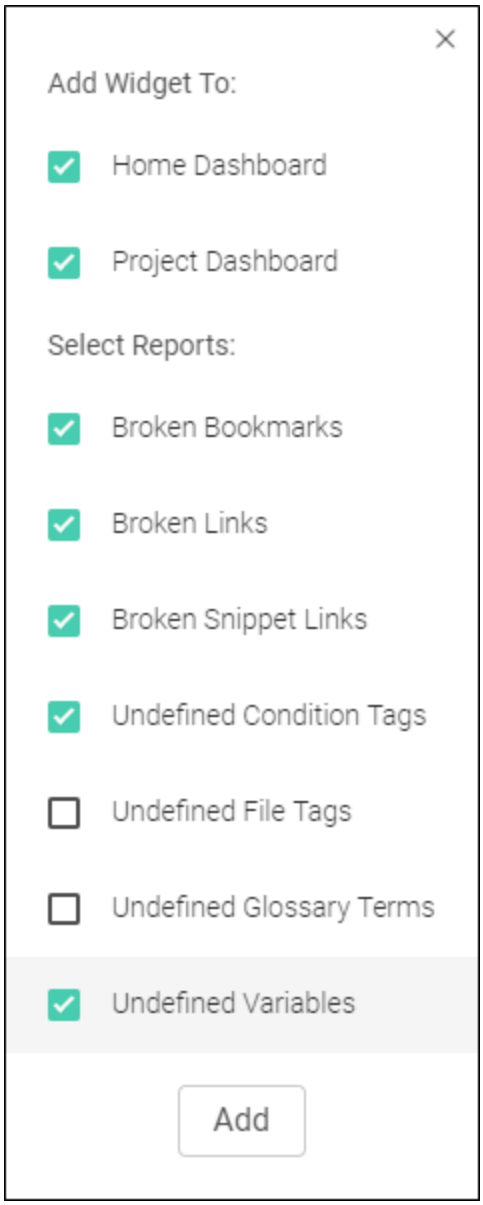
2. Do one of the following in the dialog that displays, depending on the method you are using:

METHOD 1: REPORTS VIEW (RECOMMENDED)

If you click in the Reports view, select **Home Dashboard**, **Project Dashboard**, or both, depending on where you want to see the widgets.

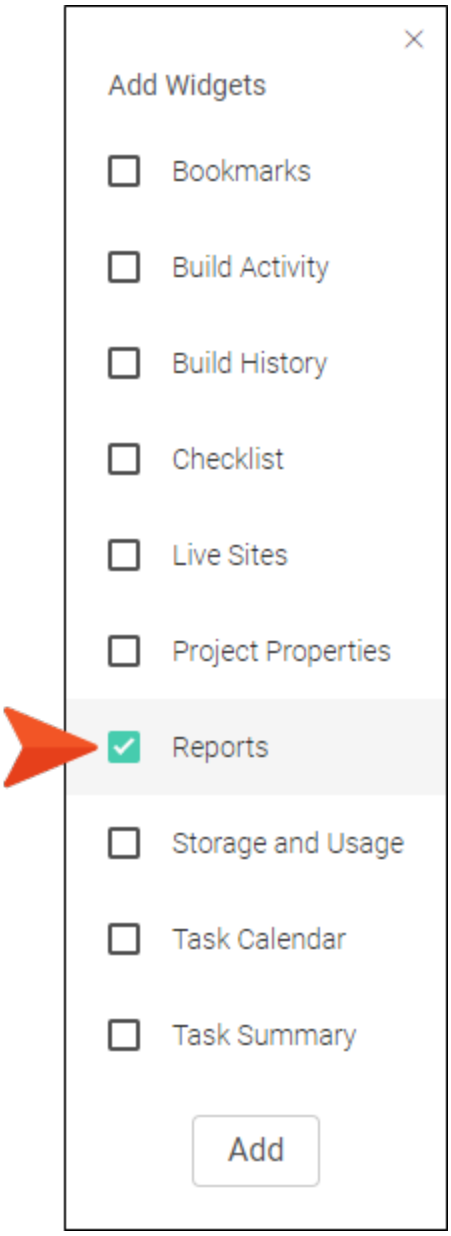
Then, below those selections, choose which types of reports you want to include in widgets.

After that, click **Add**.



METHOD 2: HOME OR PROJECT DASHBOARD

If you click on a dashboard, select **Reports** and click **Add**.



Repeat these steps for each type of report widget you want to add to a dashboard.

If you used the Reports view method, you can go to the dashboard, then click and drag the widgets to move them where you want them.

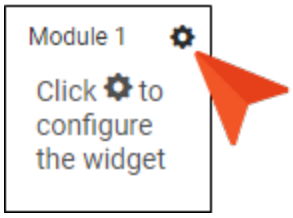
If you used the Dashboard method, the new widgets are initially empty. You need to filter them in order to choose report types. After this, you can click and drag the widgets to move them where you want them on the dashboard.

Regardless of the method, keep in mind that you might need to scroll down to see your new widgets if existing widgets are already taking up all the space toward the top of the dashboard.

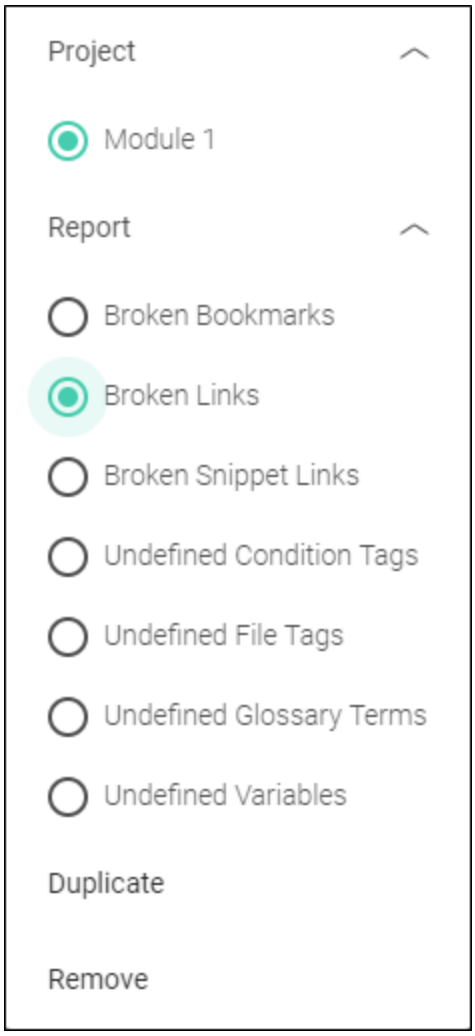
How to Filter Report Widgets

If you added a widget from the Reports view, it should be populated already. But if you have an empty reports widget, use the following steps to choose a report type in order to populate it.

1. On the left side of the interface, select either **Home** or **Projects**, depending on the dashboard where you've added the report widgets. If you selected Projects, you might also need to choose **Dashboard** at the top of the interface.
2. In the empty widget container, click ⚙ in the upper-right corner.



3. In the menu, expand **Project** (if necessary) to choose an uploaded Flare project. Then expand **Report** and select a report type.



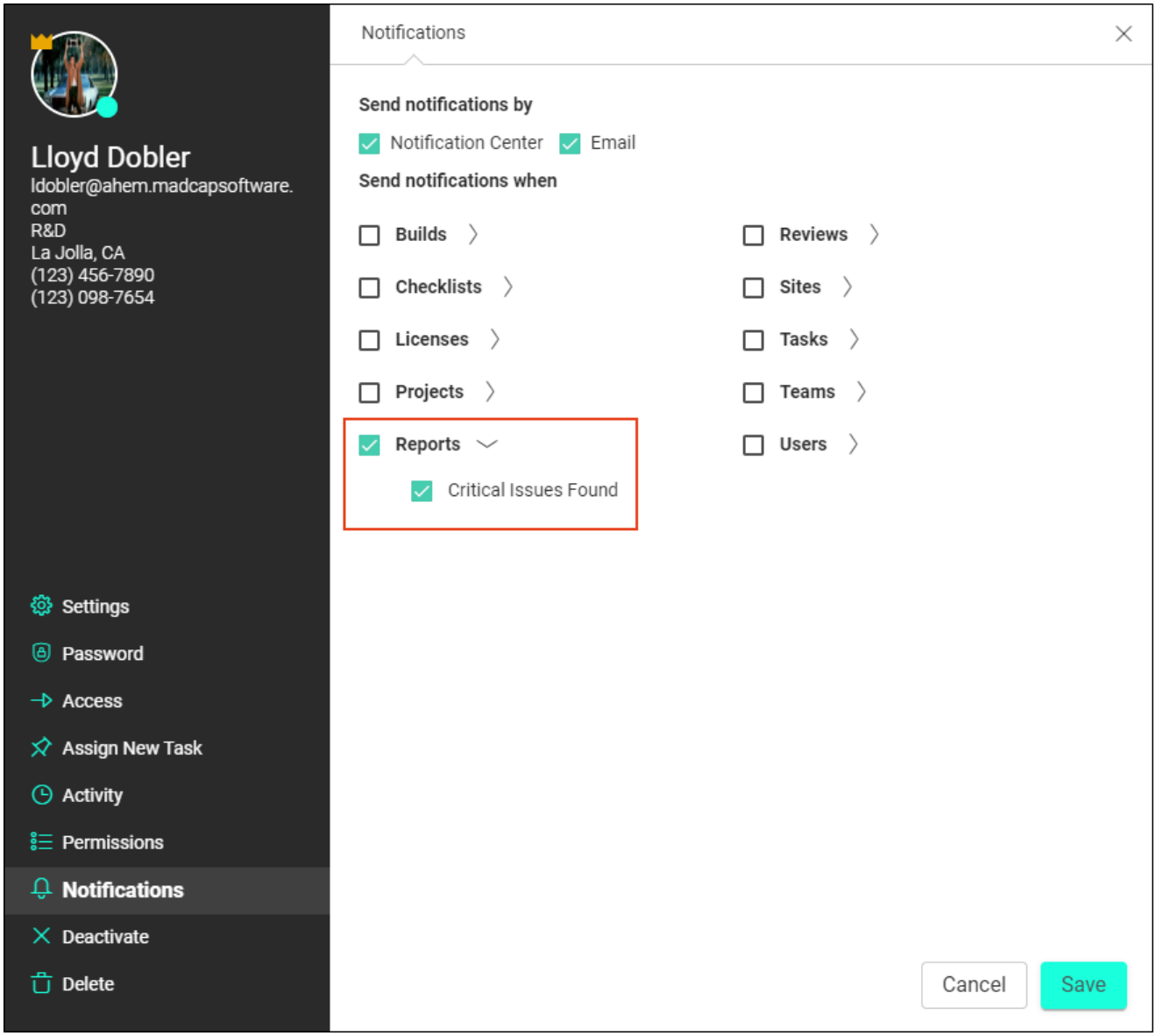
4. Click away from the dialog. The widget is now populated with data for that report type.

What's Noteworthy?

> **TIP** If you click the number on a report widget, the Reports page automatically opens with that report selected.

Setting Notifications for Reports

In your account settings, you can choose when to be notified for specific activities. The user who is notified depends on the activity. For more information, see the online Help.



CHAPTER 11

Reviews

Central supports an effective content review process when working with files from MadCap Flare. Files are bundled in a review package, which is essentially a container that can include as little as one or possibly hundreds of files (e.g., topics, snippets). Review packages can be sent from Flare to Central, reviewed and modified, and individual files are accepted back into Flare. Once all the files by all the reviewers are reviewed, the owner closes the review package.

The streamlined review package workflow enables owners to assess the overall progress of it through grids. They can see recent activity, who's reviewing, status of files or packages, or even create new packages directly in Central. Reviewers can open files for review, modify them in an editor, and submit the files within them to Flare. Files within a review package can be seen as a flat list of files or in a TOC view that provides context and general flow of content for output.

NOTE Since the review packages feature affects the reviews process between both Central and Flare, be sure to check out Flare's online Help. Flare supports an efficient workflow for sending and receiving review packages back and forth with Central.

NOTE The latest version of the Flare application is required for sending review packages. Older versions of Flare support reviews but only for sending and accepting single files for review. Files will be brought into Central as one-file packages. For example, if you have 10 files for review, they will be sent as 10 review packages—the files will not be bundled together as one package.

NOTE If you plan to incorporate sending packages for review from Flare to Central, the project first must be bound to Central.

This chapter discusses the following:

General Information for Reviews

There are various pieces of general information you should know if you plan to use this feature.

Benefits

Benefits of this review workflow include:

- Cloud Review SMEs do not need to download and install any software. The review takes place in the cloud.
- Multi-User Editing Multiple reviewers (e.g., SMEs, authors) can make changes and add comments to the same topic or snippet at the same time.
- Review-Only Interface A lightweight version of the editor means a streamlined interface. You only see options and features that are relevant to the review process.
- Auto-Save and Tracking Changes in the editor are auto-saved as you work. In addition, all changes are automatically tracked so the owner can easily locate edits for approval or rejection.
- Branching Files can be sent for review from a specific Git branch. This lets you keep reviews limited to files that are still in a state of development, as opposed to finished and ready for publication.
- Workflow Management Bundling files into review packages favorably services everyone in the review process workflow. Reviewers (e.g., SMEs) can edit and view content from a flat file list or from a contextual TOC view where files display in a TOC for output that an end user might see. Owners (e.g., authors) can edit but they can also monitor and manage reviews in Central. They have access to customizable package and file grids that show review information and progress.

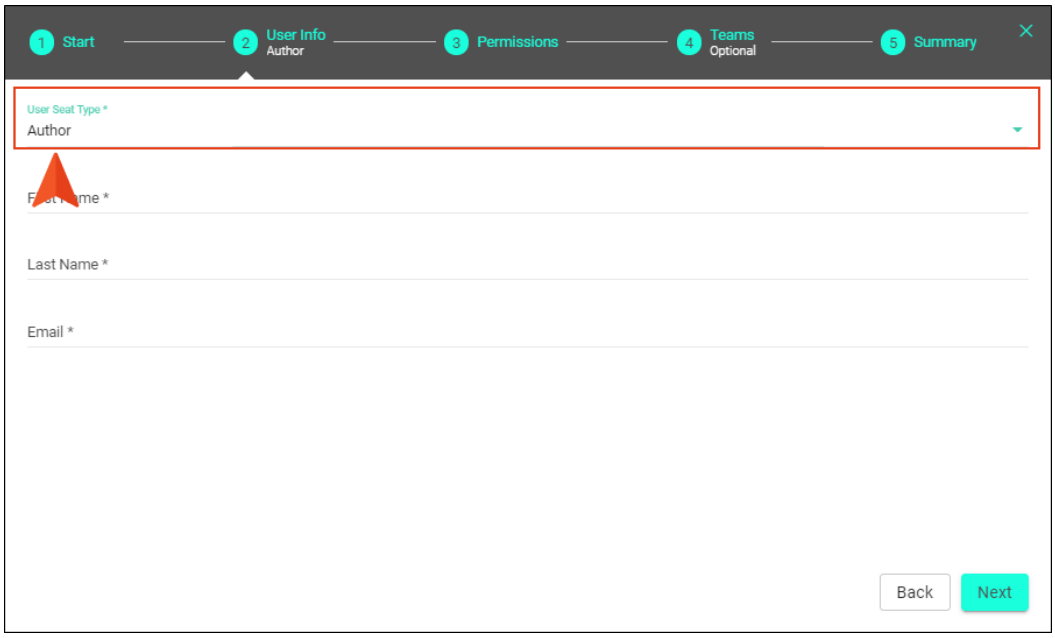
User Types for Reviews

The Central review process involves owners (those originating the process) and reviewers (those making changes and comments in the review package files). Reviewers can be either authors or subject matter experts.

Owners are by default authors and therefore, can also be reviewers. And authors with the Manage Reviews permission can also interact with the data grids (i.e., packages, files) and make changes to review packages. Owners, however, are not required to have the Manage Reviews permission and can modify anything about a package or the files within it.

Inviting Reviewers

When you invite someone to join a Central license, you can choose the user seat type for that person.



- Author An author is an individual who works in Flare projects, creating and editing content. This person can also be the "owner" of a review when they send topics and snippets that need to be reviewed by others. Owners can assign other reviewers with the author seat type and permission to manage reviews. Authors can monitor reviews, access grids for information and progress, and create review packages directly in Central. Along with the reviewers, the author can open and edit files in the Review Editor. Authors with the appropriate permissions can also create and edit content in Central via the project Files page.
- Subject Matter Expert A subject matter expert (SME) is an individual whose main purpose in Central is to review topics and/or snippets sent by an author. Therefore, a SME only sees the parts of the Central user interface that are necessary for reviews.

NOTE A person who has the author user type can function as either the owner (i.e., original author) or a reviewer during the review process.

NOTE What you see and interact with on the Reviews main page largely depends on your user type. For example, a subject matter expert (SME) is mainly exposed to packages for reviewing and editing, but an author has additional access to package and file grids, and can also create packages directly in Central.

Reviews Main Page

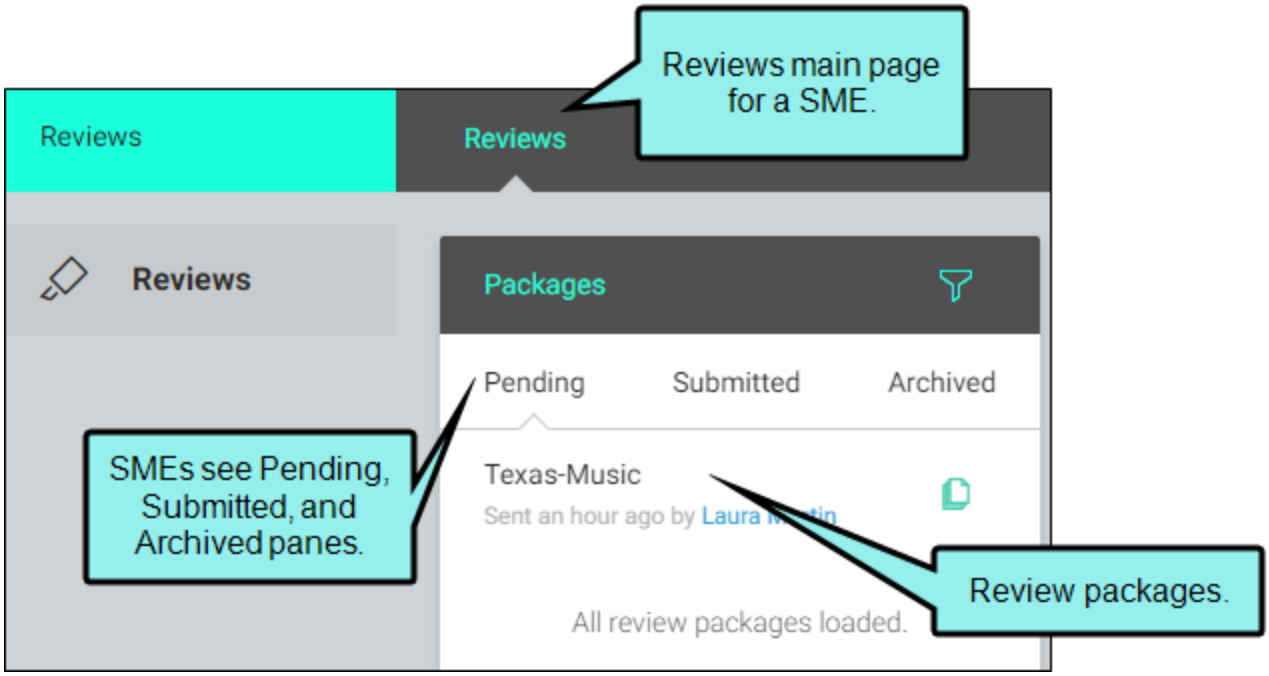
The Reviews main page is the gateway to accessing review packages in Central.

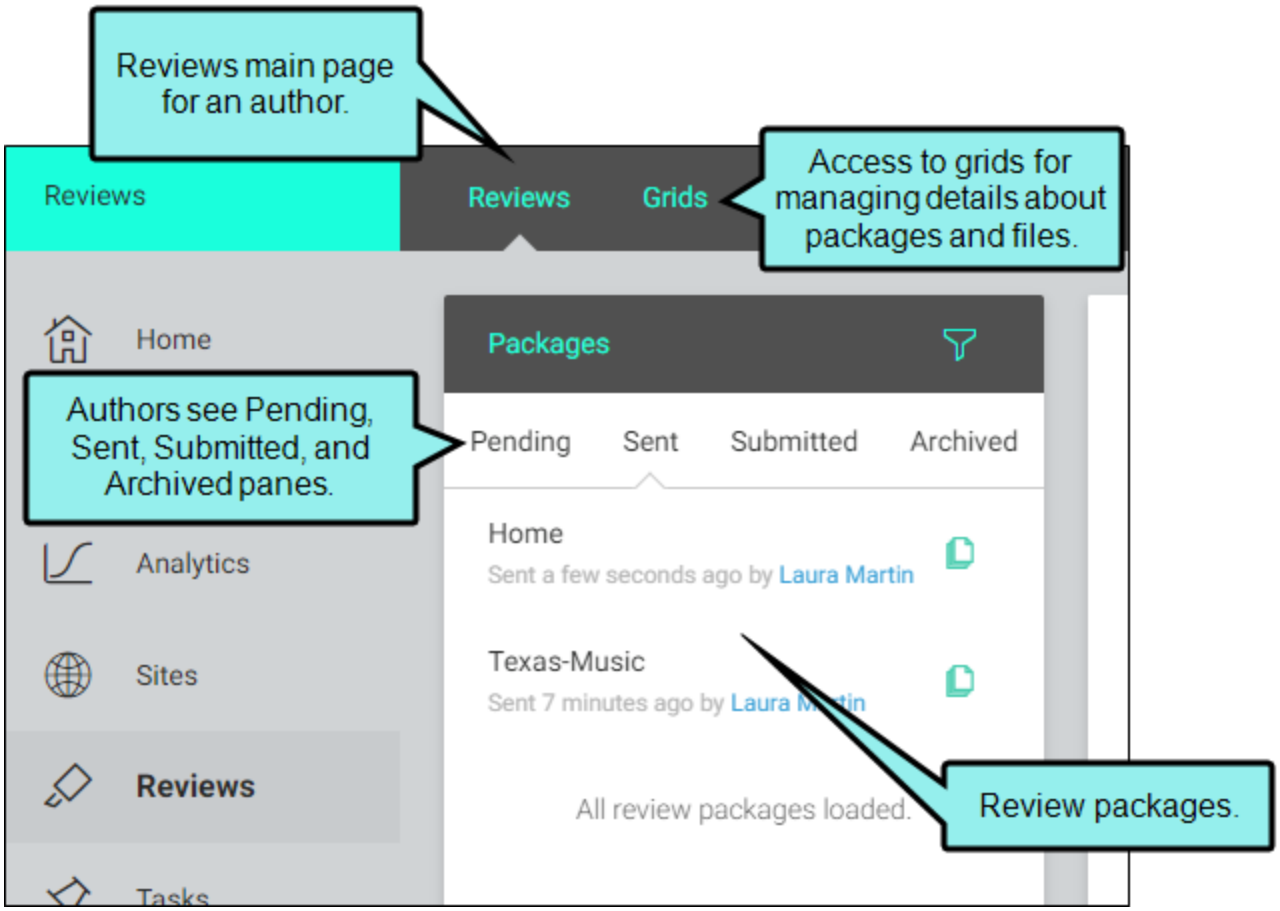
Review Explorer

What you see and interact with on the Reviews main page largely depends on your user type. For example, a subject matter expert (SME) is mainly exposed to packages for reviewing and editing, but an author has additional access to package and file grids, and can also create packages directly in Central.

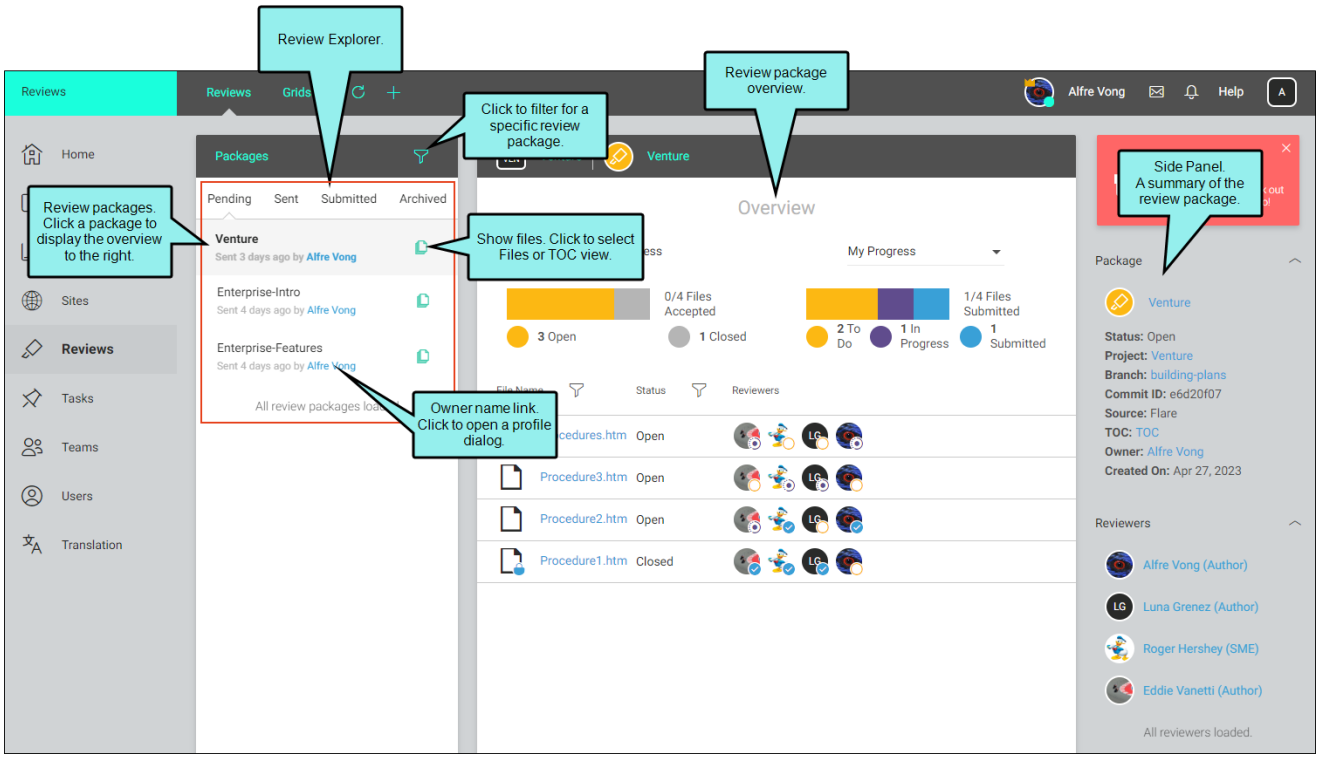
If you are the owner, the files are listed in the Sent pane. If you are a reviewer, the files are listed in the Pending pane.

- Pending This pane displays packages that the reviewer needs to review.
- Sent This pane displays packages that the owner has sent other for review.
- Submitted This pane displays packages that the reviewer has completed and submitted.
- Archived This pane displays packages that have been closed (by the owner). The files within them are read-only.





When you first click the Reviews main page, any uploaded packages that involve you display in the Review Explorer.



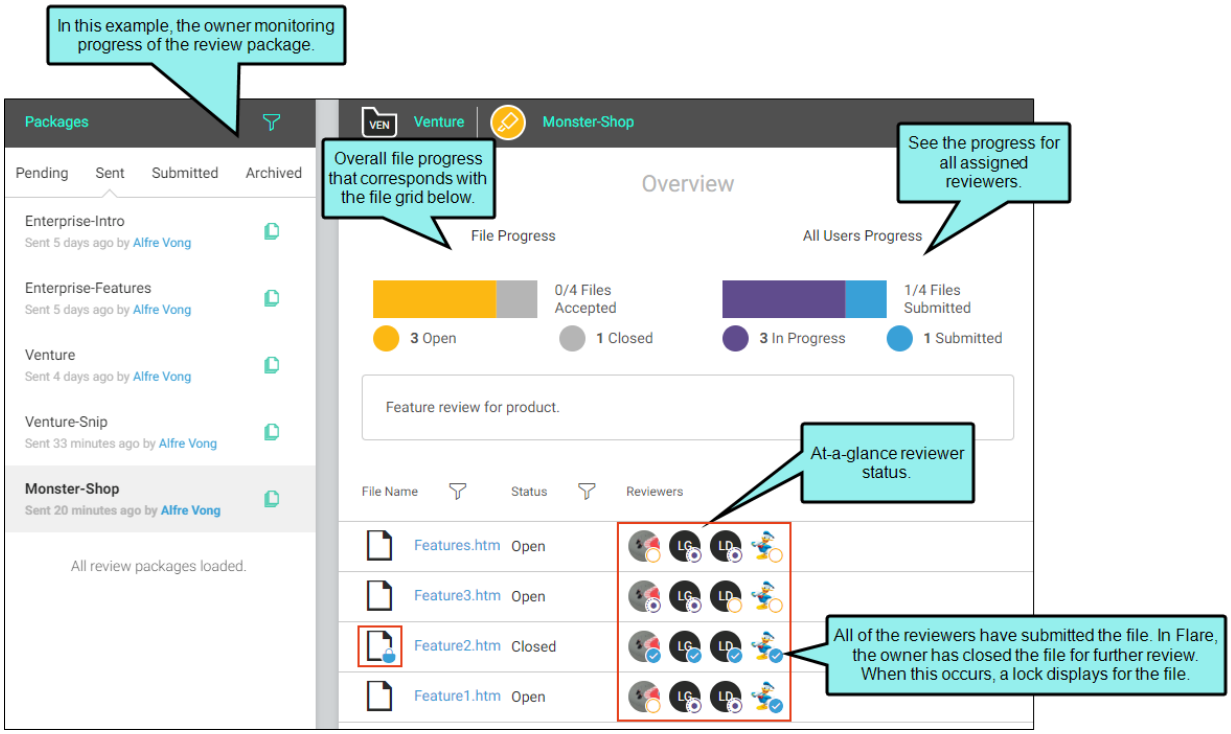
- If you click the review package, an overview page displays to the right with a chart, files, and statuses as they relate to a single package.
- If you click **Show files**, the Review Explorer changes, allowing you to toggle views between Files and Table of Contents (TOC). (If an owner chooses not to associate files with a TOC, then a TOC will not be applicable to your review package.) If you click a file, it displays in the Review Editor to the right.
- If you click the owner name link, a profile dialog opens with owner information.

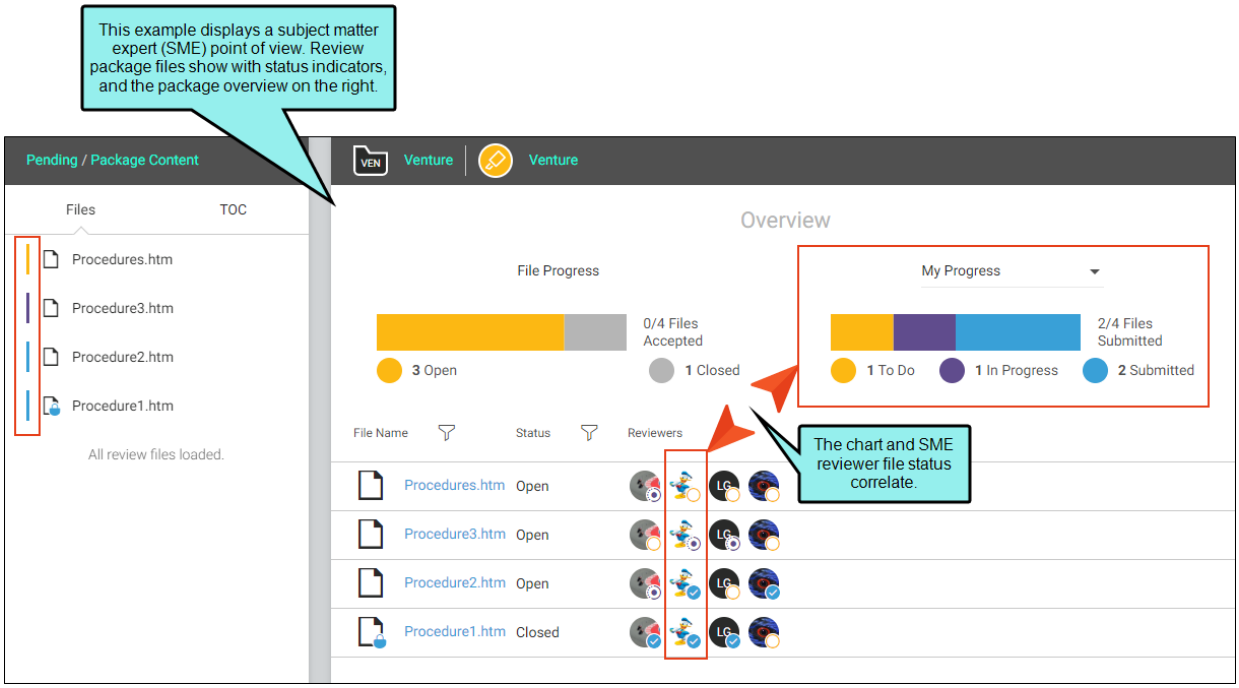
> **NOTE** Authors have access to data grids for monitoring and managing review packages in Central. From the top navigation select **Grids**. From there you can view the Packages grid all review packages in Central, or you can view a Files grid which shows all files sent for review regardless of the package.

Review Packages Overview

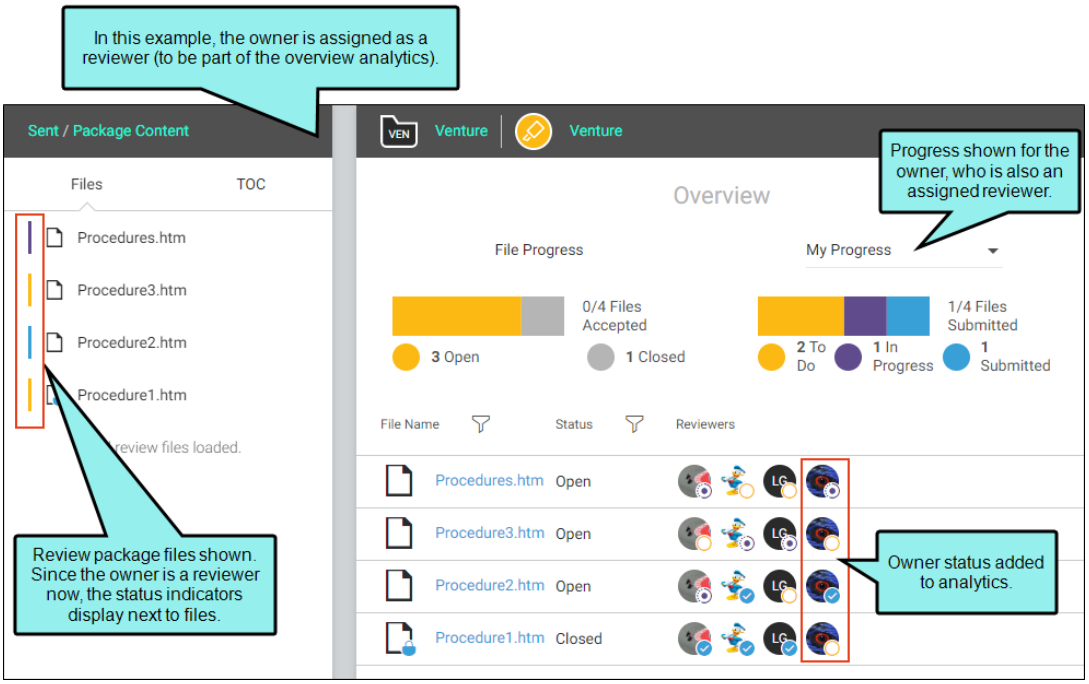
As soon as you select a review package, you can see details right away in its overview. The ability to monitor and update review packages, the file(s) within a package, and reviewers, provides value and momentum to the review process.

1. On the left side of the interface, click Reviews.
2. From the Review Explorer, select a package. The review package overview opens to the right. From the review package overview you can immediately see analytics regarding review package files. It shows the state of the files, your status, and the status of other reviewers.


In this example, the owner monitoring progress of the review package.
Packages
Pending
Sent
Submitted
Archived
Enterprise-Intro
Sent 5 days ago by Alfre Vong
Enterprise-Features
Sent 5 days ago by Alfre Vong
Venture
Sent 4 days ago by Alfre Vong
Venture-Snip
Sent 33 minutes ago by Alfre Vong
Monster-Shop
Sent 20 minutes ago by Alfre Vong
All review packages loaded.
Venture
Monster-Shop
Overall file progress that corresponds with the file grid below.
See the progress for all assigned reviewers.
Overview
File Progress
All Users Progress
0/4 Files Accepted
1/4 Files Submitted
3 Open
1 Closed
3 In Progress
1 Submitted
Feature review for product.
At-a-glance reviewer status.
File Name
Status
Reviewers
Features.htm Open
Feature3.htm Open
Feature2.htm Closed
Feature1.htm Open
All of the reviewers have submitted the file. In Flare, the owner has closed the file for further review. When this occurs, a lock displays for the file.


This example displays a subject matter expert (SME) point of view. Review package files show with status indicators, and the package overview on the right.
Pending / Package Content
Files
TOC
Procedures.htm
Procedure3.htm
Procedure2.htm
Procedure1.htm
All review files loaded.
Venture
Venture
Overview
File Progress
My Progress
0/4 Files Accepted
2/4 Files Submitted
3 Open
1 Closed
1 To Do
1 In Progress
2 Submitted
File Name
Status
Reviewers
The chart and SME reviewer file status correlate.
Procedures.htm Open
Procedure3.htm Open
Procedure2.htm Open
Procedure1.htm Closed

NOTE If as an owner, you want your changes to be included (along with other reviewers progress) in the analytics of the review, then you need to add yourself as a "reviewer" to the review package. When this happens, the owner must submit the changes just like any other reviewer. The review package overview changes slightly.

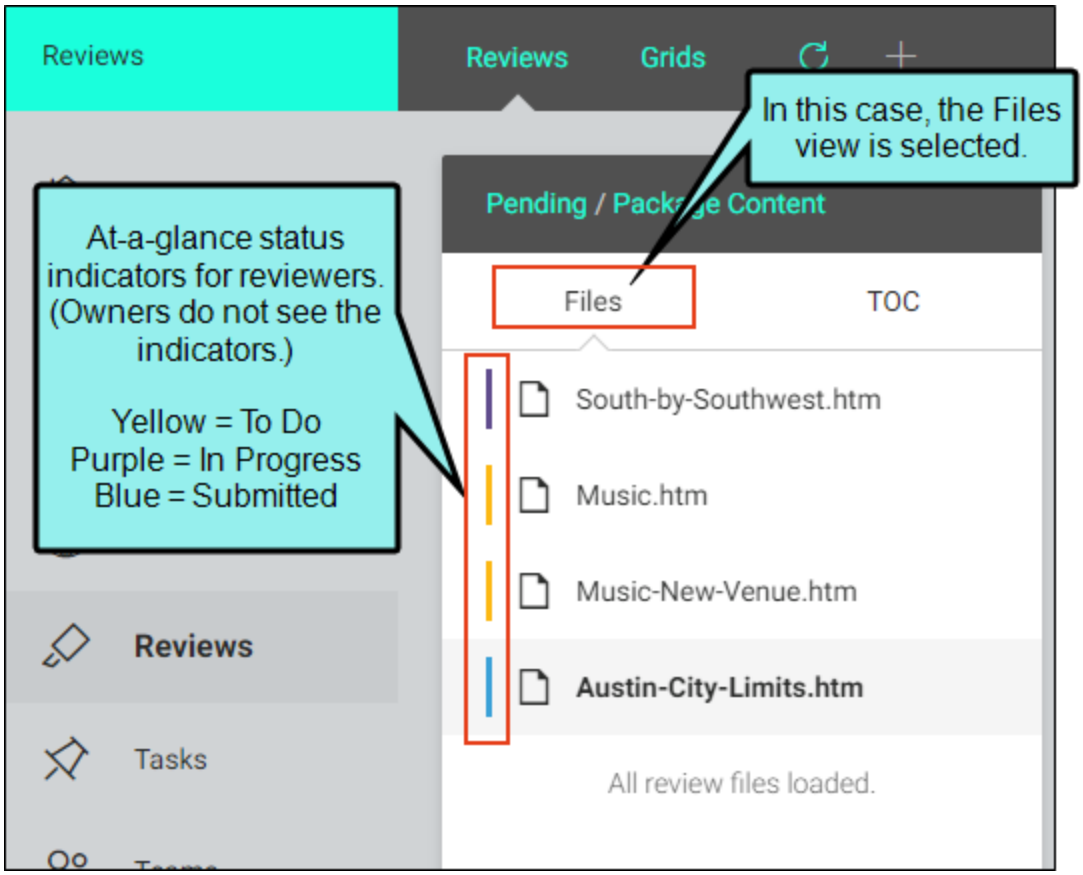


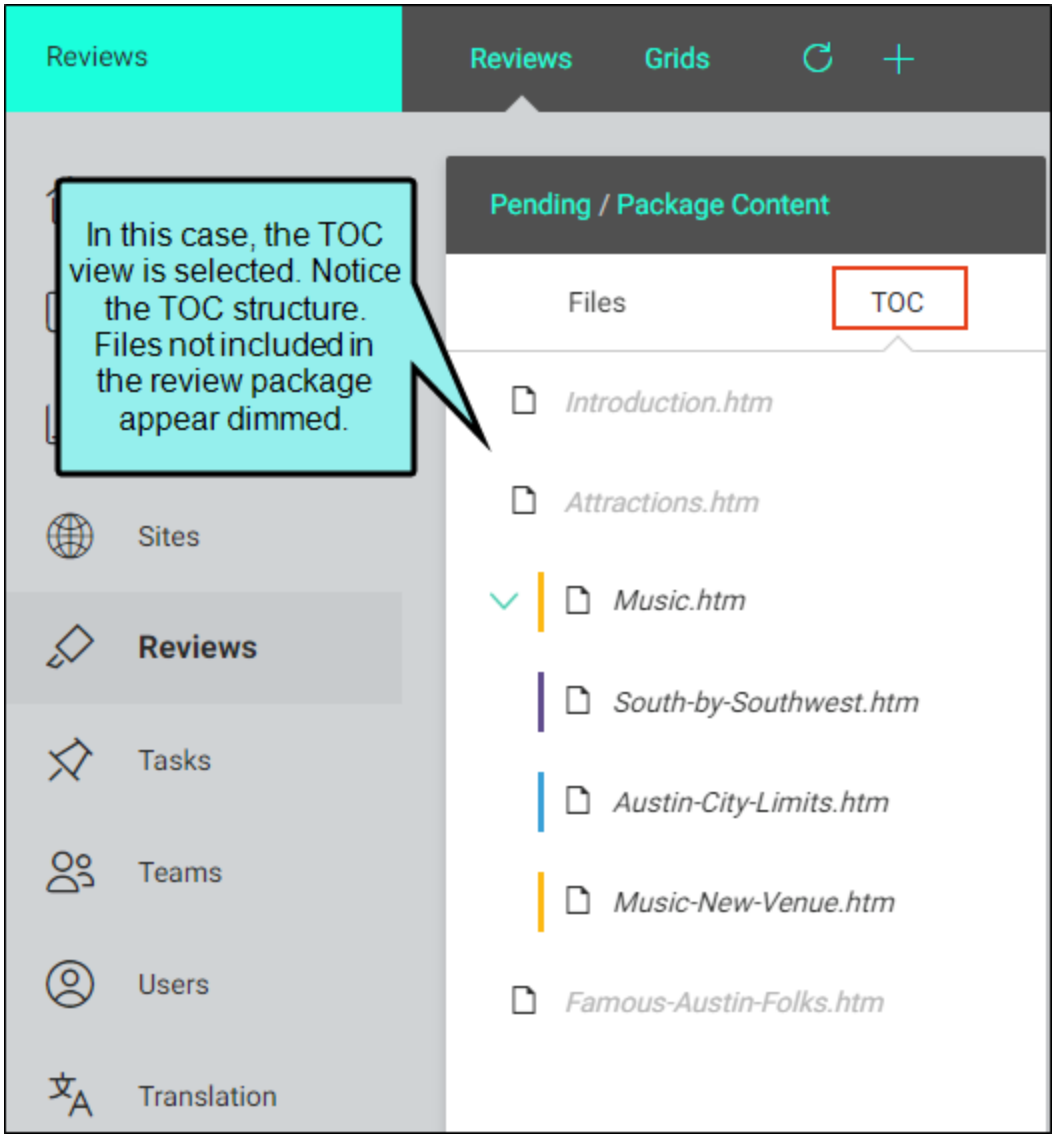
NOTE An author, who is also the owner of a review, can modify anything about a package or the files within it even if they do not have the Manage Review permission.

How to Toggle Files View and TOC View

Reviewers have the option to see individual files in a list or in a TOC view for review. The Files view means that you see a list of flat files. The TOC view, on the other hand, provides more context for content and a general flow of materials. The TOC view caters to the perspective of the end user since files are assembled as they would display in a TOC for output.

1. On the left side of the interface, click **Reviews**.
2. From the Review Explorer, select **Show files** from the desired package.
3. The pane changes to show the contents of the package. Select **Files** or **TOC** at the top for your preferred view.

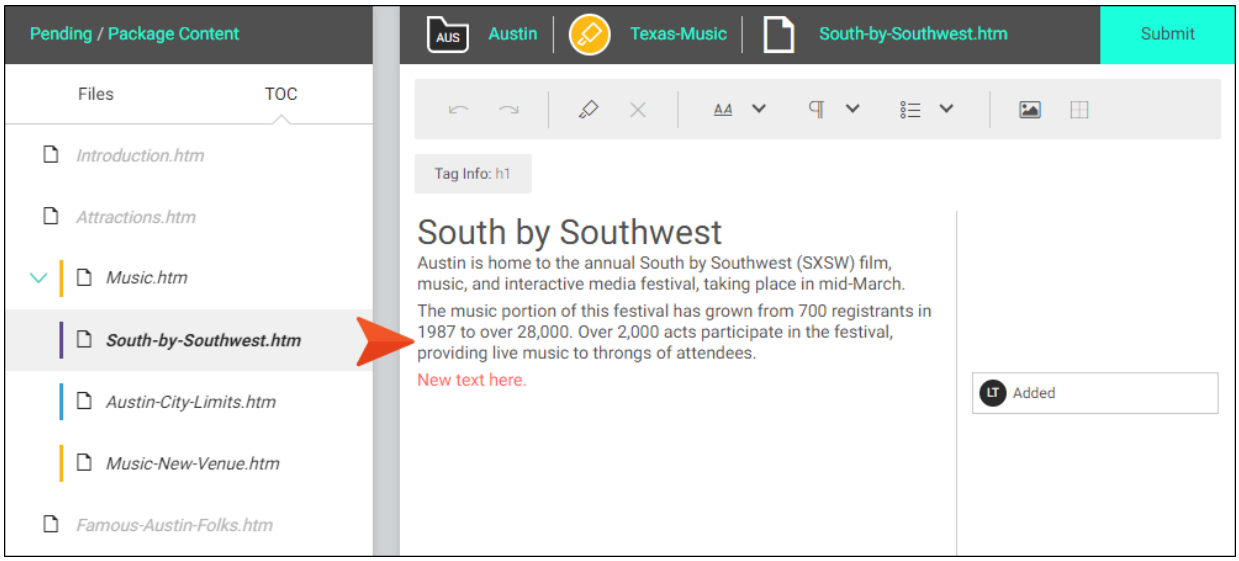




NOTE Topics and snippets can be included in a review package. Topics display in both views, but snippets only display in the Files view. Snippets do not display in the TOC view.

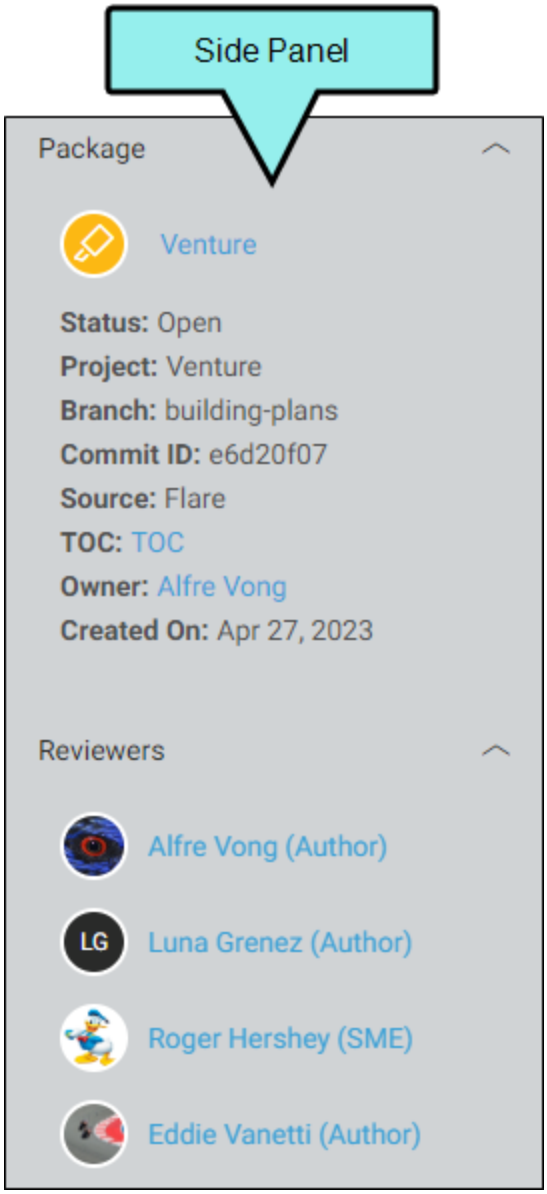
> **NOTE** Central's TOC view shows a label if it is entered manually into the TOC (i.e., in Flare, the label displays in black font). But, if you use a variable for a TOC node (i.e., you dragged the topic from the Content Explorer to the TOC and it displays in blue font), then Central shows the file name. The variables will display appropriately in the output. You can hover over the file name in the TOC view in Central, and it should show the variable.

4. Select a file. It displays in the right side for reviewing and editing.

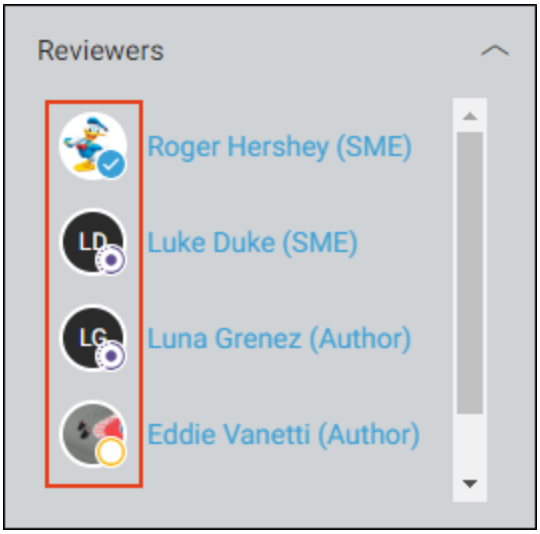


Review Package Side Panel

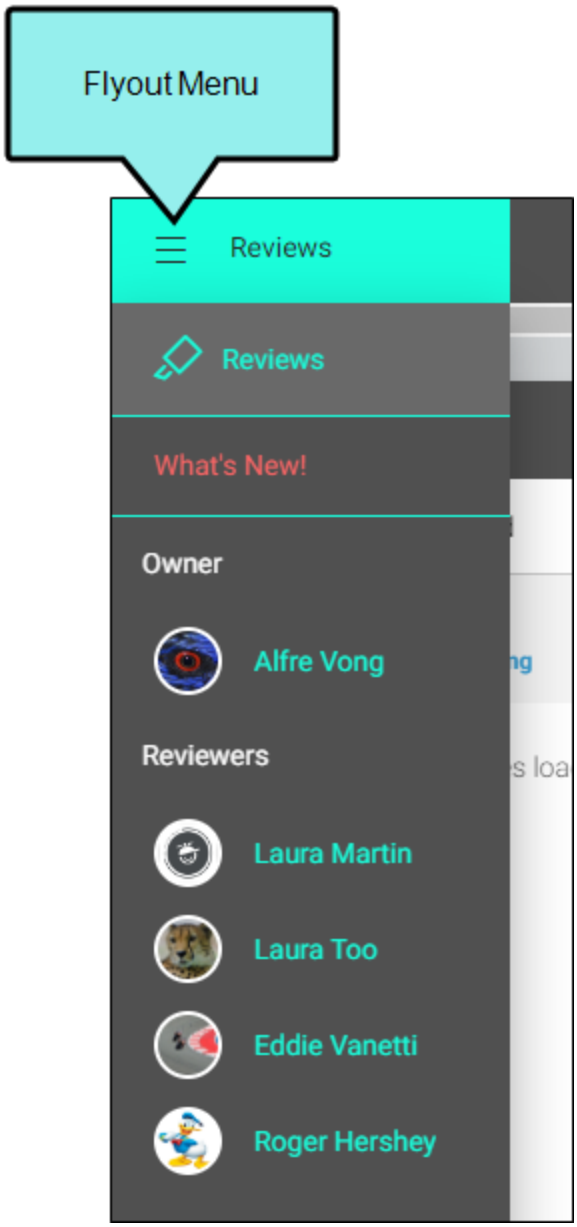
When a package is selected for review, use the right Side Panel of the interface to see at-a-glance review package information. It references package status, project, branch, commit ID, TOC (if applicable), owner, and the creation date of the package. Under that is a list of the reviewers associated with the file in the package.



NOTE When a reviewer has a file open for editing in the Review Editor, the Side Panel displays status indicators for all reviewers. This means that each reviewer can see the progress of other reviewers for the file.



If you are viewing Central on a smaller screen or at a low resolution, the information is accessible from the flyout menu on the left side, which can be opened by clicking in the upper-left corner.


Flyout Menu
Reviews
Reviews
What's New!
Owner
Alfre Vong
Reviewers
Laura Martin
Laura Too
Eddie Vanetti
Roger Hershey

Viewing Status

Virtually every component in Central reviews captures some sort of status. For example, status is reflected in the Review Explorer, the package overview, the right Side Panel, the grids, and the profiles. It ranges from what state the package is in, the overall status of all files in a package, to the individual reviewer status and where they are with a given file. Flare and Central work together to maintain the current state of files in a review package to their respective environments; giving momentum to the review process.

Statuses can be seen in different places of the user interface, but status meanings are consistent throughout reviews pages and dialogs.

Review Package Statuses

- Open The review package is open for review. This is the default setting.
- Closed All files in the package are set to a closed status (i.e., the package can no longer be reviewed). This can be set in the Review Package profile to put a review back into Flare's inbox as is (i.e., the review is not complete) and archive the review at the same time.

Review File Statuses

- Open The file is open for review. This is the default setting.
- Closed The file is closed for review.
- Accepted The file is accepted in Flare.

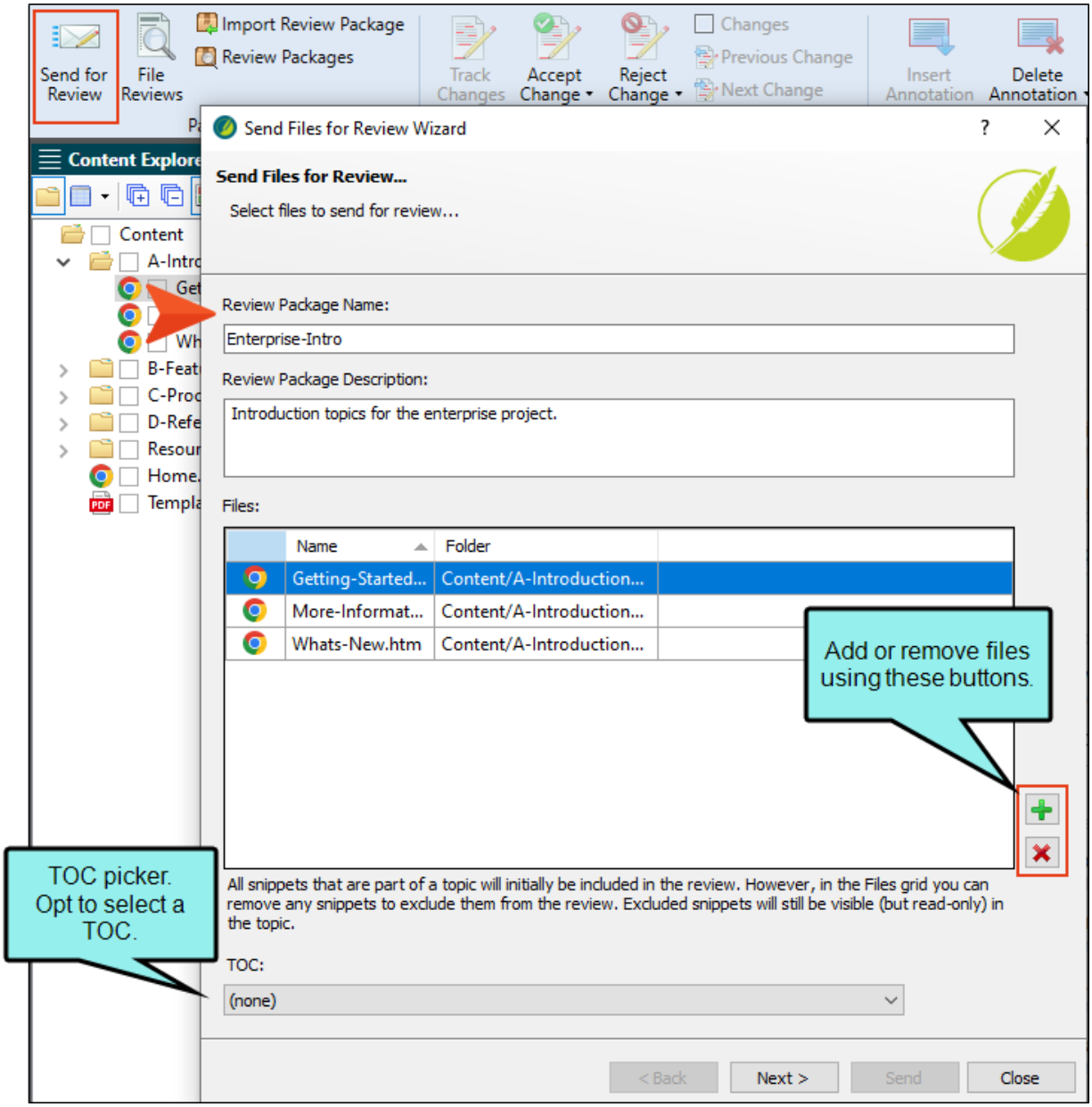
Reviewer Statuses

- To Do The reviewer has not started editing the file yet. The status color is yellow.
- In Progress The reviewer has started editing the file, but has not submitted it. The status color is purple.
- Submitted The reviewer has edited the file and submitted it. The reviewer can still edit the file until the owner closes it. The status color is blue.

Review Workflow

The Central review packages workflow involves a combination of activities between the owner originating the process and all reviewers making changes and comments in the files.

1. Owner In Central, open the Users page and invite the reviewer(s) to Central.
2. Reviewer Click the link in the email to accept the invitation and set a password.
3. Owner In Flare, switch to the appropriate branch (if necessary), select Review > Send For Review. The first screen of the Send Files for Review Wizard opens.
    a. In the Review Package Name field, enter a name.
    b. (Optional) Enter a description for the review package.
    c. (Optional) Add or remove files for the package.
    d. (Optional) The TOC drop-down defaults to (none). To associate files within a review package to a TOC in the project, select a TOC.
    e. Click Next.

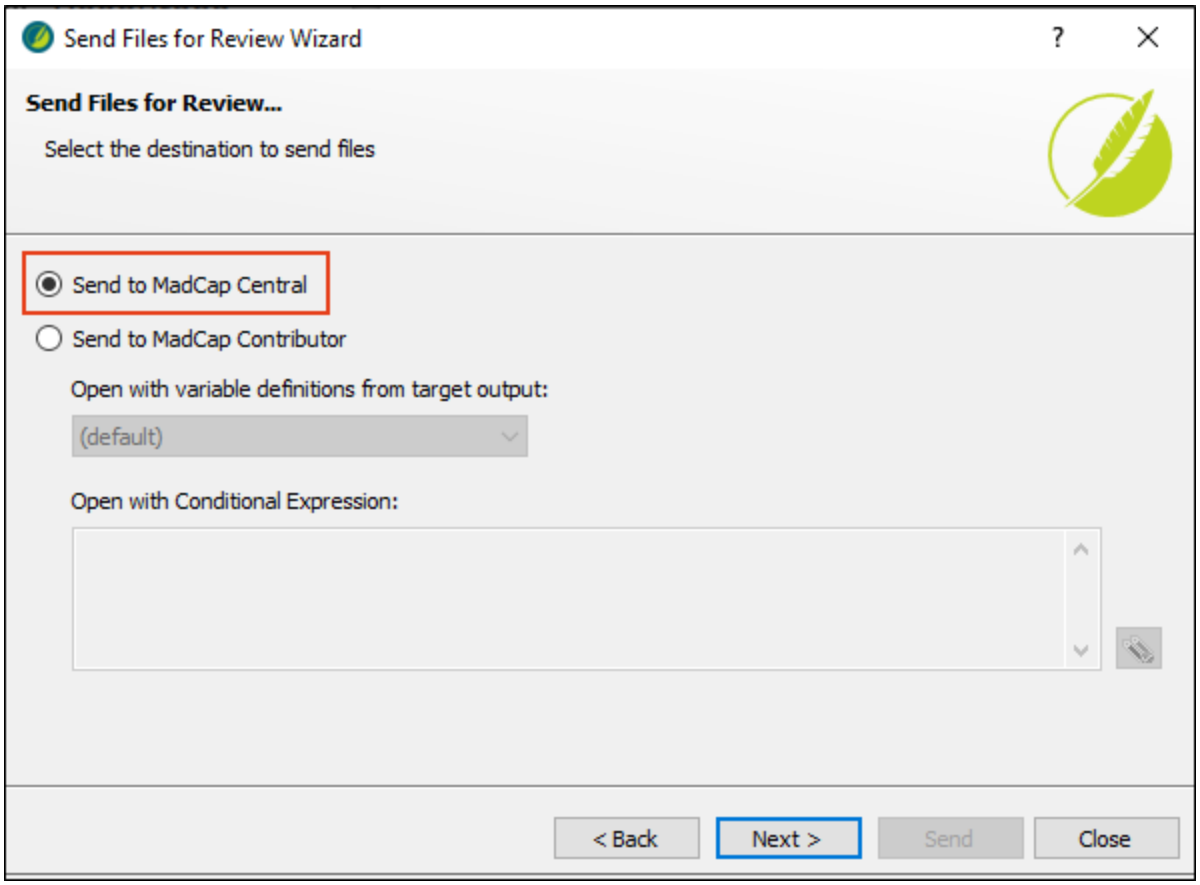

Send for Review
File Reviews
Import Review Package
Review Packages
Track Changes
Accept Change
Reject Change
Changes
Previous Change
Next Change
Insert Annotation
Delete Annotation
Content Explore
Content
A-Intro
B-Feat
C-Proc
D-Refe
Resour
Home.
Templa
Send Files for Review Wizard
Send Files for Review...
Select files to send for review...
Review Package Name:
Enterprise-Intro
Review Package Description:
Introduction topics for the enterprise project.
Files:
Name
Folder
Getting-Started...
Content/A-Introduction...
More-Informat...
Content/A-Introduction...
Whats-New.htm
Content/A-Introduction...
Add or remove files using these buttons.
All snippets that are part of a topic will initially be included in the review. However, in the Files grid you can remove any snippets to exclude them from the review. Excluded snippets will still be visible (but read-only) in the topic.
TOC picker. Opt to select a TOC.
TOC:
(none)
< Back
Next >
Send
Close

TIP All snippets that are part of a topic will initially be included in the review. However, when sending files for review from Flare, you can remove any snippets from the File grid to exclude them from the review. You might choose to exclude some (or all) snippets from the review if you think it will be too confusing and too many files for your reviewers. If you exclude snippets, they will still be visible in the topic in Central, but the snippets will be read-only. Therefore, reviewers won't be able to make edits in the snippets, but in the topic they can insert annotations (comments) around the snippets in order to provide feedback.

NOTE If you are using Git branching, make sure the appropriate branch is active. When you send files for review, they will be associated with that branch.

In Central, reviewers can determine which branch a file is from. Reviewers do not need to do anything special to make edits or comments in the files.

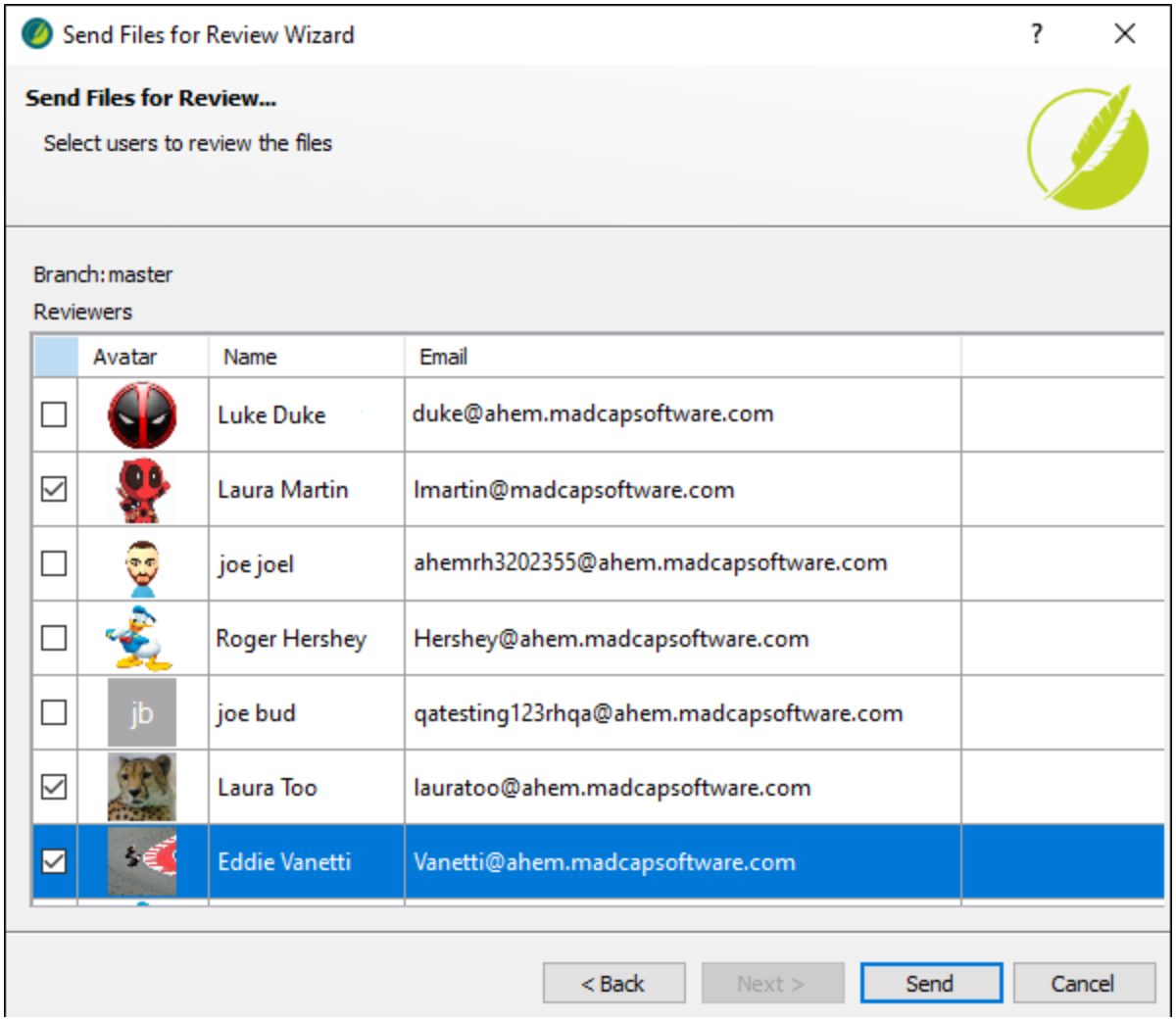
4. **Owner** Select **Send to MadCap Central** (if not already selected), and click **Next**.



> **NOTE** If necessary, Flare also prompts you to commit and synchronize any changes in your project with Central before you can proceed to the next page of the wizard.

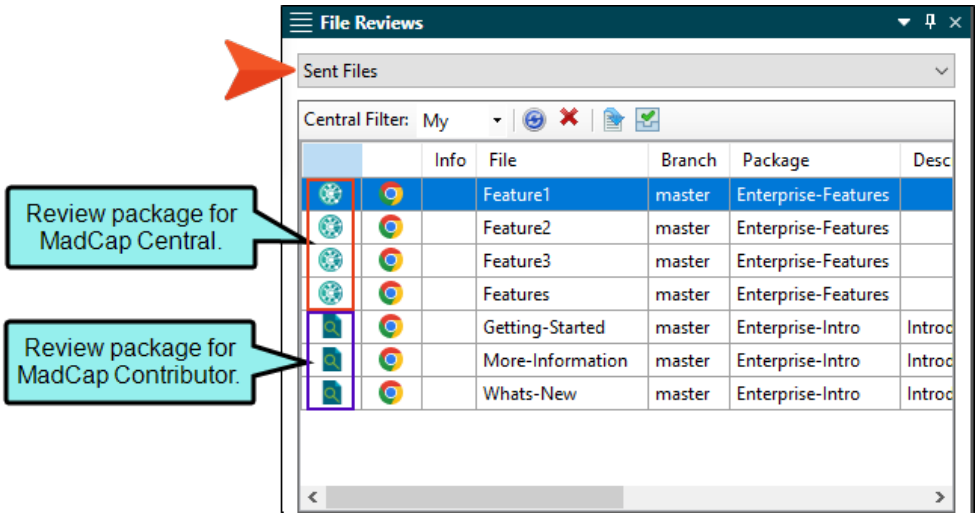
5. Owner (Optional) Select reviewers for the review package, and click Send. The wizard closes and a prompt alerts you that the files have been sent to Central, click OK.

   The owner can choose reviewers of different seat types, such as authors and SMEs. Anyone selected during this process automatically becomes associated with the Flare project on Central, and becomes an assigned reviewer for all files chosen in the wizard.



After the review package is uploaded, reviewers receive an email notification with a link to open the Reviews page in Central. (SMEs automatically receive email notifications, but Flare authors must have the review notifications and email option enabled in Central to receive emails.)

In File Reviews, the review package files display in the Sent Files area. This means the files are in an "Open" state, pending review.



> NOTE If you attempt to edit a file that has been sent out for review and is still in an "Open" state, a warning displays. You can edit the file, but you may have conflicts with the version of the file being reviewed. As an alternative to editing the file in Flare, you can edit it in Central along with your reviewers until it is finished.

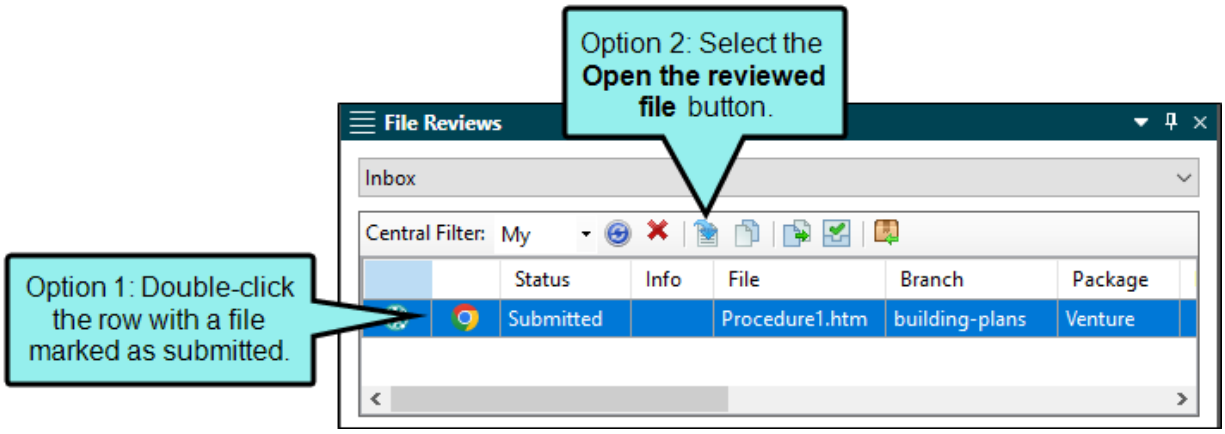
6. **Reviewer** In Central, open the Reviews page. Click each file associated with the pending review package and edit or annotate it using the Review Editor, and click **Submit**.

   The reviewer is done submitting once all the files in the queue are complete. This moves the review package to the Submitted pane. Files can be viewed from there and edited still, but once the owner closes the file, it can no longer be edited in Central.

   After all of the reviewers submit the file(s), the owner receives an email notification (if notifications are enabled in Central) and the file(s) displays in the File Reviews Inbox in Flare.

NOTE If the owner of a review package makes edits to the files in Central they will not typically see a Submit button. If as an owner, you want your changes to be included (along with other reviewers progress) in the analytics of the review, then you need to add yourself as a "reviewer" to the review package. When this happens, the owner must submit the changes just like any other reviewer.

7. Owner After reviewed files are individually submitted from Central, open Flare. Select Review > File Reviews. You can look at files as they come in; you do not have to wait for all the files in a review package in order to preview them.
8. Owner From the File Reviews Inbox area double-click a submitted file. (Or, select a file, and from the local toolbar, click Open the reviewed file .)

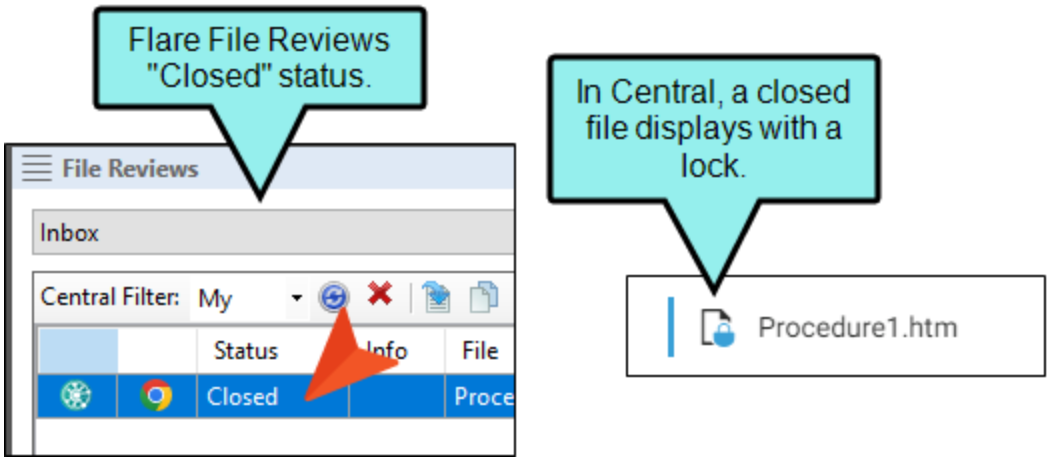


NOTE The Inbox displays files from all relevant branches (if you are using Git branching). You should switch to the matching branch before managing the tracked changes and accepting the file back into Flare.

NOTE Any authors working in that Flare project can close the review file. This is necessary in case the owner (i.e., original author) is not available to close it.

9. Owner When you click to open and preview a file, a dialog displays asking if you want to move the files to a closed state. Select Yes or No.

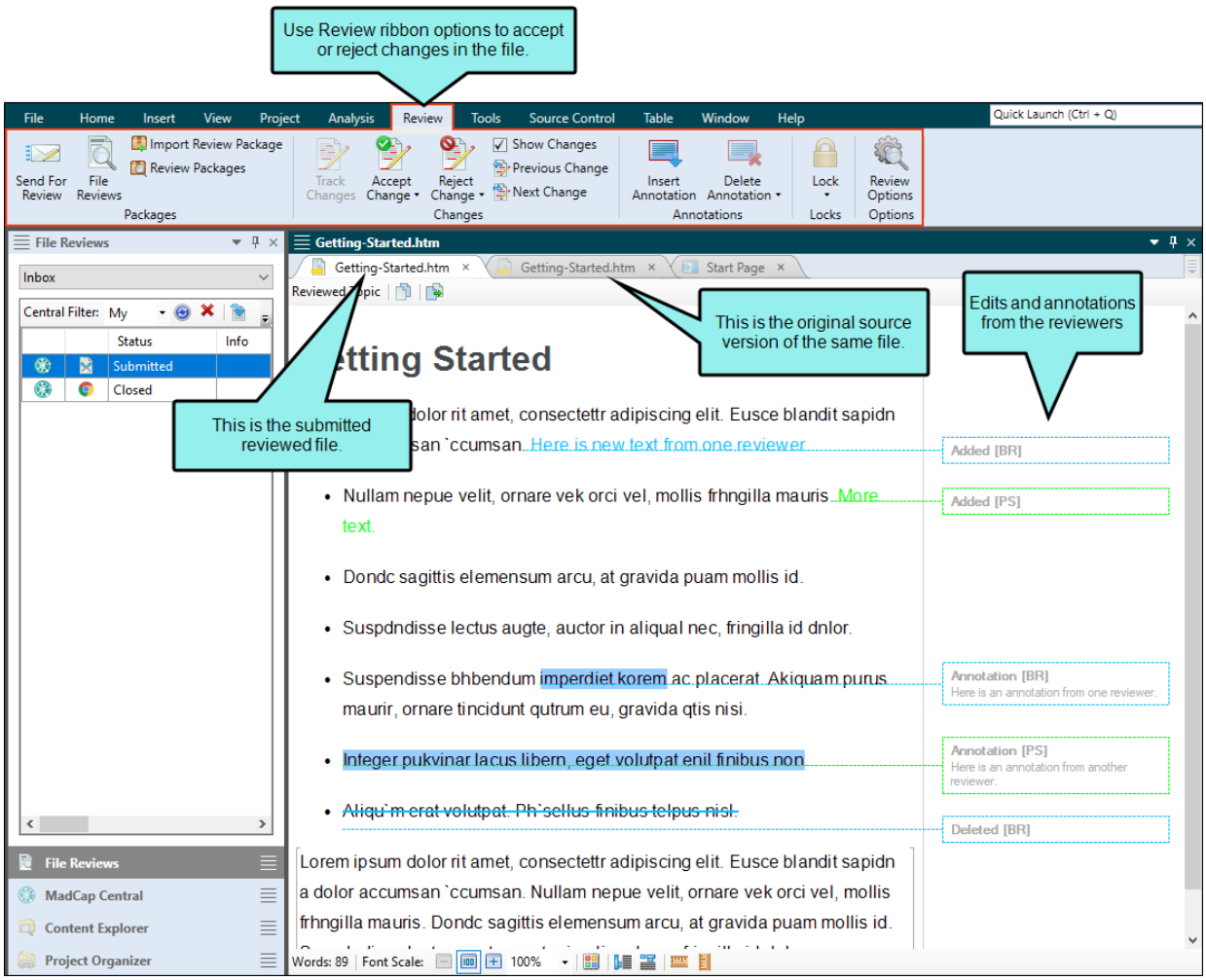
   - Yes The file switches to a closed state in the Inbox and opens in the editor. No more editing can occur in Central while the owner reviews it. In Central, when a file is closed, a lock symbol appears next to the file.



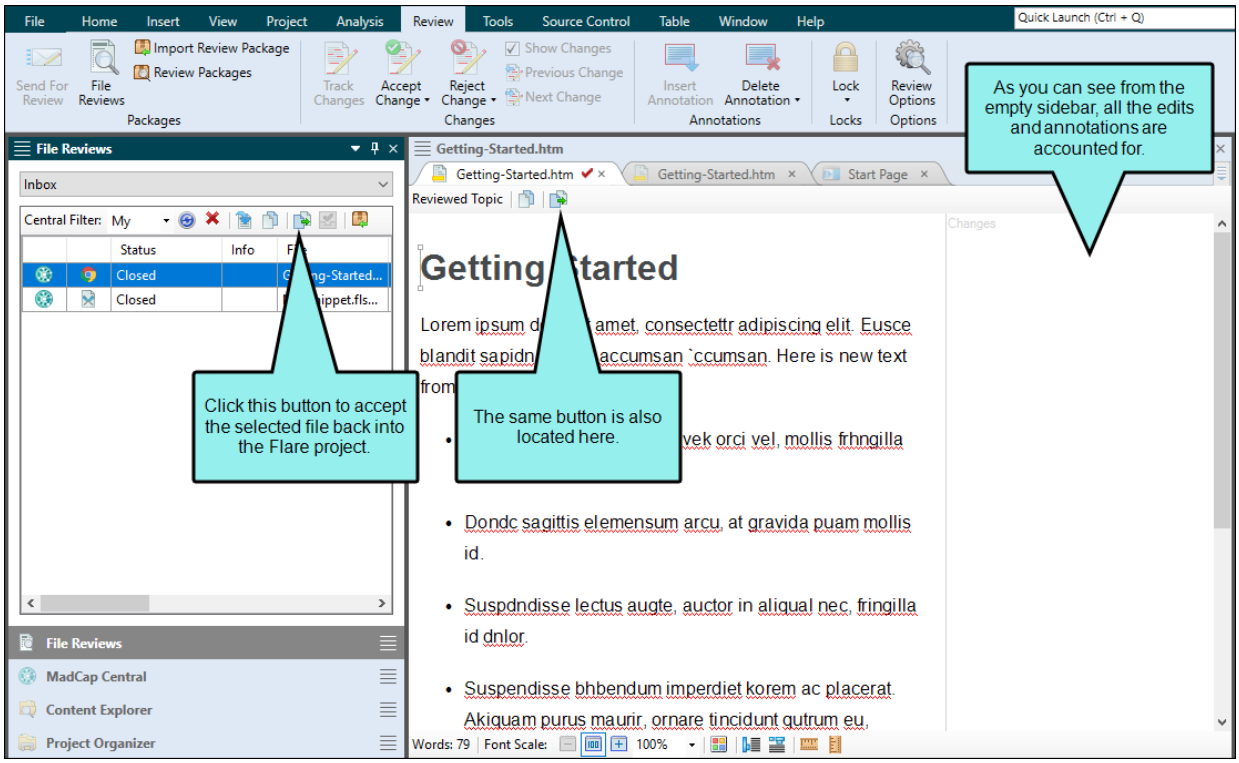
   - No The owner can preview the file in Flare—as a read-only file. The reviewer(s) can still make changes to the file in Central until the owner closes the file in Flare. The file stays in a submitted state in the Inbox, and it opens in the editor.

> NOTE The advantage of this is that you can see raw edits before the reviewer is done—without having to close the file and resend it from Flare. When you are ready to close out the file from further editing, double-click the submitted file and select Yes from the prompt.

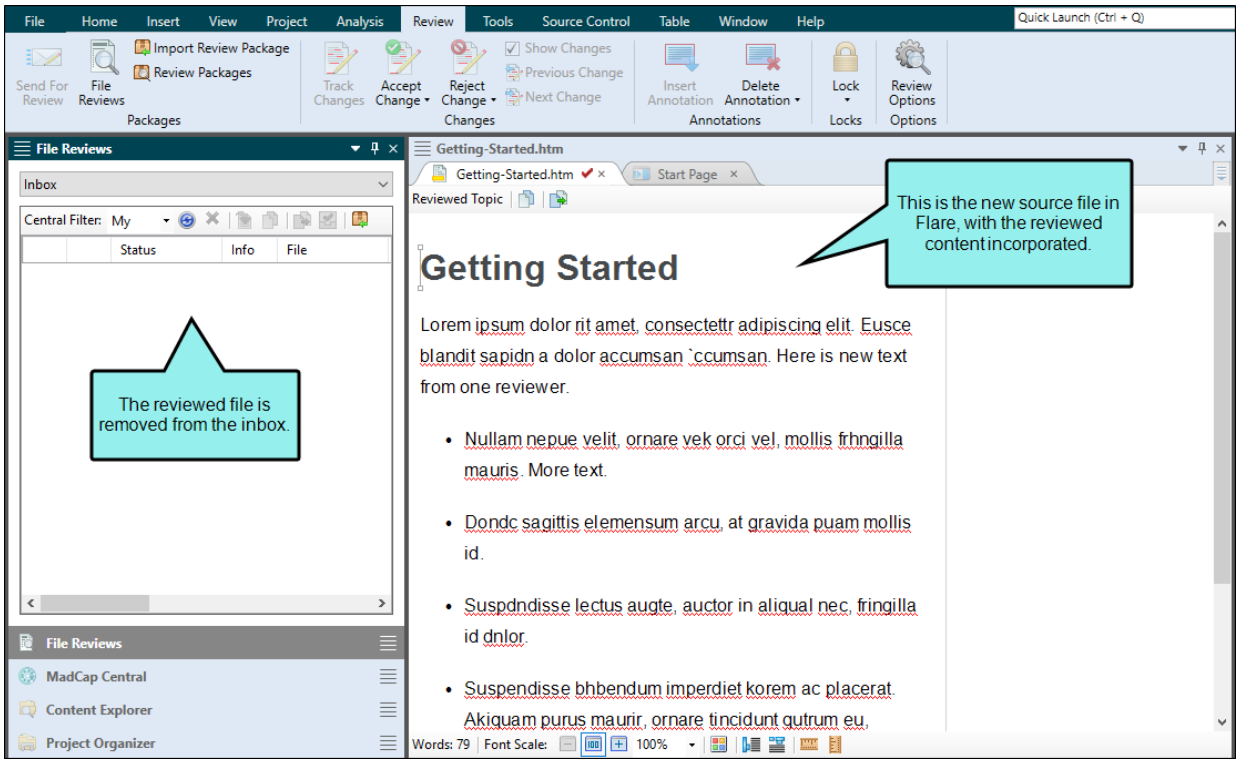
10. **Owner** In Flare, accept and/or reject the reviewer's changes in the editor.



11. **Owner** In the toolbar in Flare, click the **Accept** button.



The file is then removed from the Inbox.



What's Noteworthy?

NOTE You can also initiate the review process in Central directly. This might be more convenient if you are already working in Central.

NOTE For the review process, keep in mind that review packages are sent up, and files come down. In other words, owners are sending review packages up to Central, and then reviewers are submitting individual files back down to Flare. The advantage to this is that you can look at files as they come in; you do not have to wait for all the files in a review package in order to review them. Once all package files are accepted into Flare, the review package is closed.

Main Activities for Reviews

Some activities are particularly common and important when it comes to this feature.

- "Opening Reviews" on the next page
- "Editing Review Files" on page 254
- "Submitting Review Files" on page 258
- "Setting Notifications for Reviews" on page 260

Opening Reviews

Opening review packages containing files sent for reviews is quite simple, but keep in mind that there are multiple panes that display on the Reviews main page.

Permission Required?

Reviews do not require permission to partake in them. However, managing and creating review packages in Central does require the Manage Reviews permission.

How to Open a Review Package

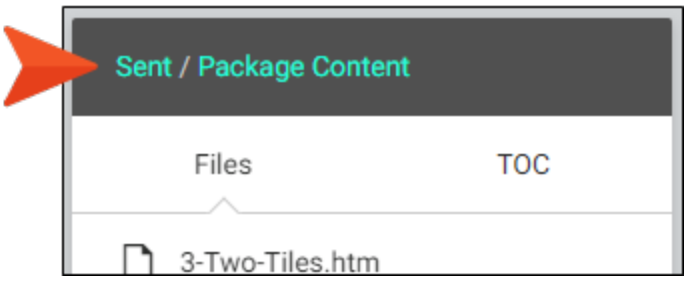
1. On the left side of the interface, click **Reviews**.
2. From the top navigation, click **Reviews**.
3. From the appropriate **Packages** pane (e.g., Pending, Sent), select the row of the desired review package.

> TIP Use the filter to search for a specific review package. This might come in handy if you have many review packages listed in the Review Explorer.

How to Open a File in the Review Editor

1. On the left side of the interface, click **Reviews**.
2. From the top navigation, click **Reviews**.
3. From the appropriate **Packages** pane (e.g., Pending, Sent), select a package:
   - If an owner has sent a package to you for review, open it from the **Pending** pane.
   - If you are the owner, you can open a package from the **Sent** pane. You might do this if you want to continue editing the file after you have sent it to others for review. Editing it in Central might be preferable to editing it in Flare at this point; otherwise, you would have to resolve your changes with those from reviewers when you bring the file back into Flare.
   - If you are a reviewer who has already submitted a file, you can open it from the **Submitted** pane. You might do this if you remember something you wanted to edit after submitting it. You can access the file in this pane until the owner closes out the review process.

   A package overview opens to the right.
4. From the selected review package in the Review Explorer, select **Show files**. At the top of the pane, notice the clickable breadcrumb navigation.



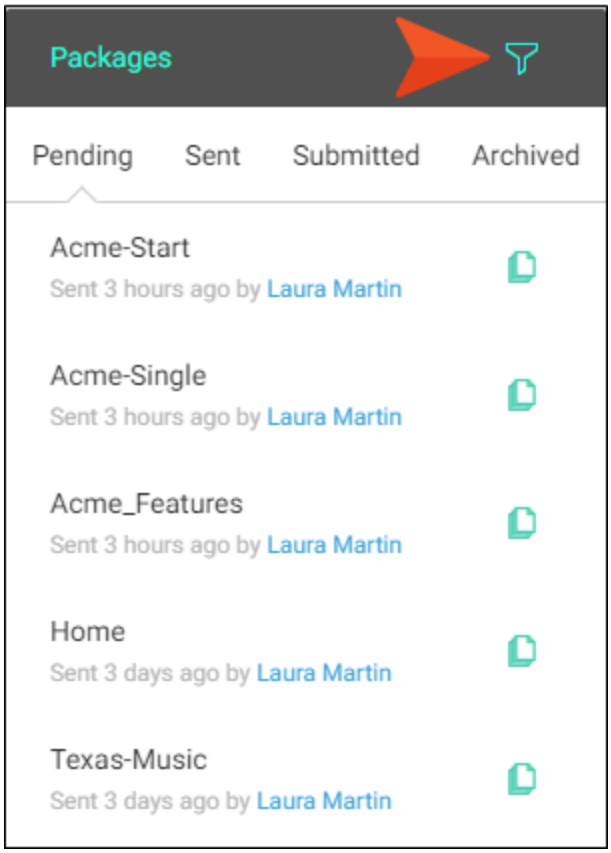
5. Select **Files** or **TOC** at the top for your preferred view. If an owner chooses not to associate files with a TOC, then a TOC will not be applicable to your review package.
6. Select a file. It displays in the Review Editor to the right.

> NOTE Keep the following in mind when loading files:
>
> - The file size limit is 500 KB. This limitation helps prevents issues.
> - If changes approach the limit, a message warns you.

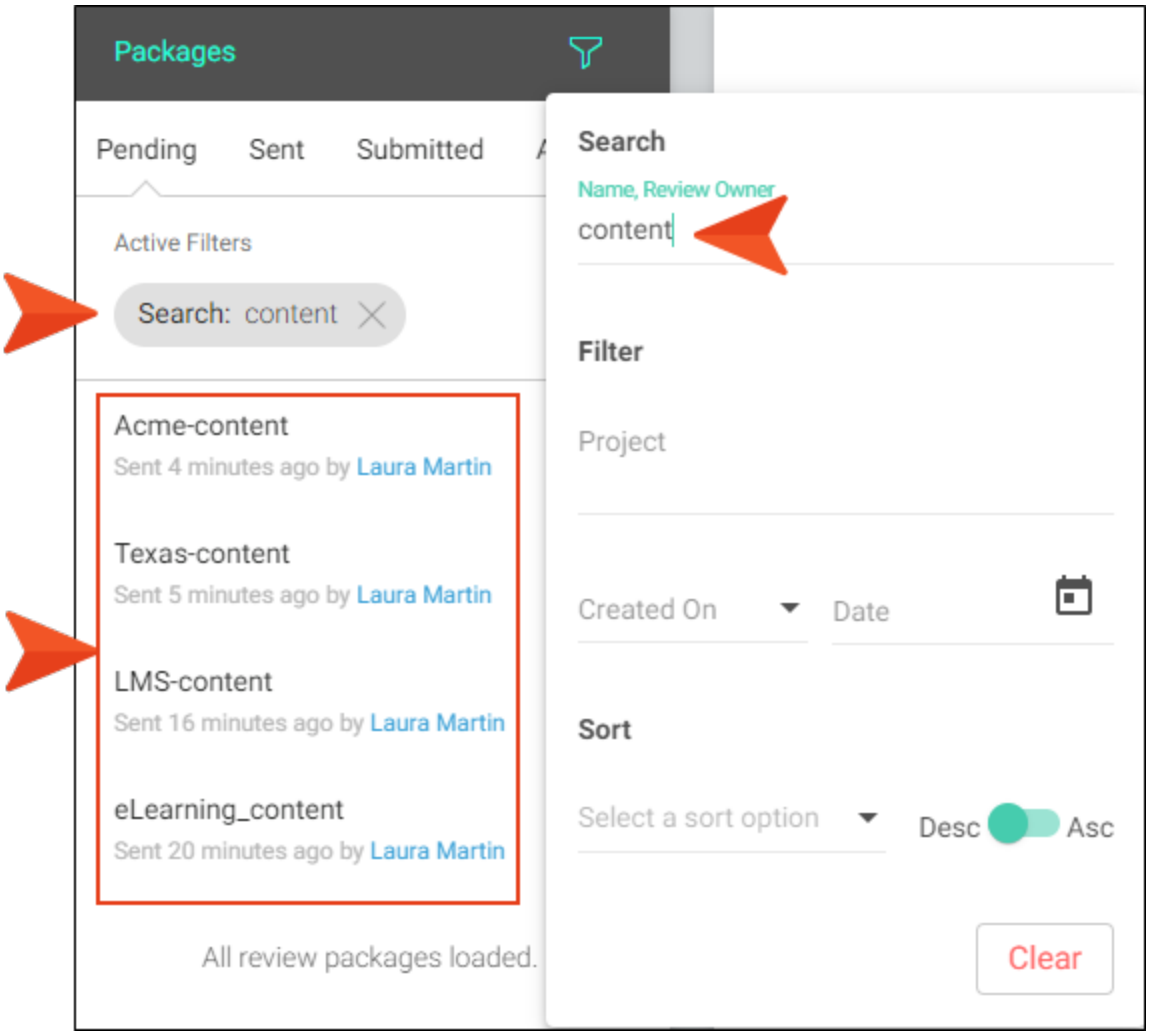
- If edits exceed the size limit, reviewers are given the opportunity to undo edits or revert the changes and return to the previous saved version of the file.
- Edits over 1 MB are ignored.
- If an author sends a file for review that already exceeds the limit, a reviewer cannot load that file in Central. To resolve the issue, the author can close the review file in Flare (in the File Reviews window pane), adjust the content to lower the file size, and then resend the file for review.

Filtering Review Packages

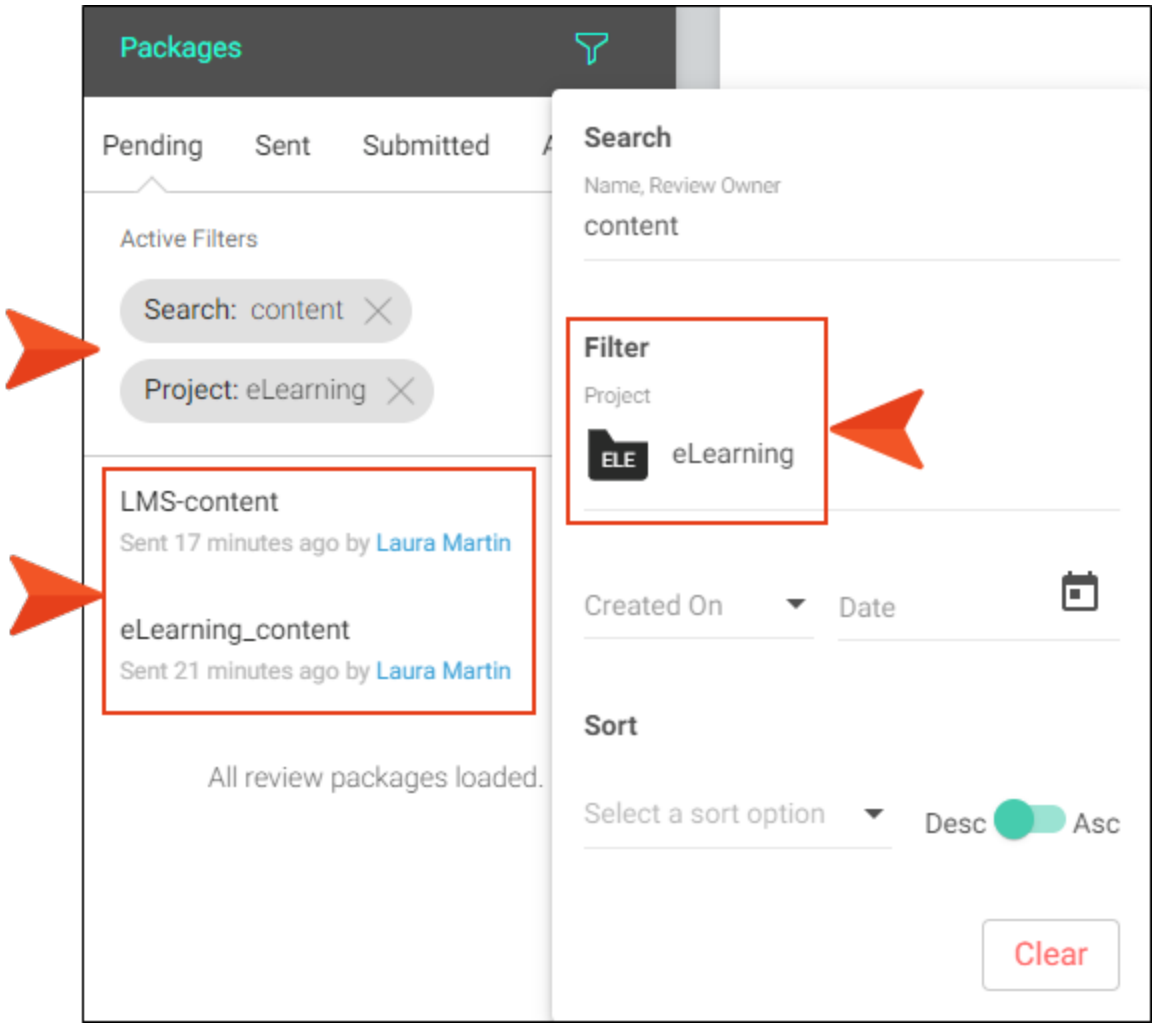
If you have a really long list of review packages shown, you can use a filter option to truncate the list to easily find what you need.



Within the dialog, you can begin typing in the Search field. This quickly adjusts the pane to the left, displaying only the packages that contain a name, or review owner matching the search text. In addition, the active filter is added at the top; you can click the X to clear that filter.

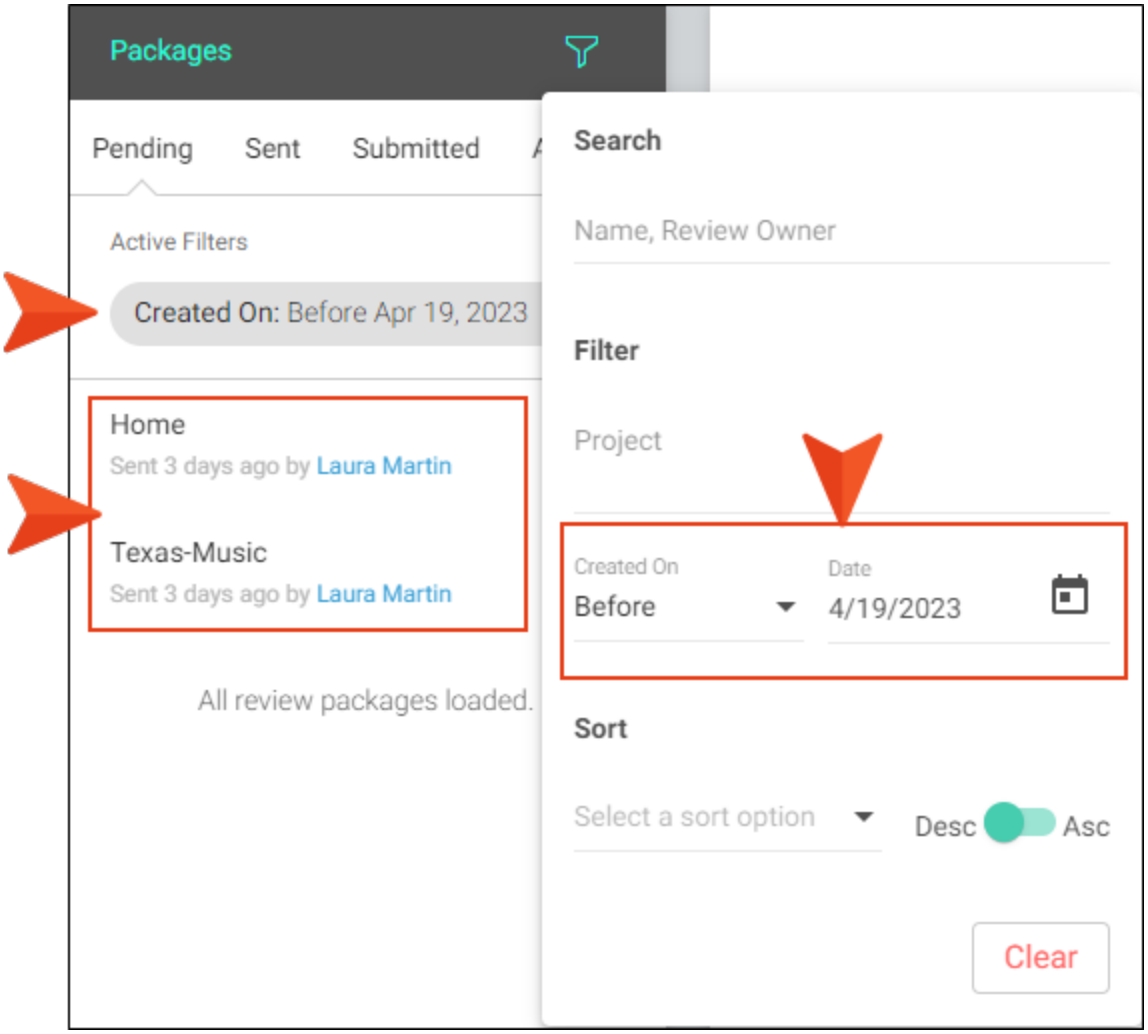


The first field in the **Filter** section lets you quickly select particular Flare projects to limit the results even more. The active filter at the top of the window pane adjusts automatically.

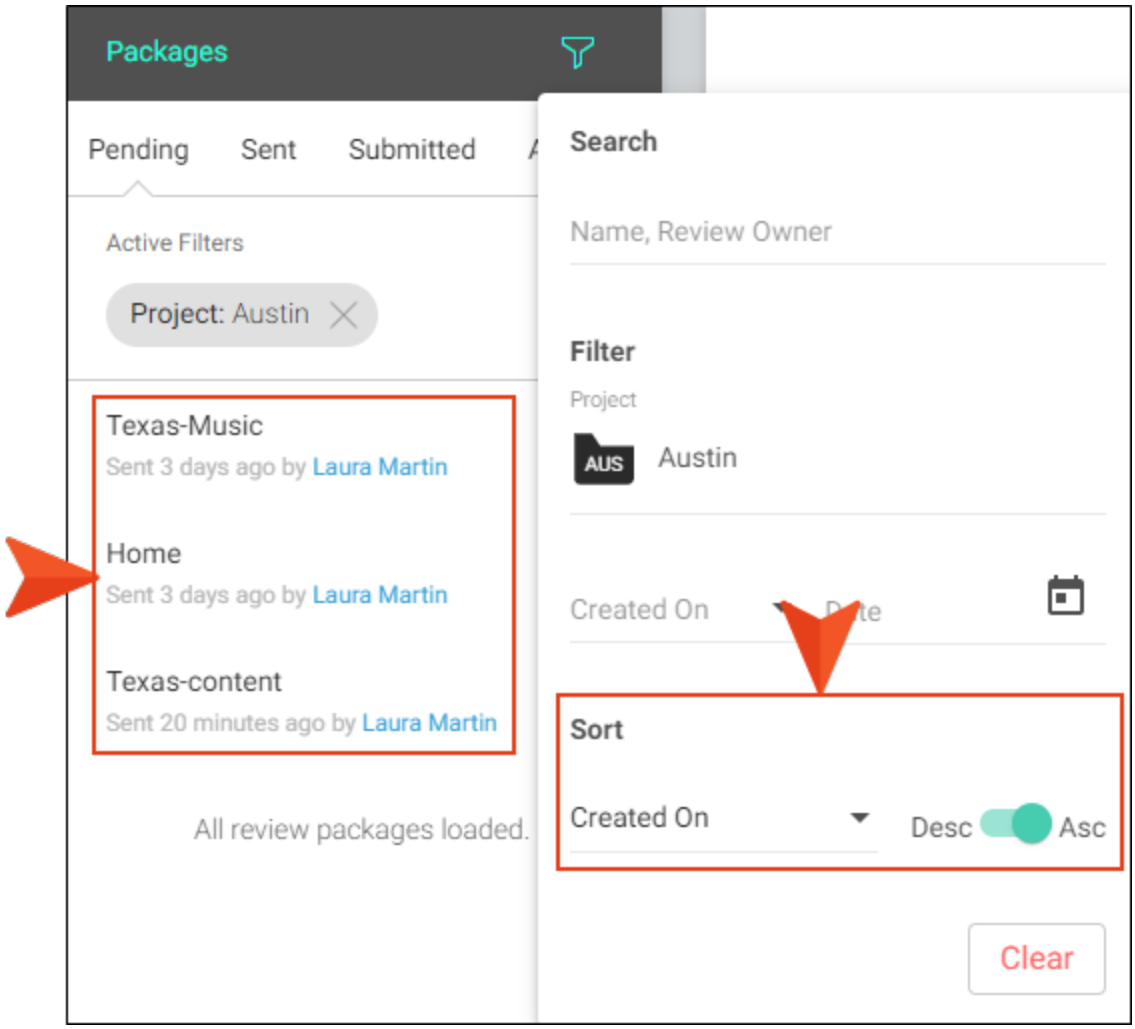


You can also select a date related to the sent date for review packages.

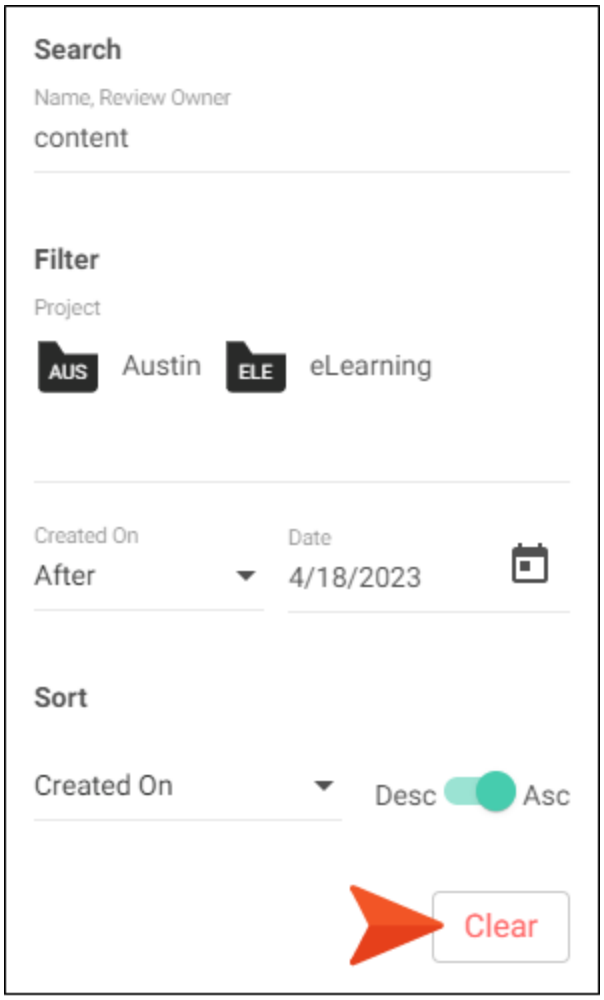
- **Created On** Shows packages created on the specific date selected.
- **Before** Shows packages sent before the specific date selected.
- **After** Shows packages sent after the specific date selected.



You can use the **Sort** section to organize the list of packages by None, Name, or Created On. Also, a toggle option also lets you sort the list in descending or ascending order.



A button at the bottom lets you clear all of the filter settings.



When you are finished setting options, just click off the dialog.

Editing Review Files

Editing the files sent to a reviewer is the crux of the review process between Flare and Central. From the Review Editor, all the reviewer needs to do is click to show files from the relevant review package, select an active file, and then edit and make annotations in the Review Editor. In addition, multiple people can edit a file at the same time, so you can see their changes in real time.

Permission Required?

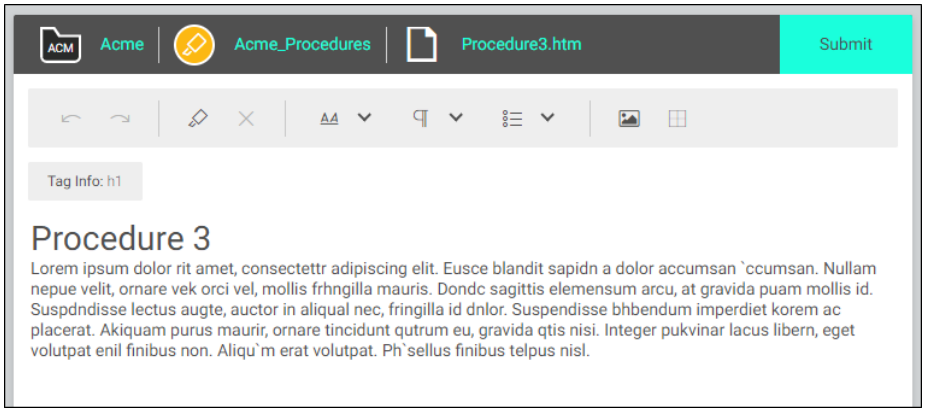
No special permission is required for this activity. All users who are associated with the review process for the specified file are allowed.

Review Editor

All reviewers associated with a review can view and interact with the Review Editor, but depending on the user seat type and permissions, the editor enables or restricts functionality accordingly.

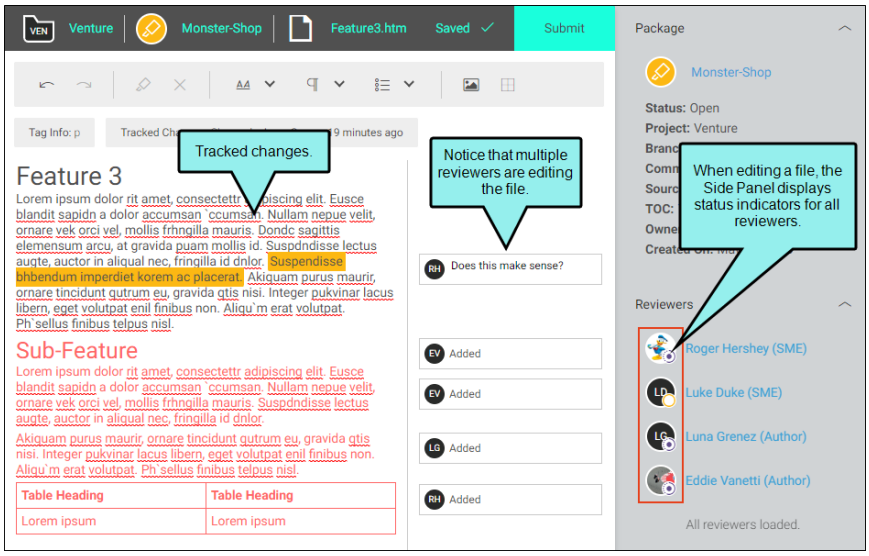
For example, an author who is also the owner of the review package can edit files, and those files save automatically—without having to submit an edited file (at the top of the editor). But for an author or a subject matter expert (SME) who has been sent a file to review, then a Submit button displays. In addition, the other elements in the banner (e.g., project icon, review package icon, review package link, content file link) are interactive for an author but not for a SME.

> **NOTE** If as an owner, you want your changes to be included (along with other reviewers progress) in the analytics of the review, then you need to add yourself as a "reviewer" to the review package. When this happens, the owner must submit the changes just like any other reviewer.



How to Edit a File

1. On the left side of the interface, click **Reviews**.
2. From the top navigation, click **Reviews**.
3. From the appropriate **Packages** pane (e.g., Pending, Sent), find the relevant review package, and select **Show files** .
4. Select **Files** or **TOC** at the top for your preferred view. If an owner chooses not to associate files with a TOC when initially sending a review, a TOC will not be applicable to your review package.
5. Select a file. It displays in the Review Editor to the right.
6. Start editing the file. For example, you can add or delete text, add bulleted lists, add images, add tables, bold text, create annotations, etc.



As you edit notice that all changes are tracked and auto-saved as you work.

7. When you are done editing the file, select **Submit** at the top of the editor.

> **NOTE** When you click Submit, you are not submitting all the files associated to a review package, but you are submitting only that one file. Once you submit a file, it moves to the Submitted pane in the Review Explorer and the owner is notified to accept it back into Flare. Until the owner closes the file, you can reopen the submitted file and make more changes to it if necessary.

Editor Features

For details about all of the features in the Review Editor, please see the online Help.

What's Noteworthy?

NOTE In addition to typing, you can copy and paste content. When you paste content into the editor, any styling is removed. The pasted content will take on the style of the content where it was pasted.

NOTE Aside from a few basic tags available in the toolbar (B, I, U, P, H1-H6, and lists), you cannot apply styles to a file that is being reviewed. You can only edit the content. Also, the files will not look exactly like your source files because the formatting from your stylesheet is not rendered in this editor.

NOTE Keep the following in mind when loading files:

- The file size limit is 500 KB. This limitation helps prevents issues.
- If changes approach the limit, a message warns you.
- If edits exceed the size limit, reviewers are given the opportunity to undo edits or revert the changes and return to the previous saved version of the file.
- Edits over 1 MB are ignored.
- If an author sends a file for review that already exceeds the limit, a reviewer cannot load that file in Central. To resolve the issue, the author can close the review file in Flare (in the File Reviews window pane), adjust the content to lower the file size, and then resend the file for review.

Submitting Review Files

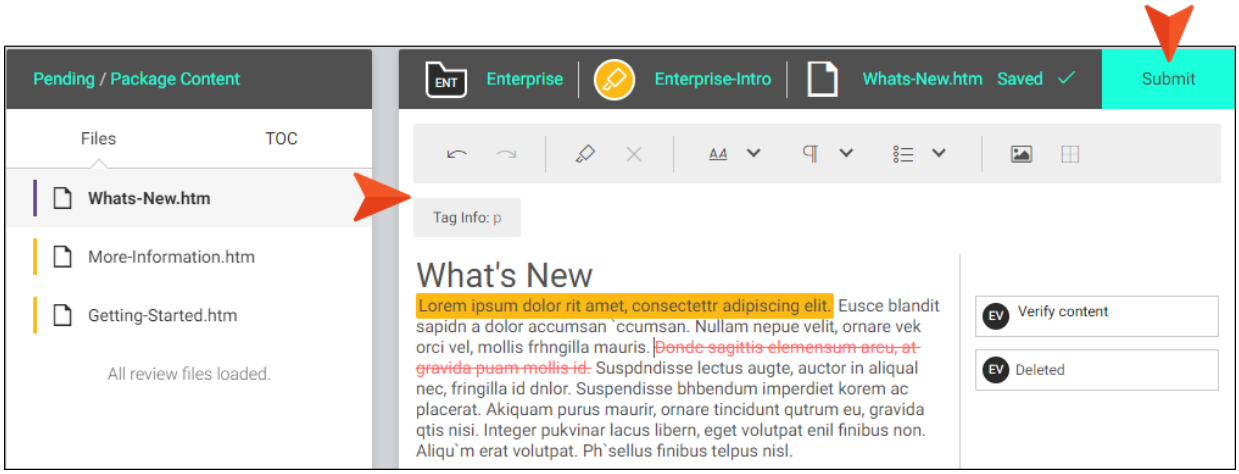
Reviewers can return to a file in the Pending pane repeatedly over time to make changes to it. When reviewers finish working on a file and do not need to make any further edits, they submit the file.

Permission Required?

No special permission is required for this activity. All users who are associated with the review process for the specified file are allowed.

How to Submit Review Files

1. On the left side of the interface, click **Reviews**.
2. From the top navigation, click **Reviews**.
3. From the appropriate **Packages** pane (e.g., Pending, Sent), find the relevant review package, and select **Show files**.
4. Select **Files** or **TOC** at the top for your preferred view. If an owner chooses not to associate files with a TOC when initially sending a review, a TOC will not be applicable to your review package.
5. Select a file. It displays in the Review Editor to the right.
6. Edit the file.
7. In the upper-right corner, click **Submit**.

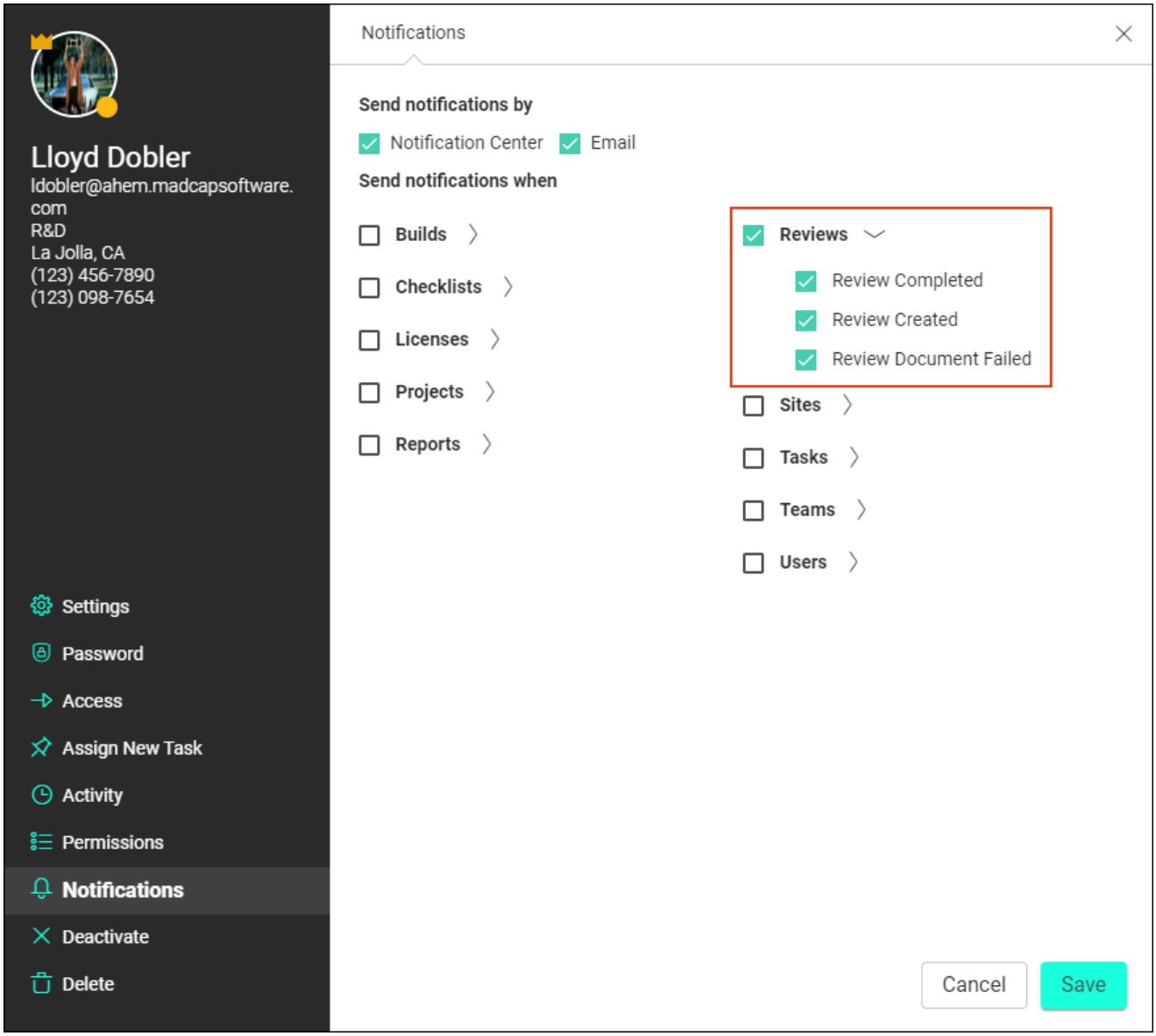


8. Repeat the steps for every file included in the review package (i.e., select the file, edit the file, and submit the file).

   The reviewer is done submitting once all the files in the queue are complete. This moves the review package to the Submitted pane. Files can be viewed from there and edited still, but once the owner closes the file, it can no longer be edited in Central.

Setting Notifications for Reviews

In your account settings, you can choose when to be notified for specific activities. The user who is notified depends on the activity. For more information, see the online Help.



Other Activities for Reviews

In addition to the main activities, there are some other tasks you might perform regarding this feature.

Managing Review Packages

Once review packages are in Central, SMEs and authors can edit, annotate, set status, and send files back to the owner.

In addition to the reviewing and editing capabilities, bundling files in a review package provides authors with options for managing packages and files in Central. How you use and control the available options depends on what is important to your specific review workflow.

Permission Required?

For this activity, you must have the following permission setting:

For more information about permissions, see the Central online Help.

Why Manage Reviews?

- Package Updates Review packages can be adjusted after they are created and sent for review. You can easily access the Review Package profile to add or delete files for a package, add or remove reviewers, change the TOC, or update the status of a package for any review package that is in Central.
- Revision History You can glean information from the Projects > Commits page since it acts as a history log for file versions that are connected to your Flare project. The commit ID is a reference point for when an item was created in the repository regardless of what it includes (e.g., review packages, files).
- Timely Feedback Collaborative content reviews are a vital part in the documentation process. The review packages feature enables owners to keep an active eye on any reviews in progress. Stale statuses or date and time gaps can trigger owners to follow up on progress.
- Tracking Use the data grids to follow every aspect of a review package as it moves through the review process. You can click different path links in the Reviews interface to locate and find more information about projects, files, or users in Central. You can filter and sort data to quickly search statuses, names, dates, projects, etc.
- Viewing Status Virtually every component in Central reviews captures some sort of status. For example, status is reflected in the Review Explorer, the package overview, the right Side Panel, the grids, and the profiles. It ranges from what state the package is in, the overall status of all files in a package, to the individual reviewer status and where they are with a given file. Flare and Central work together to maintain the current state of files in a review package to their respective environments; giving momentum to the review process.

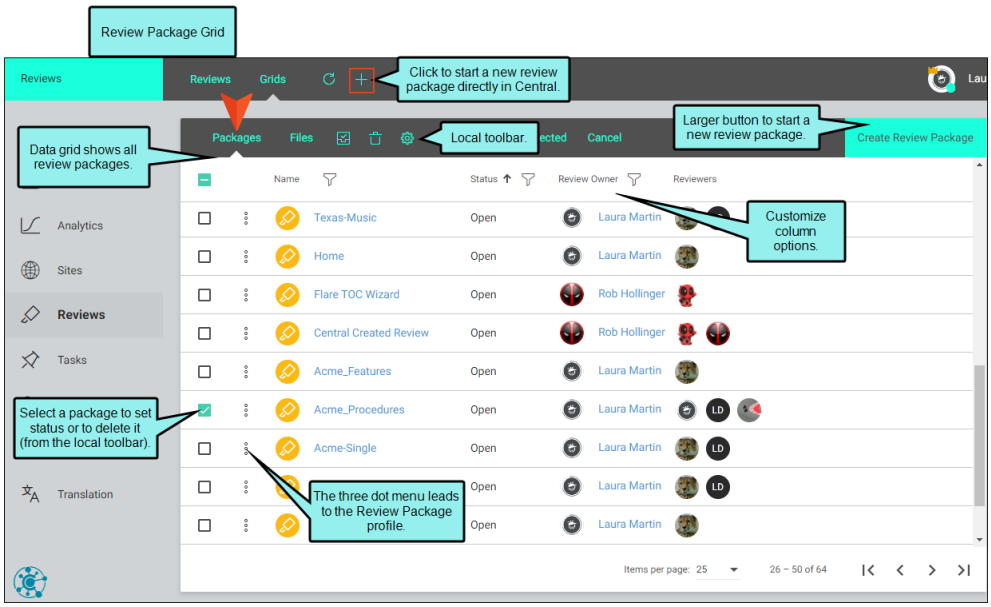
Packages and Files Grids

The Packages grid displays all the review packages in Central, while the Files grid shows all files sent for review regardless of the package. The grids provide a way to view, modify, or compare data. Each grid has a local toolbar, and most column elements are clickable, leading you to relevant pages, context menus, and profiles for further inspection of details and manipulation.

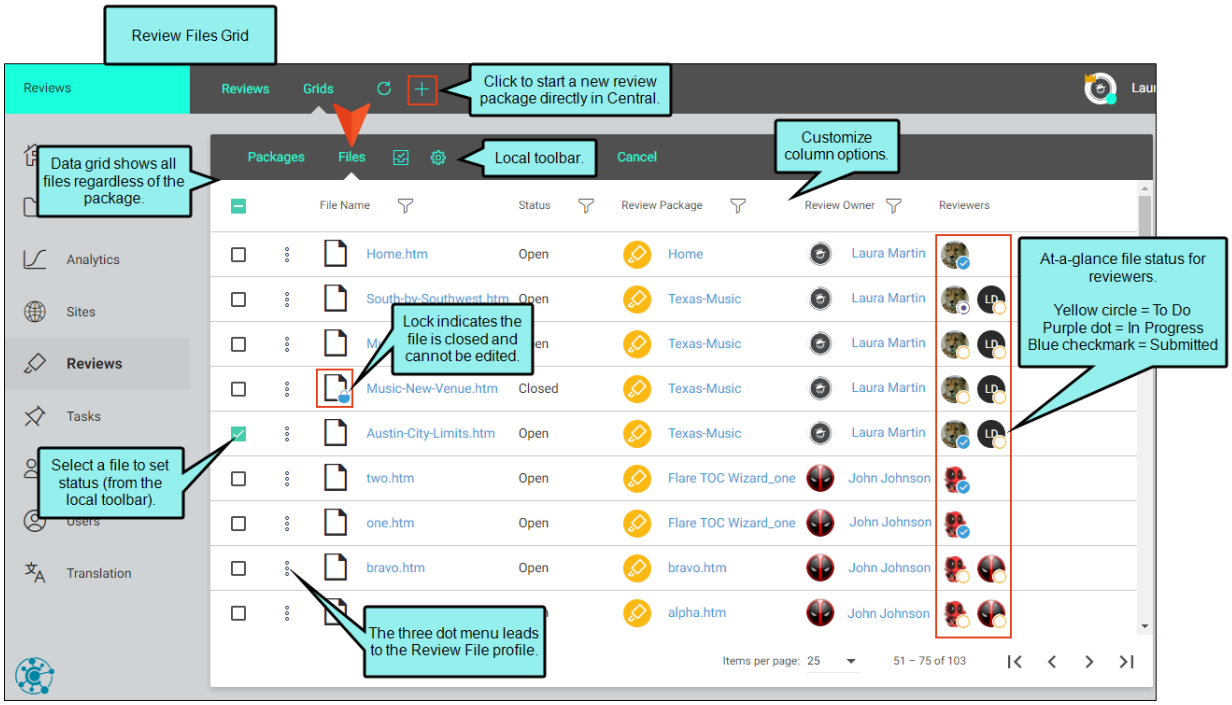
The main difference between the two grids is a matter of perspective—for managing reviews. You can do many of the same functions (e.g., adding reviewers, setting statuses, adding files), but the grid you use depends on if you want a package- or files-centric view of the project. Of course, there some things unique to each grid. For example, if you have a file in a package but are not sure which one, you can search the Files grid to find it.

- Packages Grid Displays every review package. In the Packages local toolbar you can set package status, delete review packages, and customize grid columns. Columns include: Name, Status, Review Owner, Reviewers, Project, Branch, Commit ID, Created On, Description, and TOC Path. Many of the columns include a filtering function for finding specific items. In the main toolbar, you can click to create a new review package.

  The Packages grid also includes a Create Review Package button where you can start a new package.



- **Files Grid** Displays all files sent for review. In the Files local toolbar you can set file status, and customize grid columns. Columns include: File Name, Status, Path, Review Package, My Status, My Submitted Date, All Users Status, All Users Submitted Date, Review Owner, Created On, Reviewers, Project, Branch, and Commit ID. Many of the columns include a filtering function for finding specific items. In the main toolbar, you can click to create a new review package.

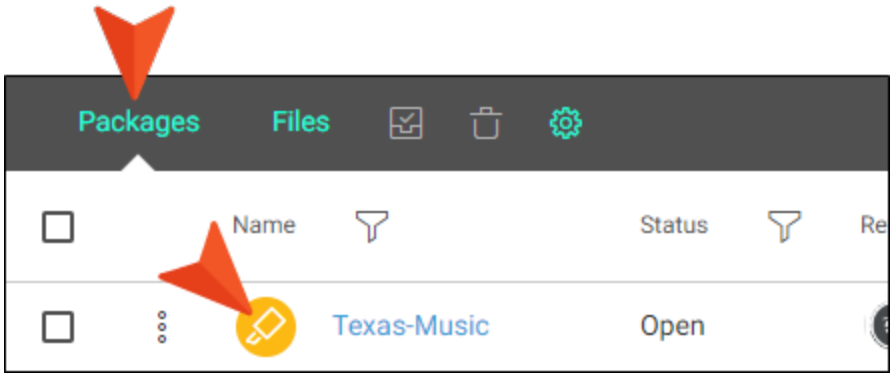


Updating Review Packages

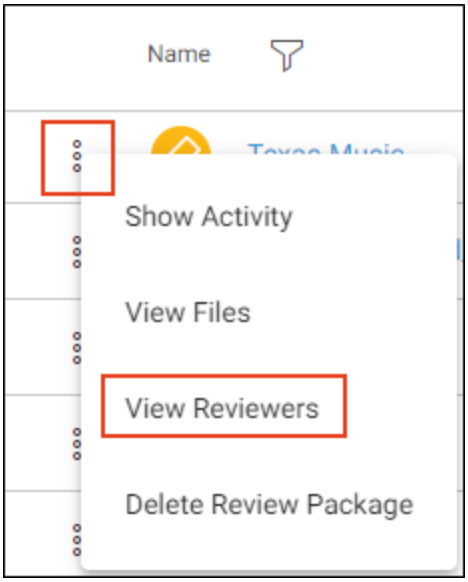
With the permission to manage reviews, you can make changes to a review package after it has been created and uploaded to Central.

How to Add or Remove Reviewers

1. On the left side of the interface, click **Reviews**.
2. From the top navigation, click **Grids**.
3. Select **Packages**.
4. For the review package you want to add reviewers to or remove reviewers from, select its icon (under the **Name** column). The Review Package profile opens.

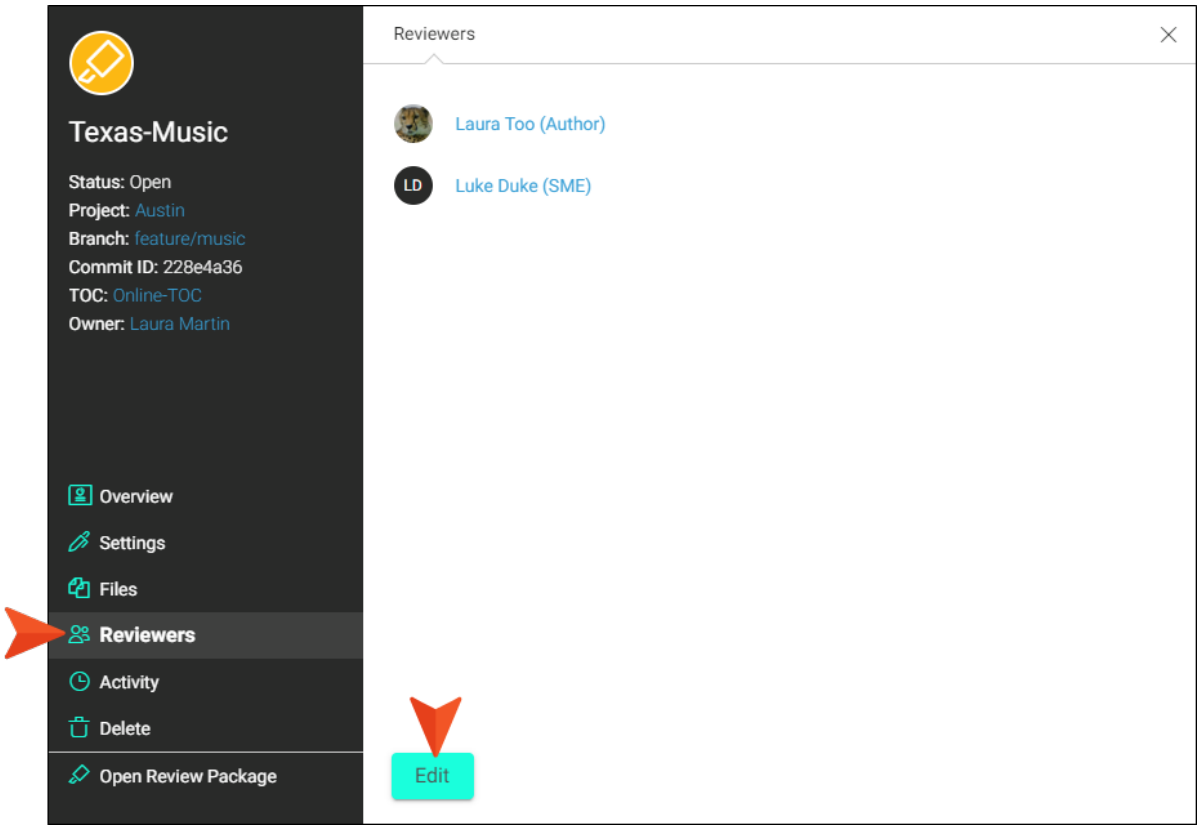


NOTE Alternatively, you can click the three dot vertical menu and select **View Reviewers**. This way also opens the Review Package profile, directly to the Reviewers option.



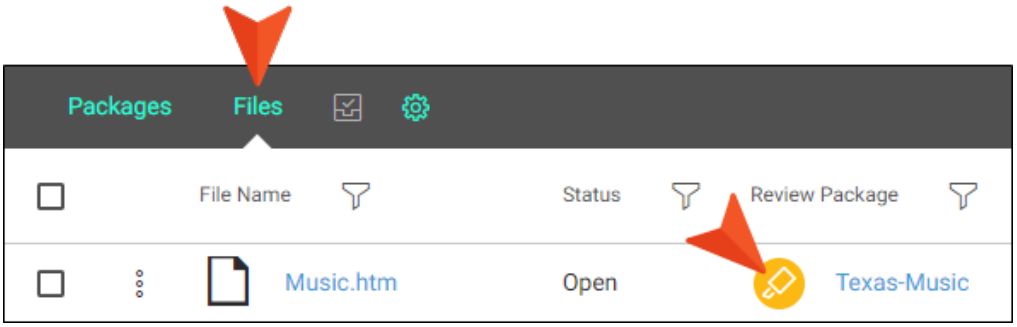
5. Select **Reviewers**.

6. On the right, the current reviewers are listed. Select **Edit**.



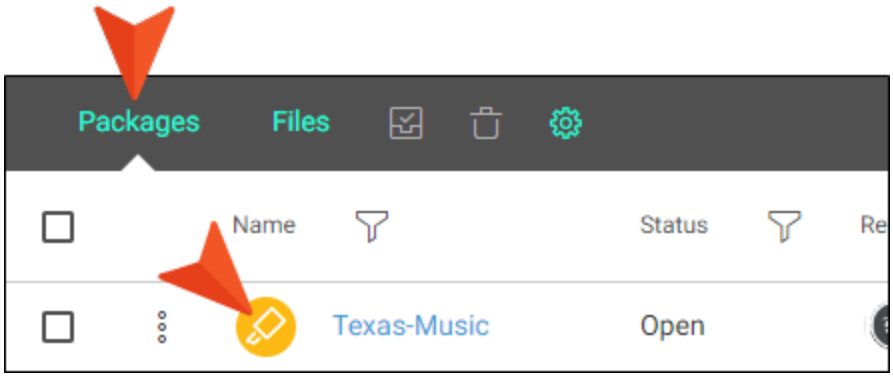
7. From the list of available reviewers, select to add a reviewer (or deselect to remove a reviewer.) Click **Save**.

> **NOTE** You can also use the Files grid to add or remove reviewers. The process is the same except you would select the review package icon under the Review Package column.

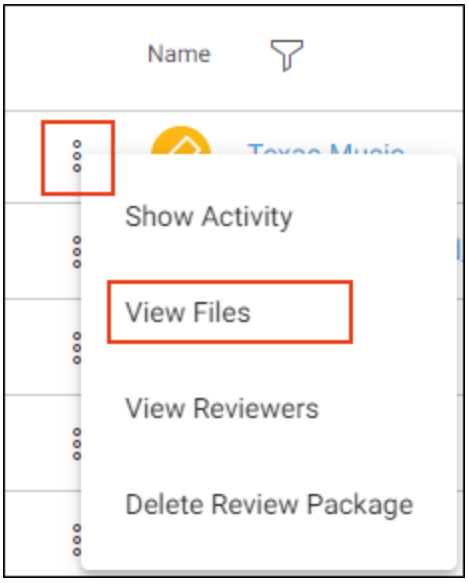


How to Add or Remove Files

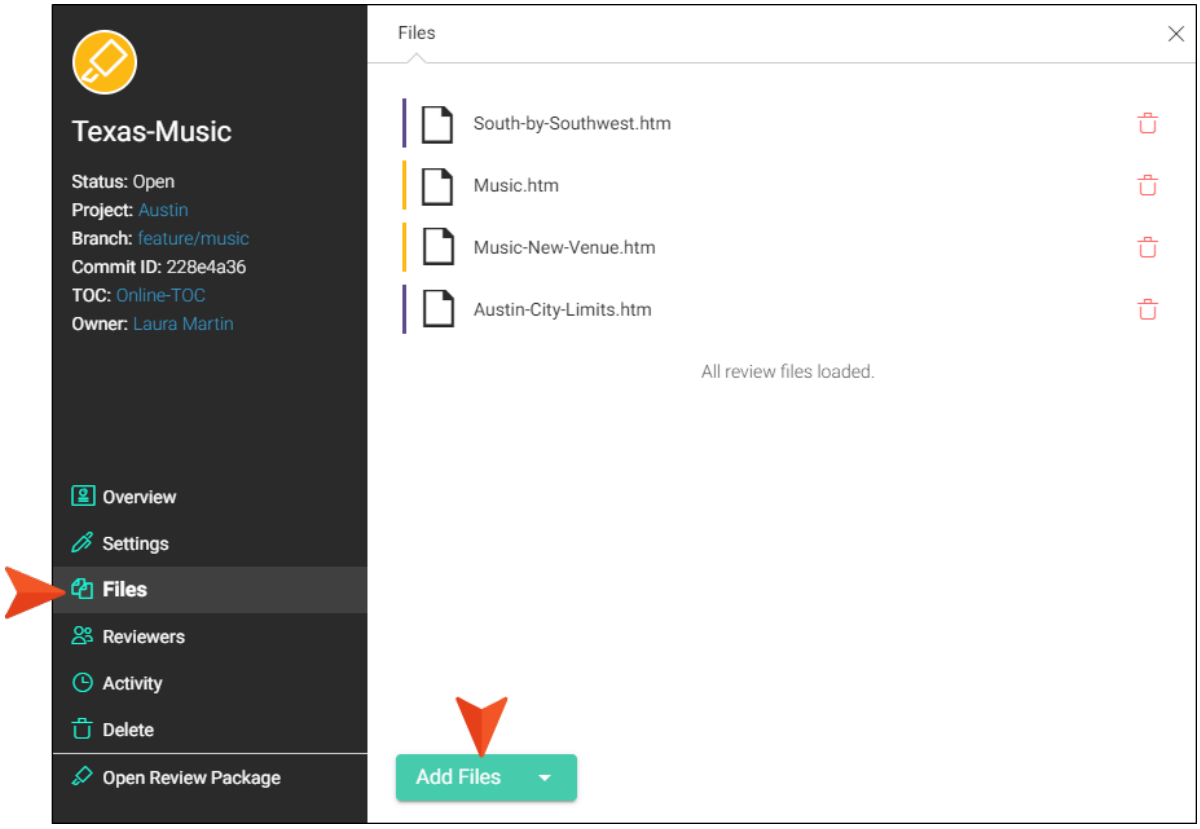
1. On the left side of the interface, click **Reviews**.
2. From the top navigation, click **Grids**.
3. Select **Packages**.
4. For the review package you want to add files to or remove files from, select its icon (under the **Name** column). The Review Package profile opens.



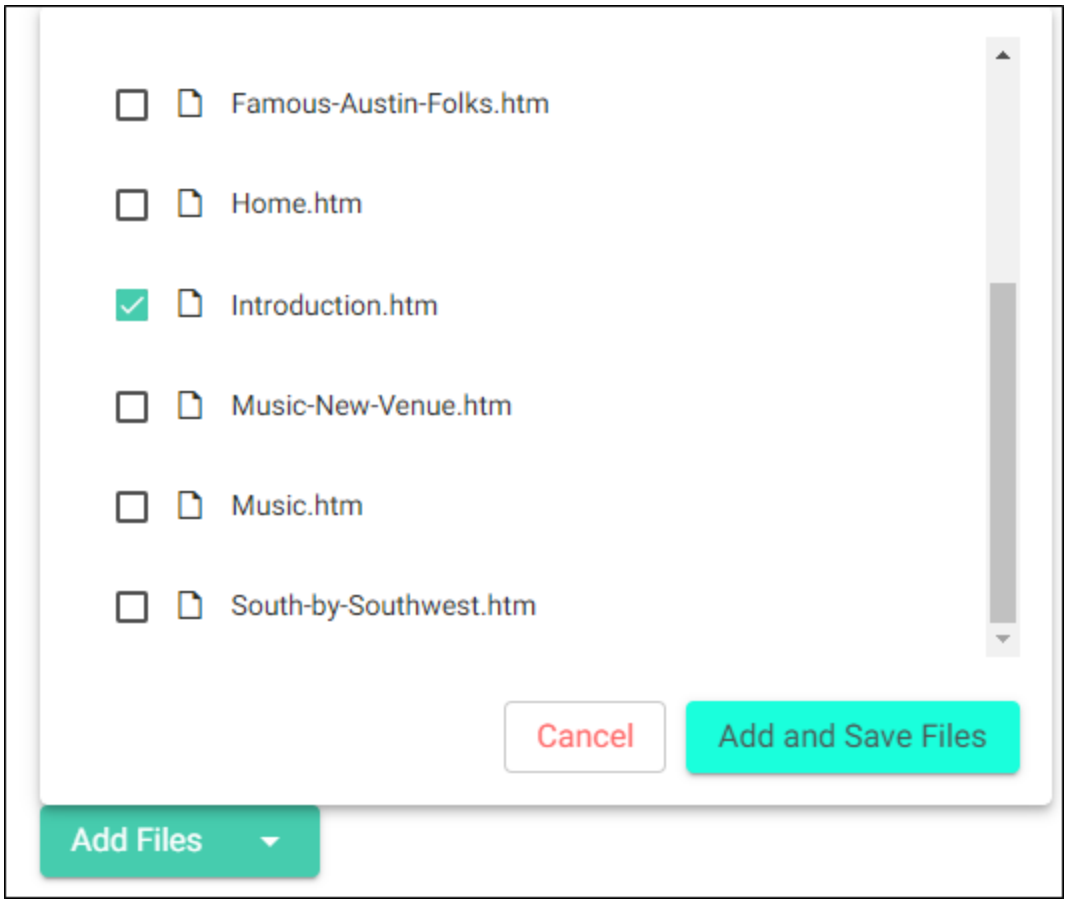
NOTE Alternatively, you can click the three dot vertical menu and select **View Files**. This way also opens the Review Package profile, directly to the Files option.



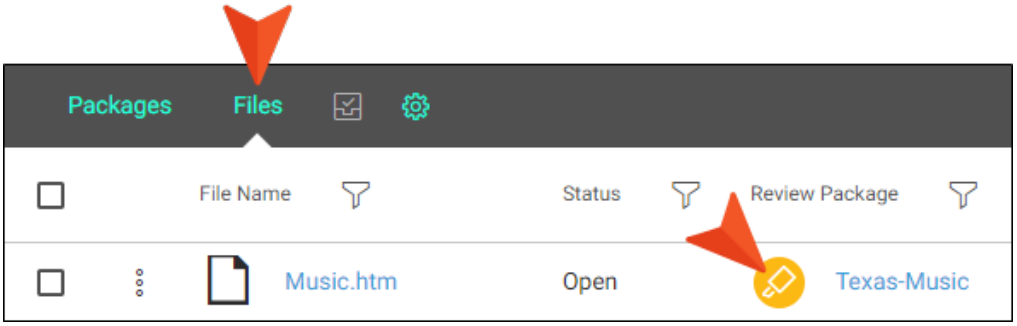
5. Select **Files**.
6. On the right, the current files are listed. Select **Add Files**. Also, to delete a file, you can click the trash icon to the right of the file.



7. From the popup, select other files from the project to add to the review (or deselect to remove files.) Click **Add and Save Files**.



NOTE You can also use the Files grid to add files. The process is the same except you would select the review package icon under the Review Package column.

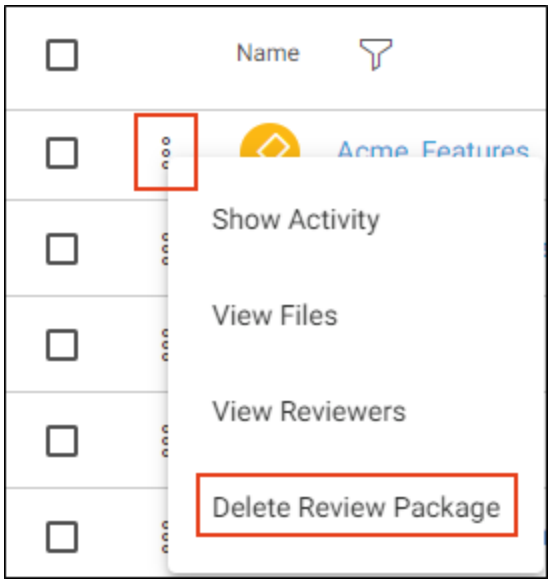


How to Delete a Package

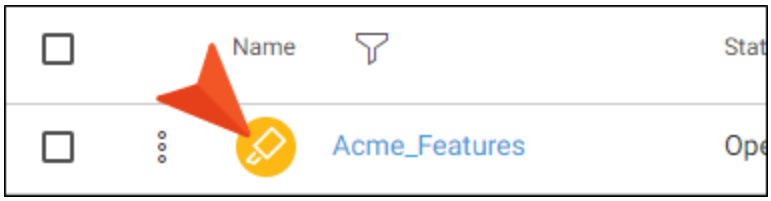
1. On the left side of the interface, click **Reviews**.
2. From the top navigation, click **Grids**.
3. Select **Packages**.

4. Do one of the following:

   - From the package row click the vertical three dot menu, and select **Delete Review Package**. The Review Package profile opens to the Delete option.



   - From the package row, click the package name icon.

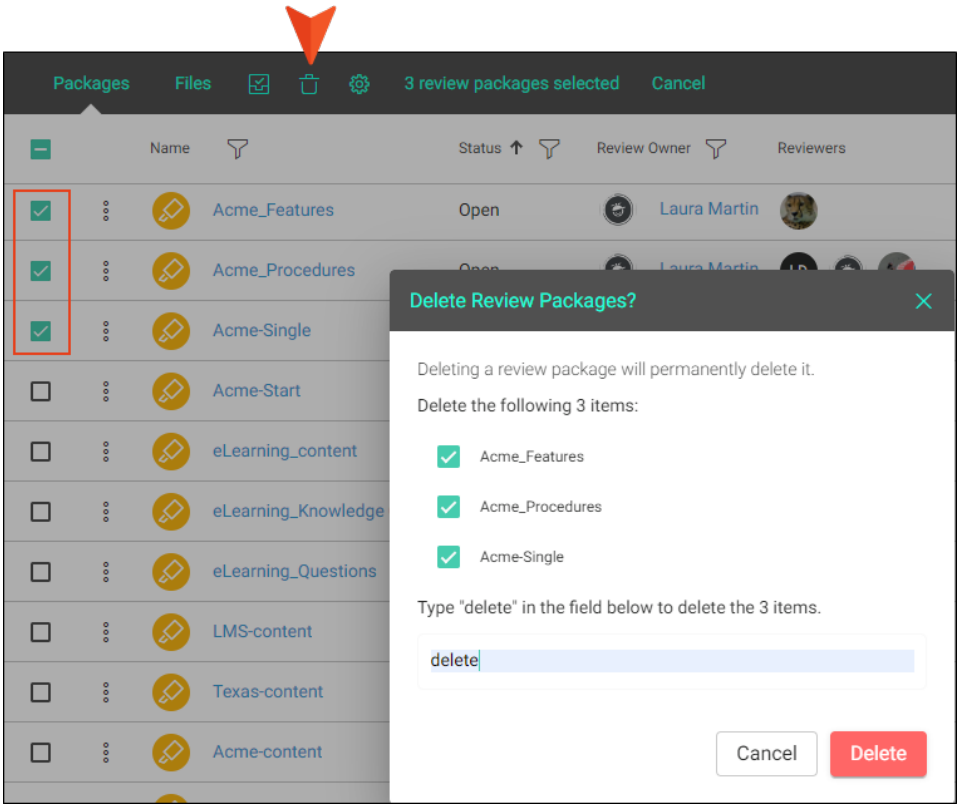


   The Review Package profile opens to the Overview option. Select the **Delete** Option.

5. Confirm the delete action, and click **Delete**. Deleting a review package cannot be undone.

NOTE Alternatively, you can select one or multiple review packages to delete.

1. Select the package row. This action enables options in the top toolbar.
2. Click **Delete Review Packages**.
3. In the dialog, confirm the delete action, and click **Delete**.



NOTE If you add or remove files associated with a review package, keep the following in mind—deleting a is a "soft" delete. If you add the same file back, it "undeletes" the file. But if a reviewer is added or removed while it is deleted, those changes are reflected in the package if the file is added back. If a package is closed and then reopened, you can opt to reset file and reviewer statuses to the To Do state.

Creating Review Packages

There are a few ways to create review packages (e.g., from Flare to Central, directly in Central). No way is better than the other but there are some considerations. For example, if you are using Flare, creating a review package in Flare is convenient. If a TOC is included in a review package, reviewers have the added benefit of being able to view files for review in the TOC view. When a package is created from Flare's TOC Editor, it is favorable to select single or multiple TOC nodes. Conversely, if you are working in Central, it might be better to create a review package in Central. The disadvantage is that you select individual files for review, and then separately select a TOC.

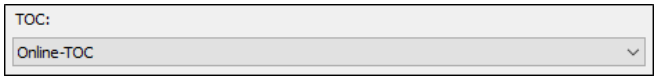
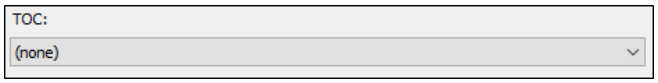
Permission Required?

For this activity, you must have the following permission setting:

For more information about permissions, see the Central online Help.

Sending Review Packages (Central vs. Flare)

Send Review Packages Via...	Caveats
Central's Review Package Wizard	You first select files for review, and then select a TOC. When selecting files, you need to know in advance what TOC those files are associated with to select it in the subsequent wizard screen. Selecting a TOC is optional, and only necessary if you want reviewers to be able to review files in the TOC view.

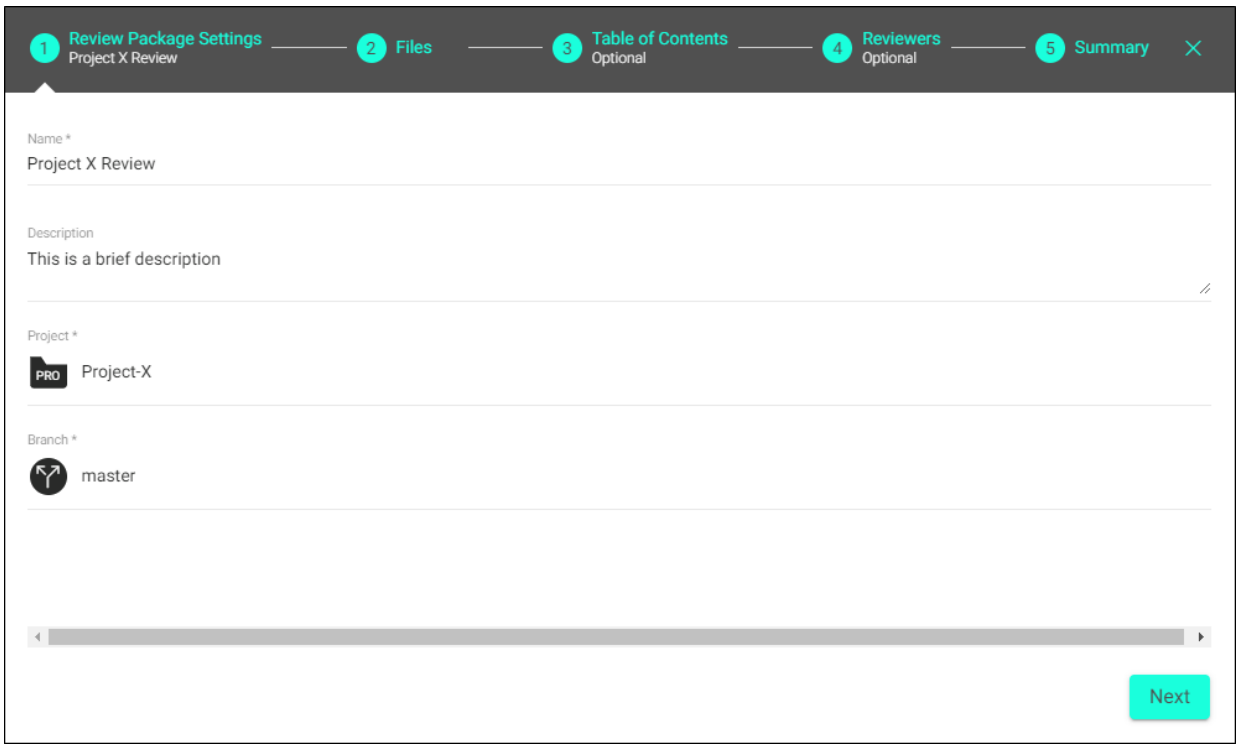
Send Review Packages Via...	Caveats
Flare's TOC Editor	Sends individually selected files or multiple files and folder nodes as a review package with a TOC associated with it. This enables reviewers to open files in Central's TOC view which provides better context for content. In this case, the TOC drop-down in the Send Files for Review Wizard defaults to pick the TOC that the wizard is initiated from. 
Flare's Review Ribbon	Sends individually selected files, or folders with multiple files selected within them as review packages. The files have no association with a TOC. In this case, the TOC drop-down in the Send Files for Review Wizard defaults to (none). However, you can choose a TOC for the review package if you want to (for reviews to display in a TOC view in Central). 

NOTE If you decide to create a review package in Central, keep in mind that even though the process starts in Central, it still ends in Flare (just like if you initiated the process from Flare).

How to Create a Review Package in Central

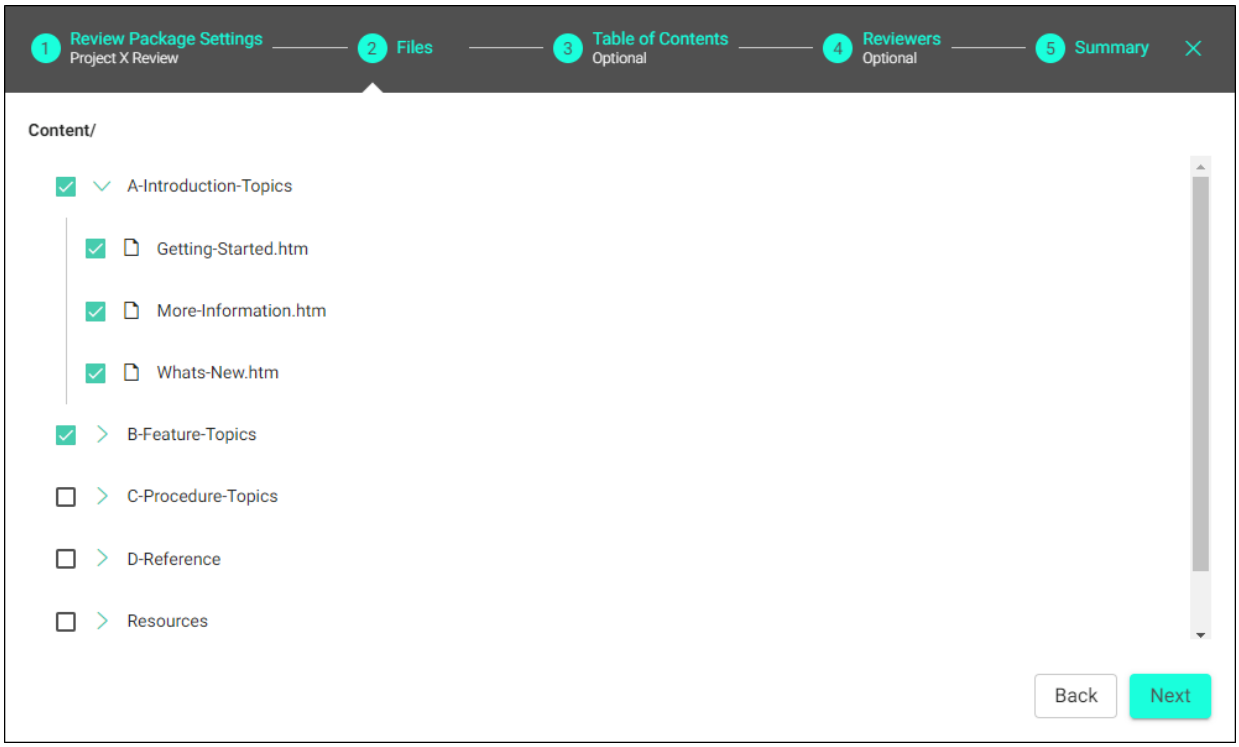
Owners can initiate the review process in Central directly, bypassing Flare as the starting point.

1. On the left side of the interface, click **Reviews**.
2. From the top navigation, click **Grids**.
3. Select **Packages** (if not already selected).
4. In the upper-right corner, select **Create Review Package**. The wizard opens.
5. At a minimum, enter a **Name**, and select a **Project** and **Branch**. Click **Next**. You can optionally enter a review package description.

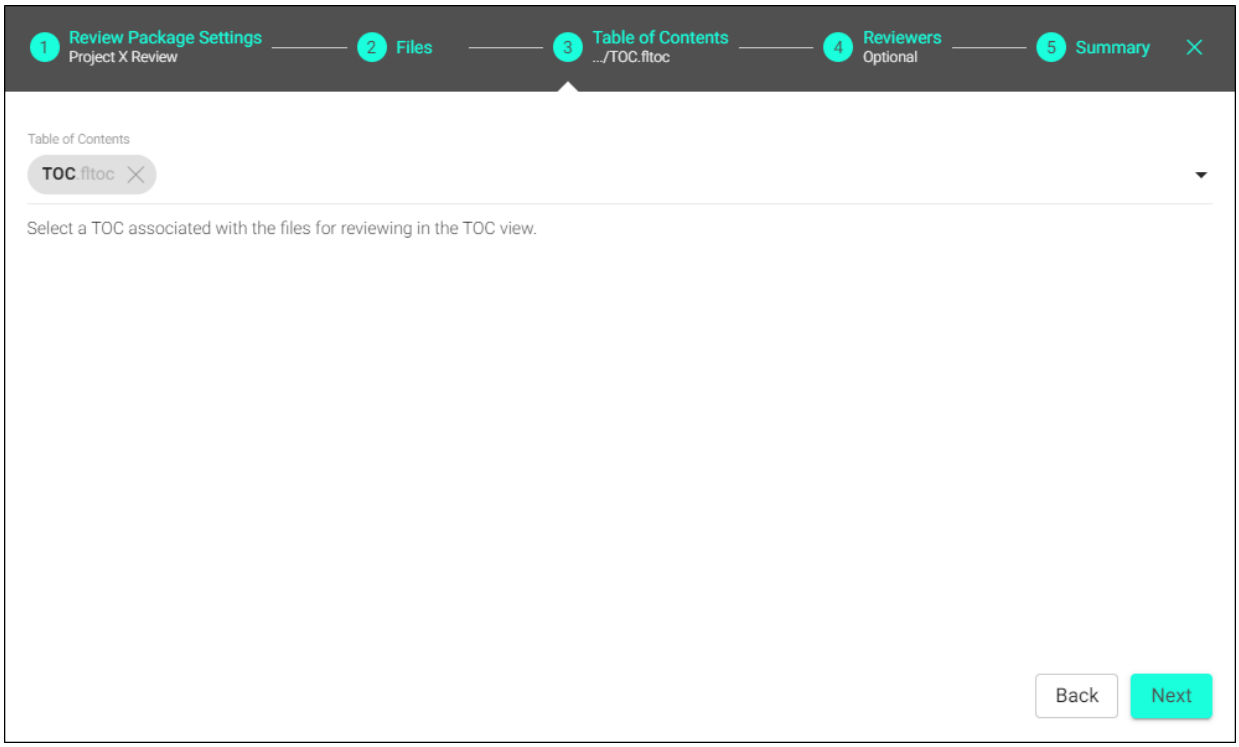


> **NOTE** You can only create a review package for a project in Central that you are a part of (i.e., a user or a team member of the project).

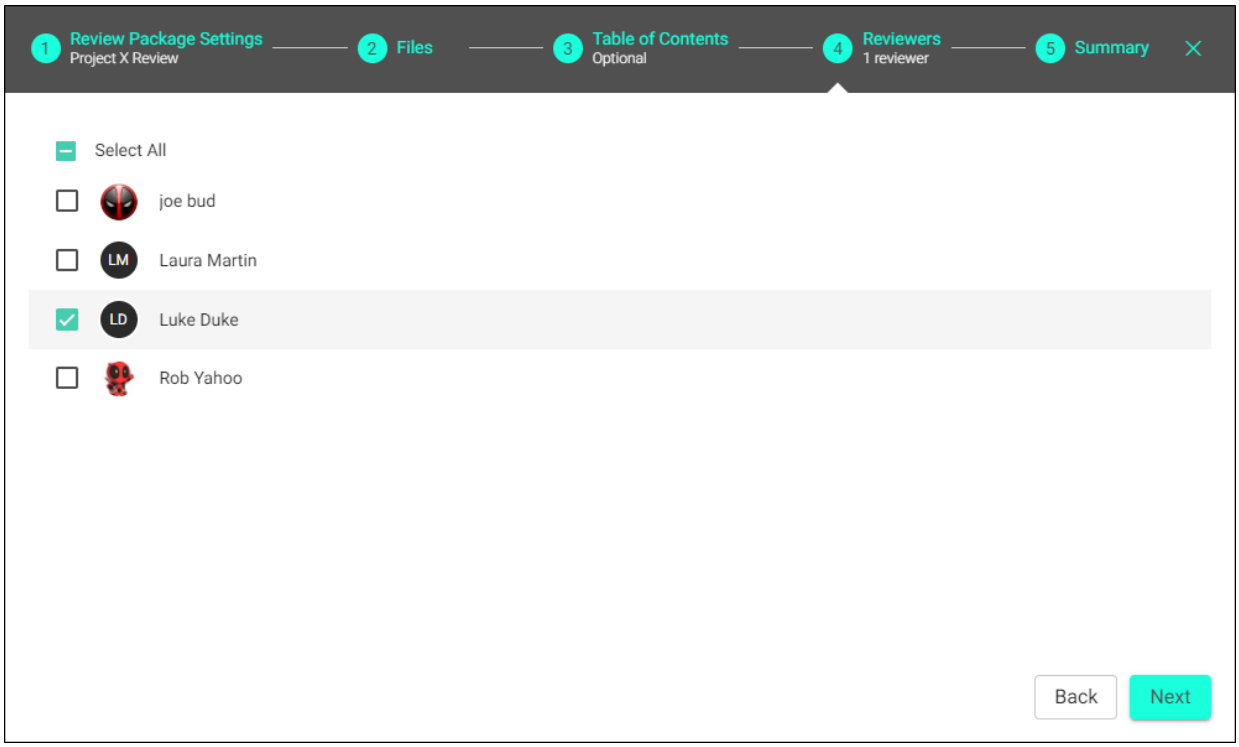
6. Expand the content folders to select files for the package. You can also select a folder to choose everything under it. Click **Next**.



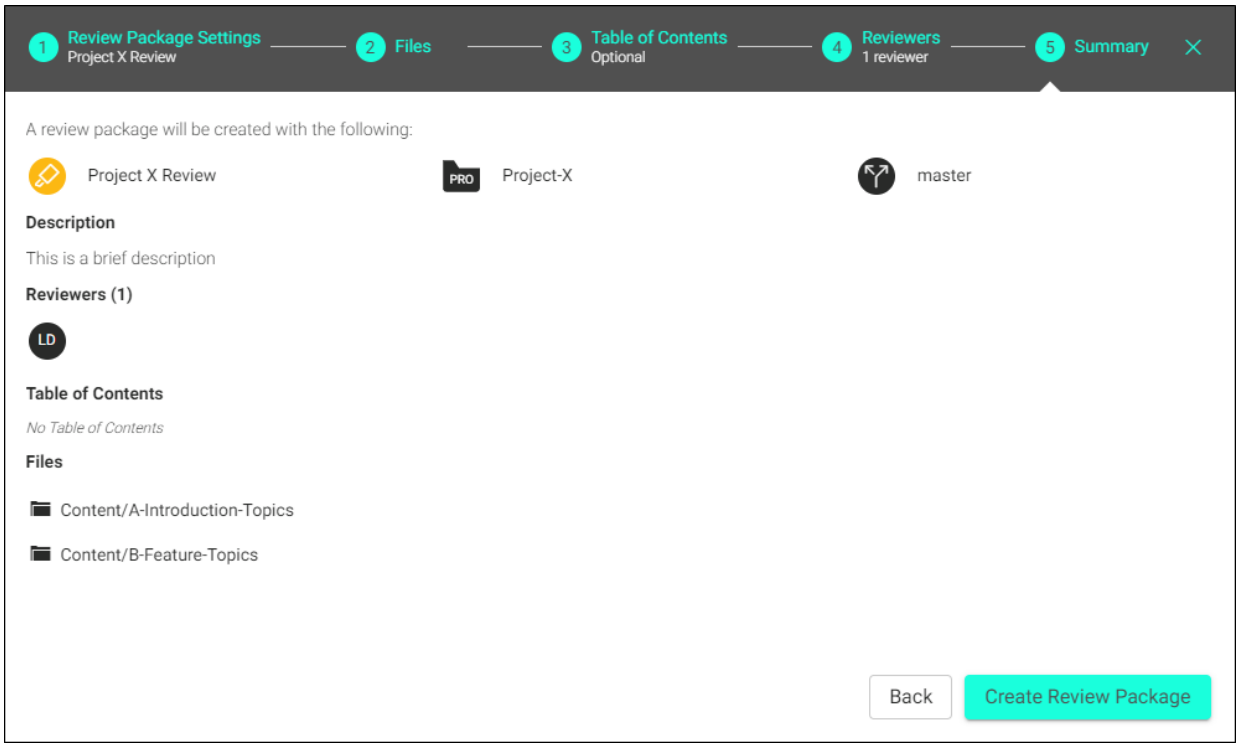
7. (Optional) You can choose a **Table of Contents** that is associated with the files selected. This is only necessary if you want reviewers to be able to review files using the TOC view in Central. Click **Next**.



8. (Optional) Select reviewers. Click **Next**.



9. Review the summary of the package. Click **Create Review Package**. The new review package displays in Central's Reviews page.



NOTE You can optionally click the **Back** button on any of the wizard screens to change review package information before creating it.

NOTE Depending on what you need to do, keep in mind that you can edit existing review packages rather than create new ones. For example, you can add and delete files or add and remove reviewers using the Review Packages profile.

Viewing Review Profiles

For authors, the Review Package profile and the Review File profile are available to view package review settings. In order to make changes, then having permission to manage reviews is required.

Permission Required?

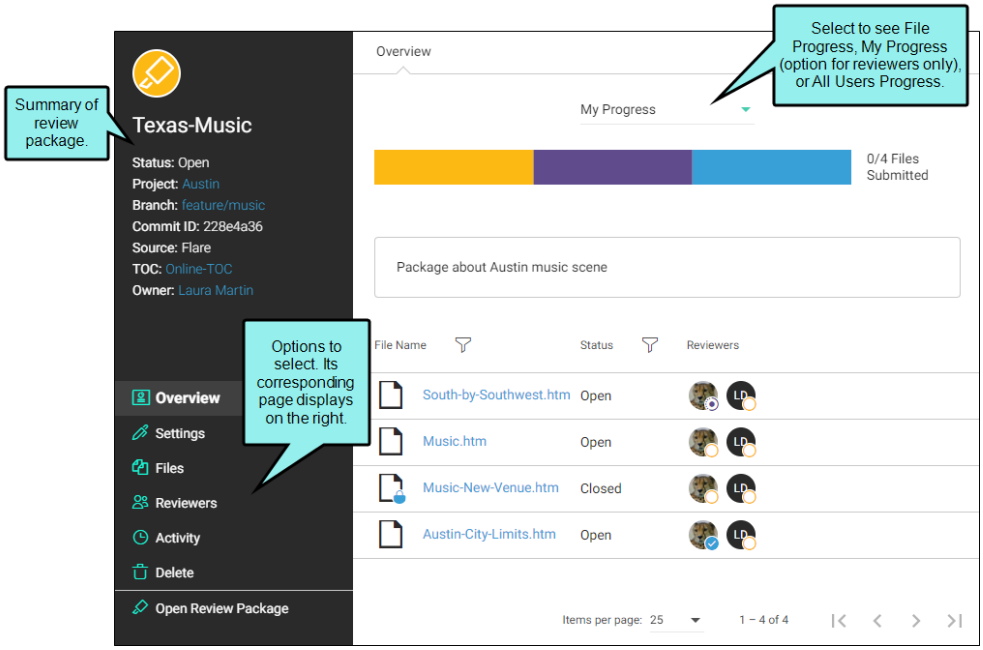
For this activity, you must have the following permission setting:

☑ Manage Reviews

For more information about permissions, see the Central online Help.

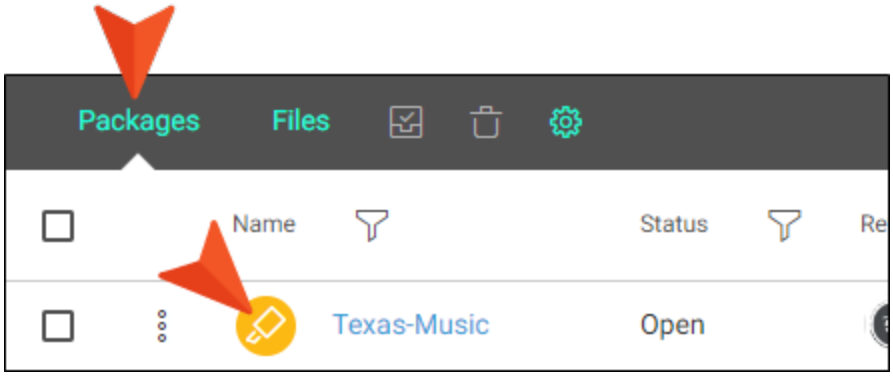
Review Package Profile

You can view a profile for a review package. It snapshots a working overview of the package and allows you to perform activities for managing a review package. Some useful tasks include adding or deleting files, adding or removing reviewers, and setting package status. Note that if a review package is set to closed, then it closes all the files within it.



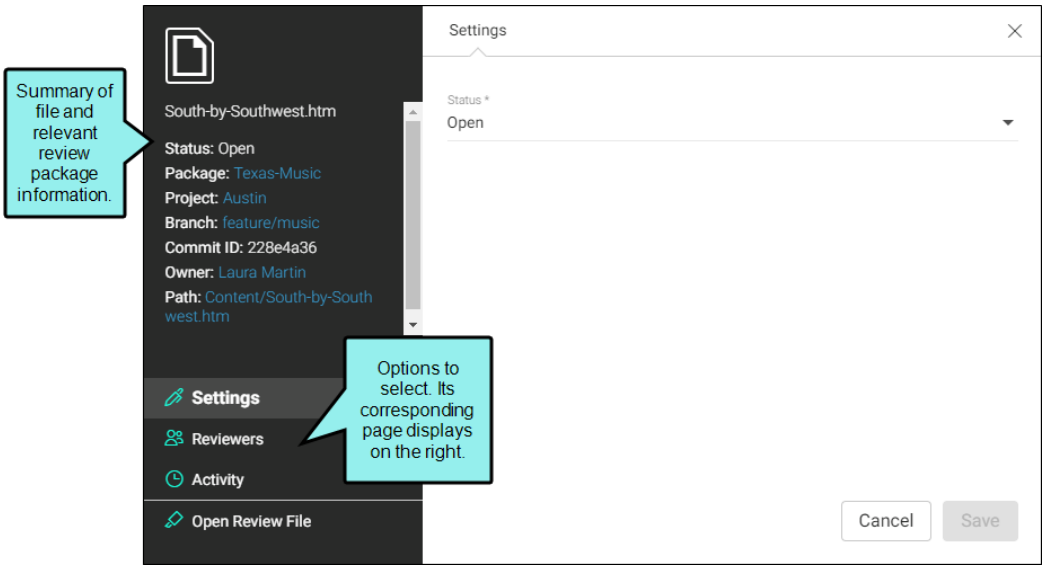
How to View a Review Package Profile

1. On the left side of the interface, click **Reviews**.
2. From the top navigation, click **Grids**.
3. Select **Packages**.
4. Click the review package icon (under the **Name** column).



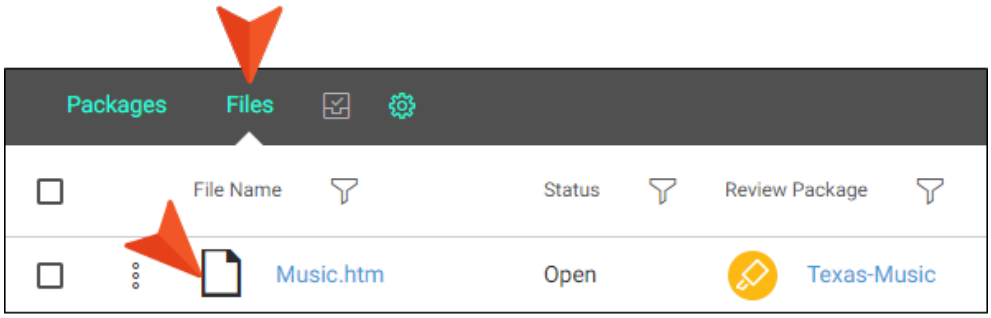
Review File Profile

You can view a profile for a file in a review package. It opens with settings that show the review status of the file. You can also see who's assigned as a reviewer for the file (and click to view the user profile), see recent file activity, and open the review file directly from the Review File profile.



How to View a Review File Profile

1. On the left side of the interface, click **Reviews**.
2. From the top navigation, click **Grids**.
3. Select **Files**.
4. Next to the file of interest, click the file icon (under the **File Name** column).



Updating Review Status

One of the benefits of having noticeable statuses throughout Reviews in Central, is for authors who are managing review packages to set statuses on-the-fly if necessary.

Permission Required?

For this activity, you must have the following permission setting:

For more information about permissions, see the Central online Help.

Available Review Statuses

Review Package Statuses

- Open The review package is open for review. This is the default setting.
- Closed All files in the package are set to a closed status (i.e., the package can no longer be reviewed). This can be set in the Review Package profile to put a review back into Flare's inbox as is (i.e., the review is not complete) and archive the review at the same time.

Review File Statuses

- Open The file is open for review. This is the default setting.
- Closed The file is closed for review.
- Accepted The file is accepted in Flare.

Reviewer Statuses

- To Do The reviewer has not started editing the file yet. The status color is yellow.
- In Progress The reviewer has started editing the file, but has not submitted it. The status color is purple.

- **Submitted** The reviewer has edited the file and submitted it. The reviewer can still edit the file until the owner closes it. The status color is blue.

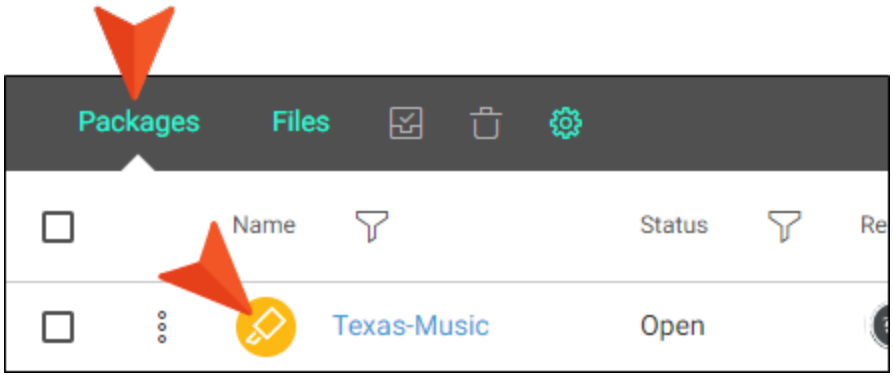
Setting Status

The status can be set singularly through the package or file profile dialogs, or in bulk through the set status dialogs.

How to Set the Package Status

In the review process, statuses are updated behind the scenes for the most part. However, there might be circumstances from a management perspective where a status setting needs to be updated manually.

1. On the left side of the interface, click **Reviews**.
2. From the top navigation, click **Grids**.
3. Select **Packages**.
4. For the review package you want to set the status on, select its icon (under the **Name** column). The Review Package profile opens.

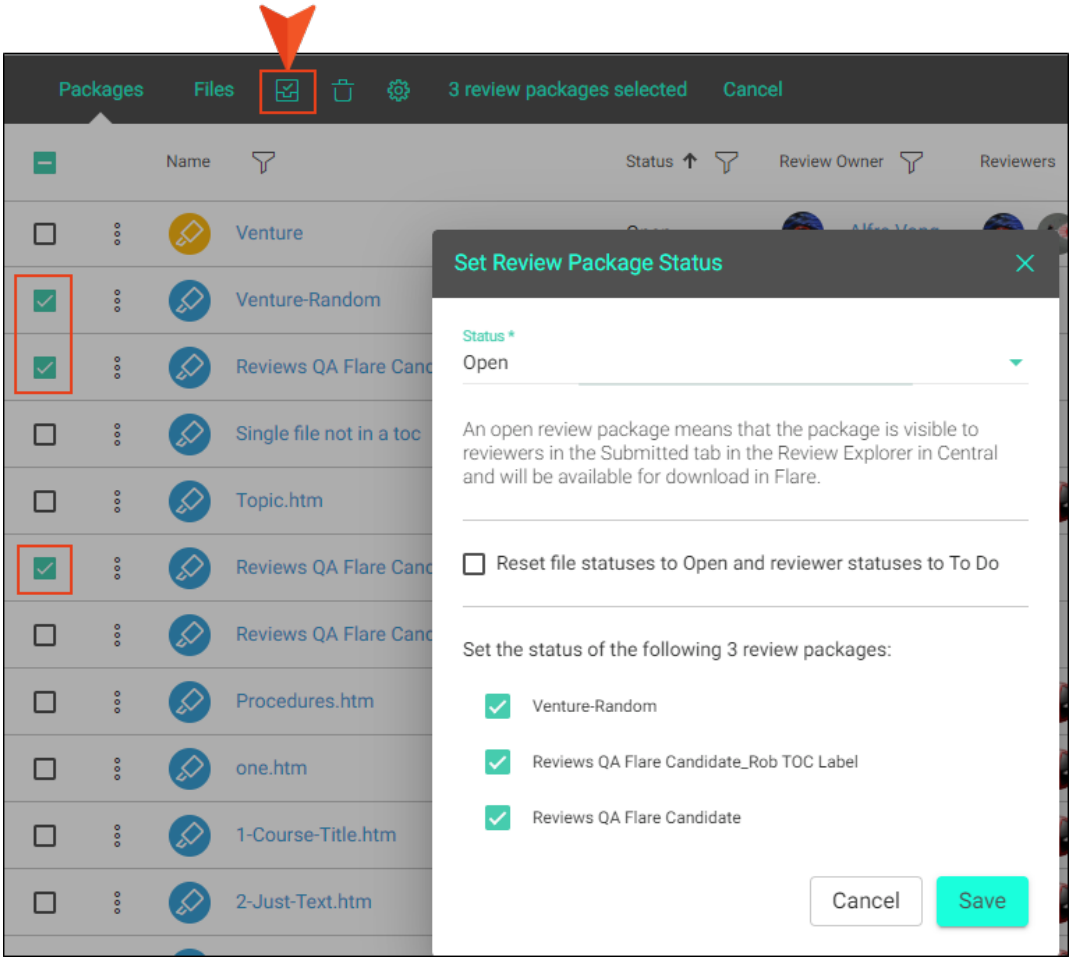


5. Select **Settings**.
6. On the right, select from the **Status** drop-down (i.e., open, closed). Click **Save**.

- **Open** The review package is open for review. This is the default setting.
- **Closed** All files in the package are set to a closed status (i.e., the package can no longer be reviewed).

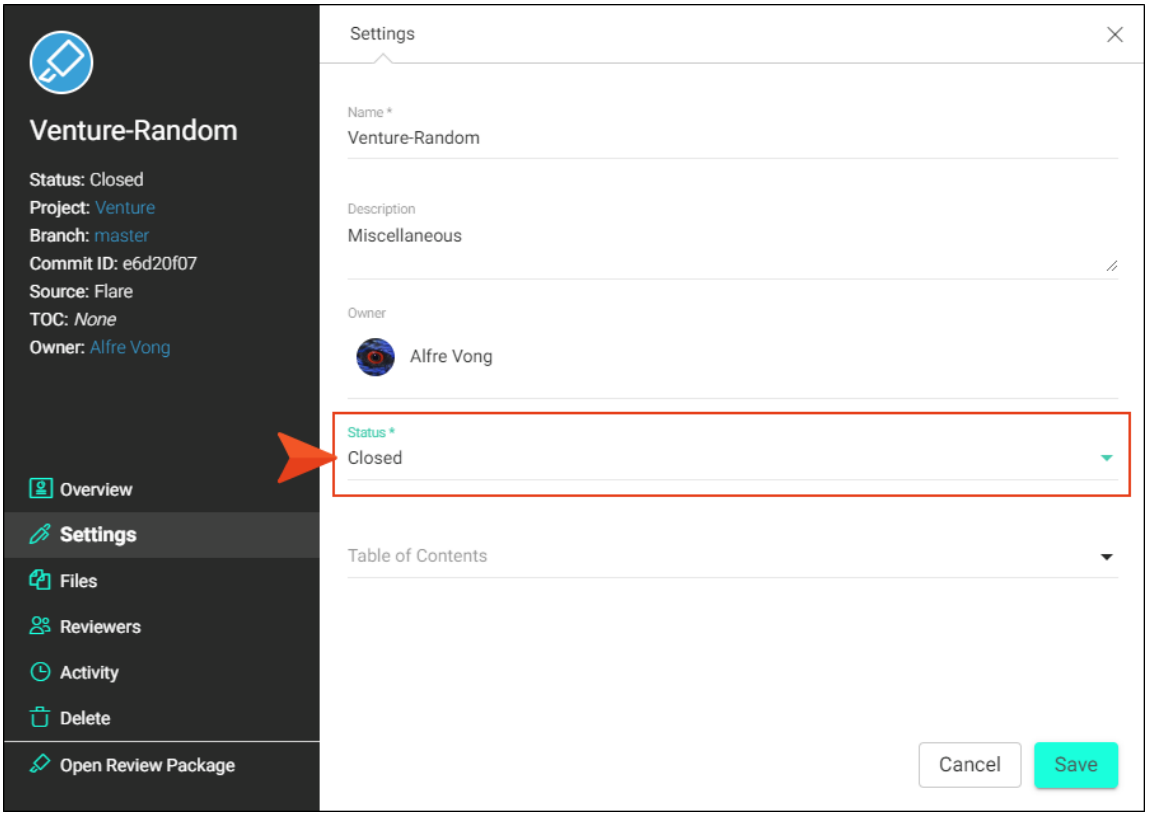
NOTE Alternatively, you can set the package status for multiple packages (in bulk) from the Packages grid local toolbar.

1. In the package row, select the check box.
2. From the local toolbar, select **Set review package status**.
3. In the Set Review Package Status dialog, select the drop-down and update the status.

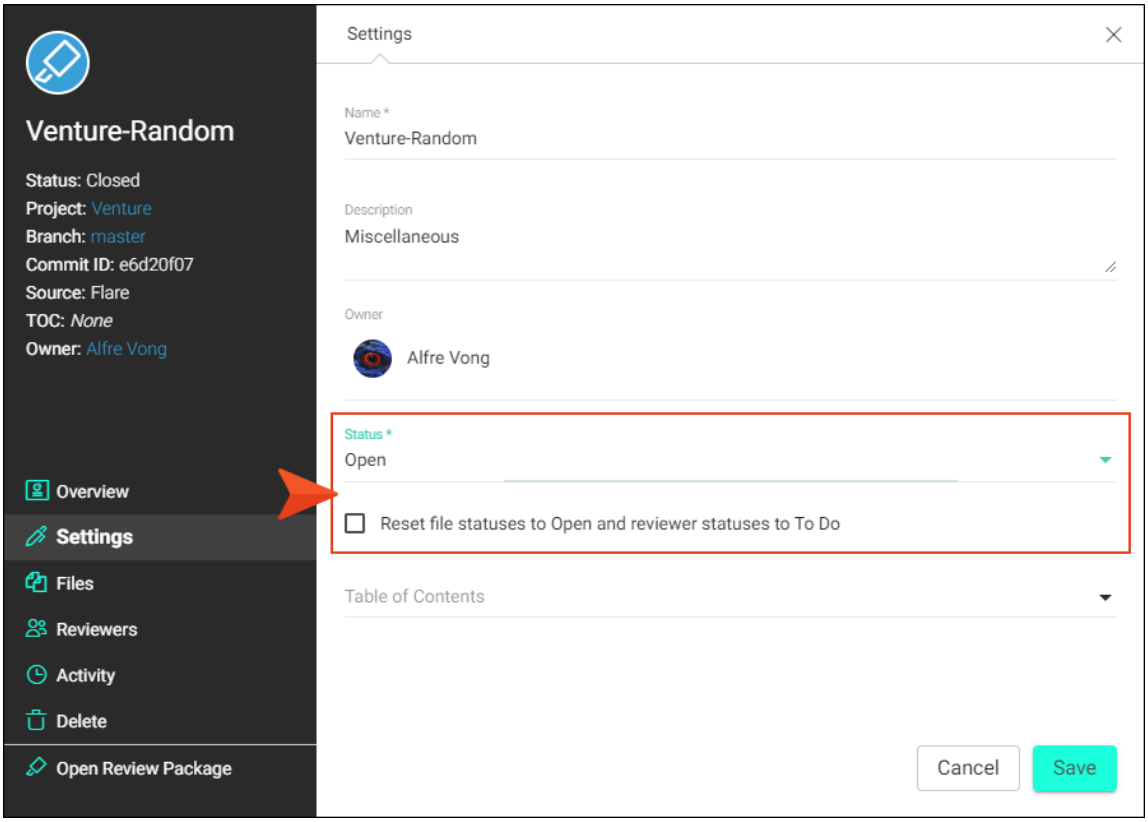


4. Click **Save**.

EXAMPLE If a reviewer is working a package that is "In Progress," and edits have been made, an author managing the review can set the package to a "Closed" state.



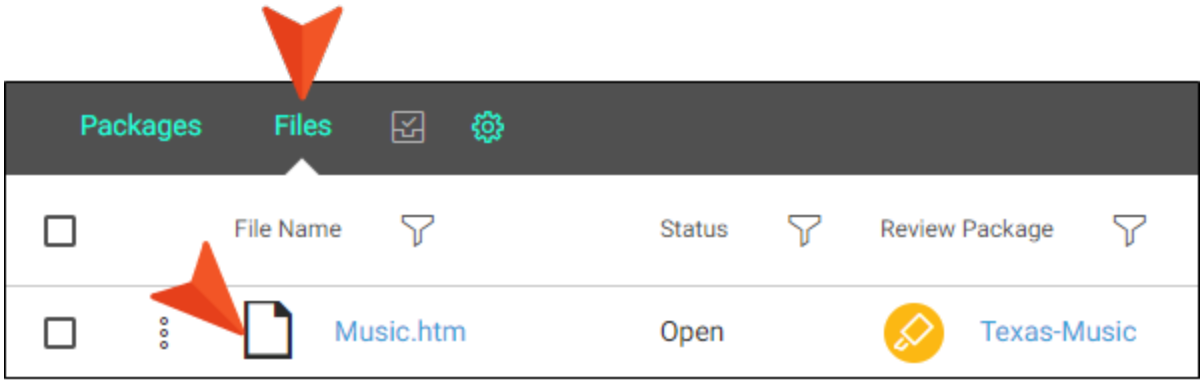
Then, the author could decide to set the package back into an "Open" state. Using the Settings option for updating status, the author has the choice to either keep the previous reviewer statuses by not selecting the given check box, or that person can select the check box and all of the files statuses and reviewer statuses will be set back to the original "To Do" status for that package.



How to Set the File Status

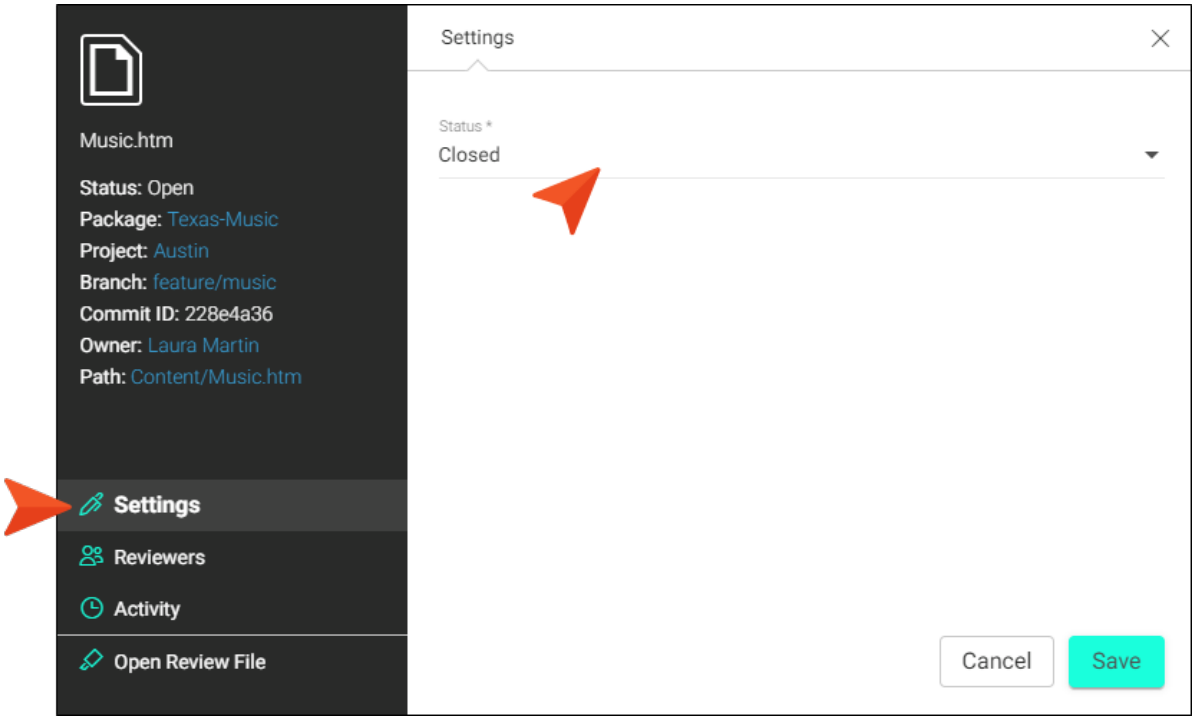
In the review process, statuses are updated behind the scenes for the most part. However, there might be circumstances from a management perspective where a status setting needs to be updated manually.

1. On the left side of the interface, click **Reviews**.
2. From the top navigation, click **Grids**.
3. Select **Files**.
4. For the file you want to change the status on, select its icon (under the **File Name** column). The Review File Profile opens.



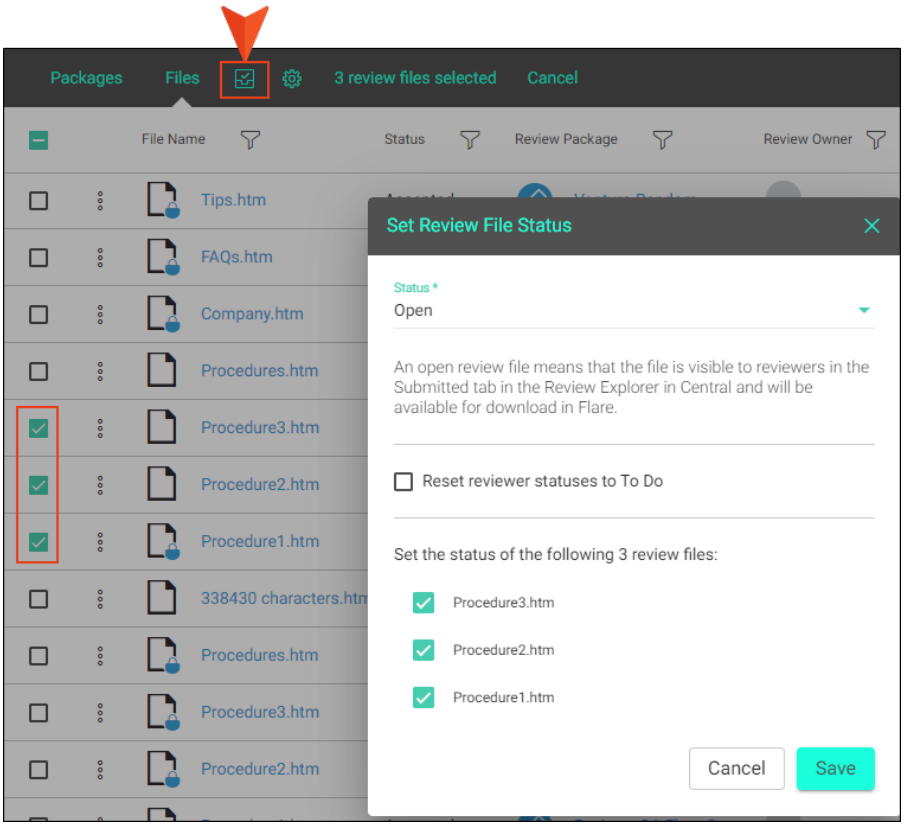
5. Select **Settings**.

6. On the right, select from the **Status** drop-down. Click **Save**.
    - **Open** The file is open for review. This is the default setting.
    - **Closed** The file is closed for review.
    - **Accepted** The file is accepted in Flare.



NOTE Alternatively, set the file status from the Files grid local toolbar.

1. In the file row, select the check box.
2. From the local toolbar, select **Set review file status**.
3. In the Set Review File Status dialog, select the drop-down and update the status.



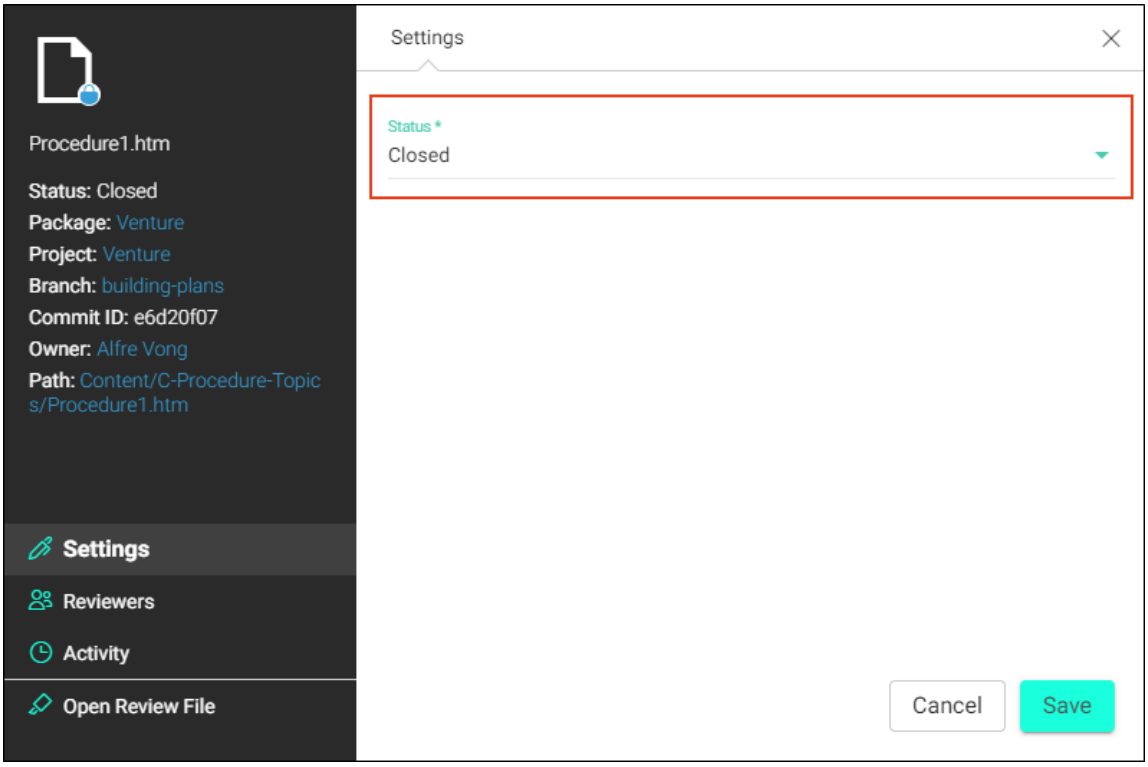
4. Click **Save**.

Even more options are:

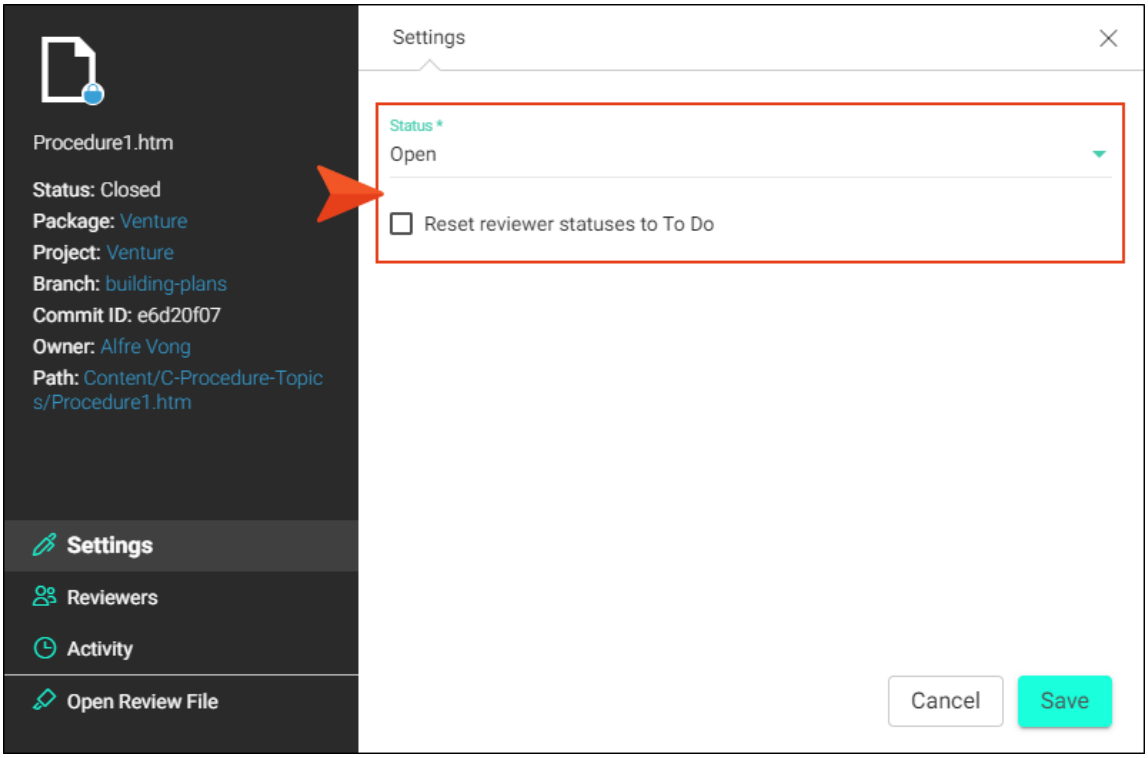
In the Files grid, you can click the review package icon under the Review Package column. Or in the Packages grid, you can click the review package icon under the **Name** column. In both cases, the Review Package profile opens. If you select the **Files** option, you can click the file icon to open the Review File profile to update file status.

NOTE From the reviewer's perspective, the changing of the status is done automatically as the reviewer works through the review. After contributing to the file, the reviewer can simply press the Submit button. Reviewers do not have to manually update their file status.

EXAMPLE If a reviewer is working a file that is "In Progress," and edits have been made, an author managing the review can set the file to a "Closed" state.



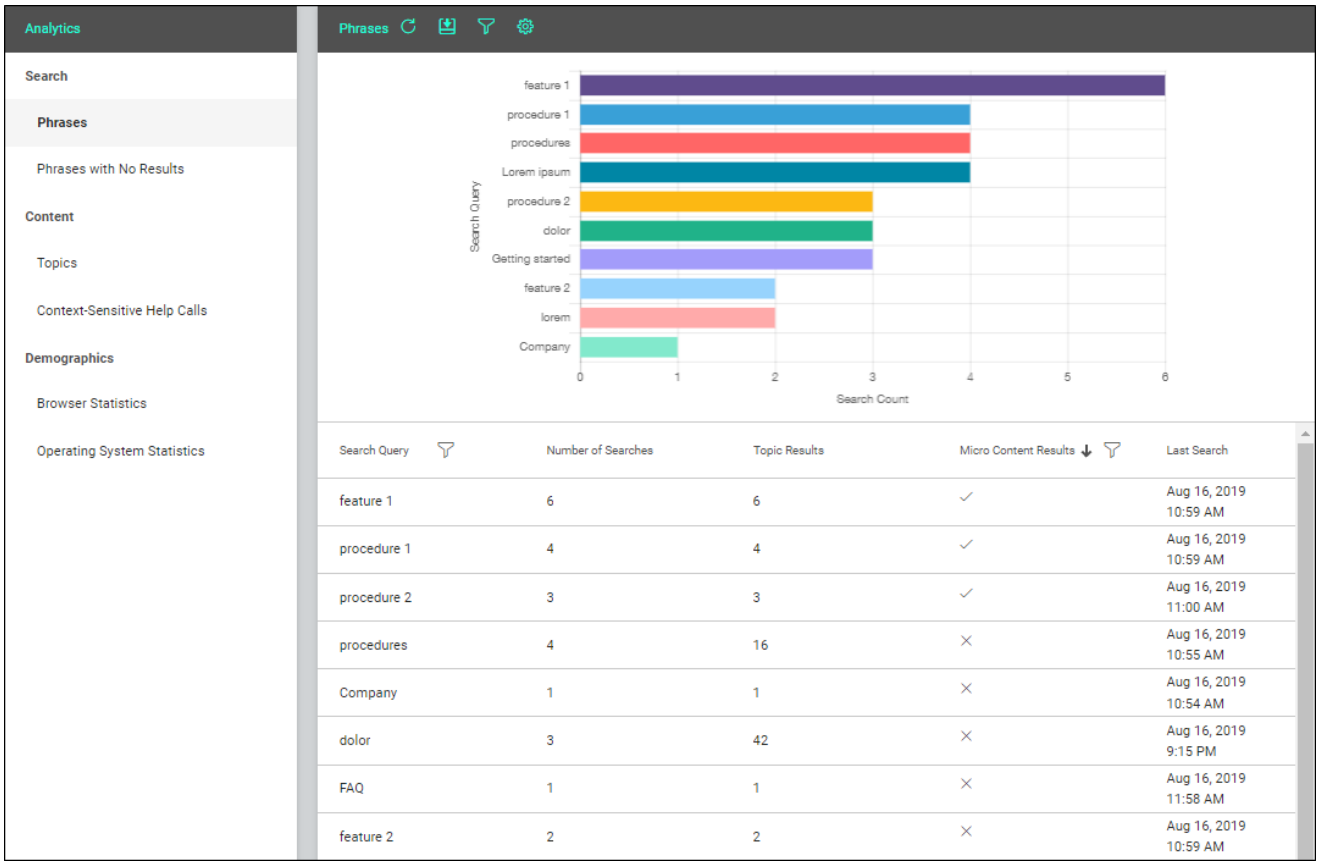
Then, the author could decide to set the file back into an "Open" state. Using the Settings option for updating status, the author has the choice to either keep the previous reviewer statuses by not selecting the given check box, or that person can select the check box and all of the reviewer statuses will be set back to the original "To Do" status for that file.



CHAPTER 12

Analytics

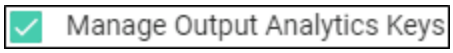
The Analytics page lets you view user activity on published Flare HTML5 output. This includes search phrases used, search phrases with no results, topics viewed, context-sensitive Help calls, and demographic statistics (browsers and operating systems).



This chapter discusses the following:

Permission Required?

To create and manage keys, you must have the following permission setting:



However, all users on the license can view analytics data, regardless of their permission settings.

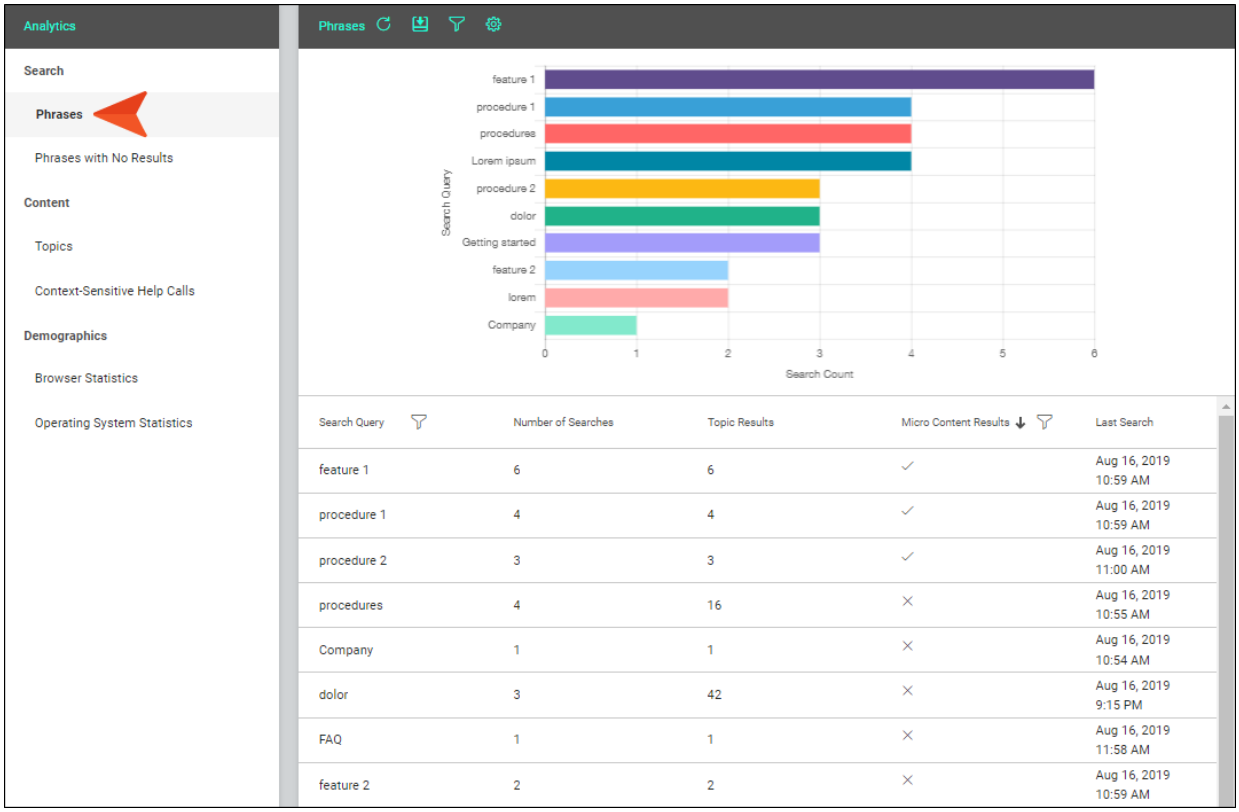
For more information about permissions, see the Central online Help.

Types of Analytics Information

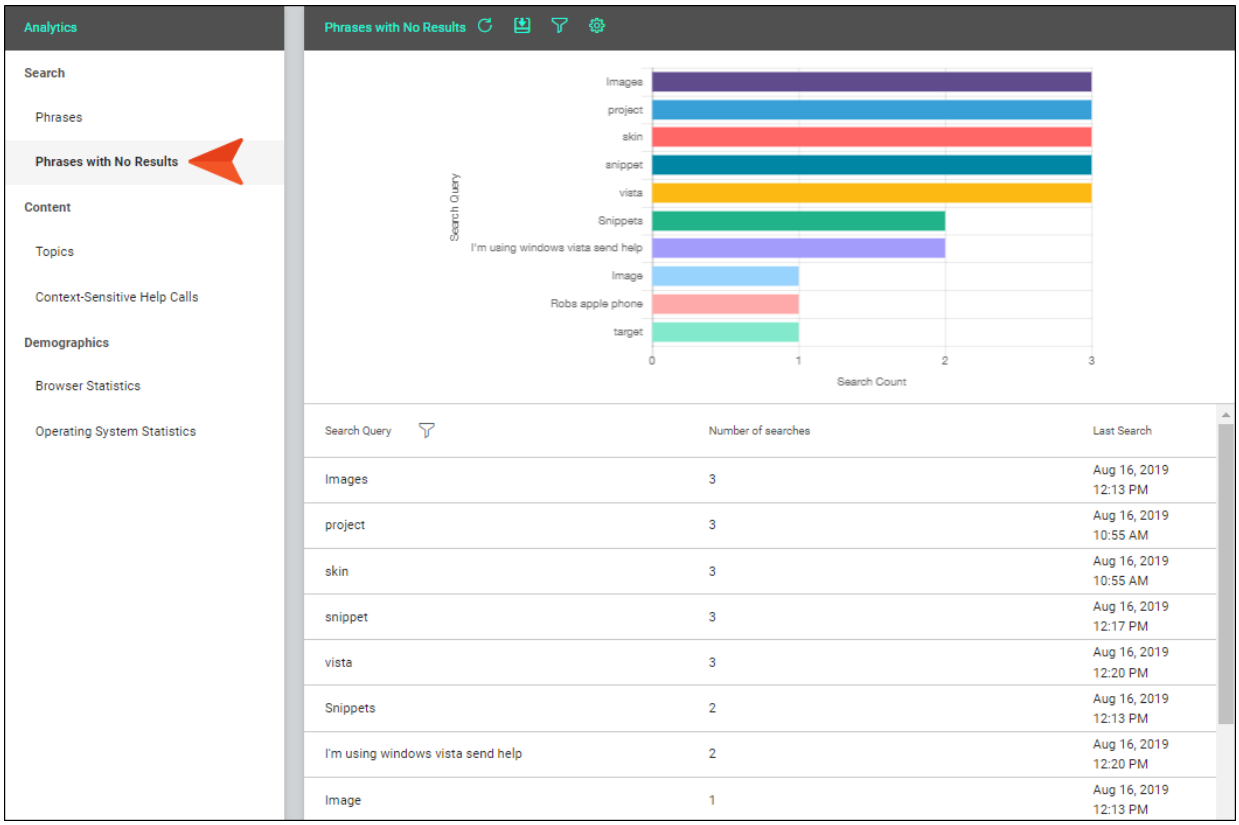
By using Central Analytics, you can obtain various types of information about your output.

Search

- **Phrases** Shows all search queries made to the output, including whether a search term returned any micro content results. *This is useful because it indicates the kind of documentation most often sought by users, as well as whether you should create new micro content to match certain search terms.*

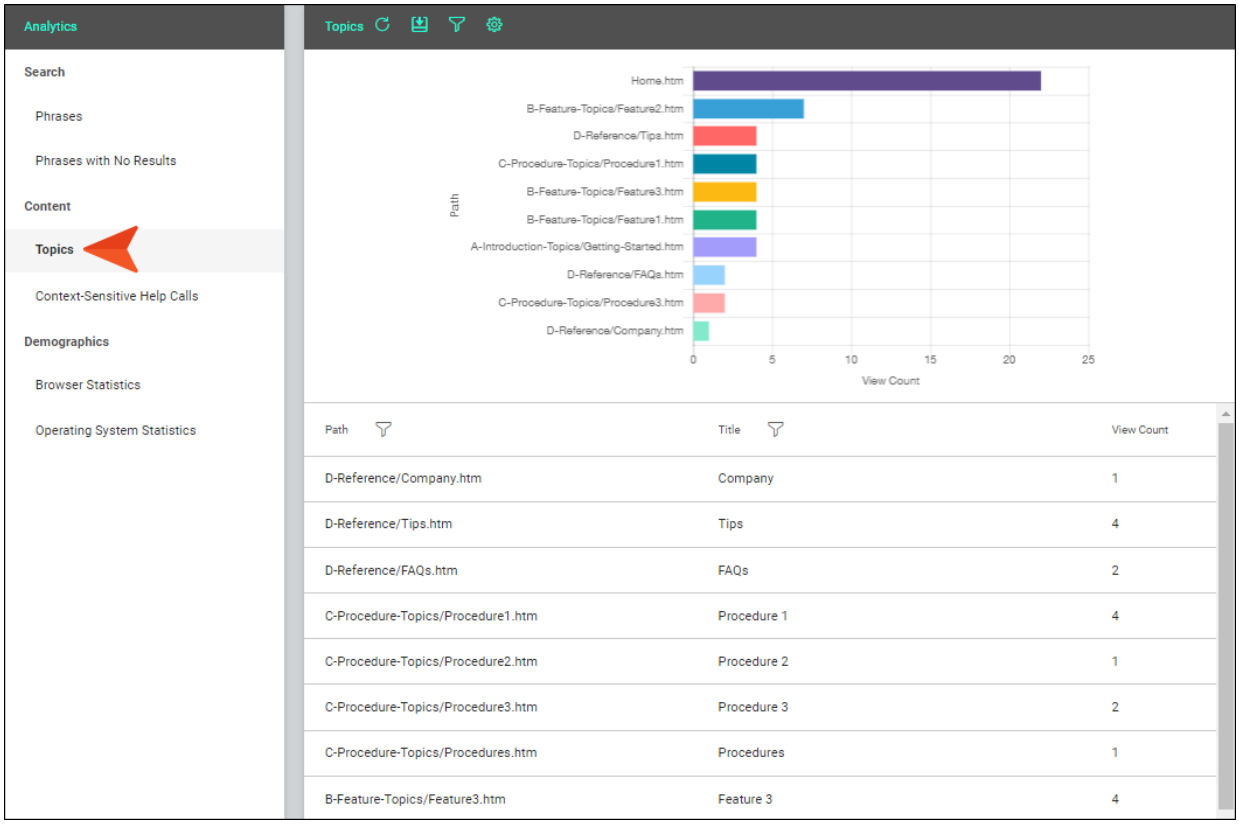


- **Phrases with No Results** Shows all search phrases that didn't produce any results. *This is useful because it indicates if you need to produce new documentation to provide results for users, or perhaps create synonyms that will show results for existing documentation.*

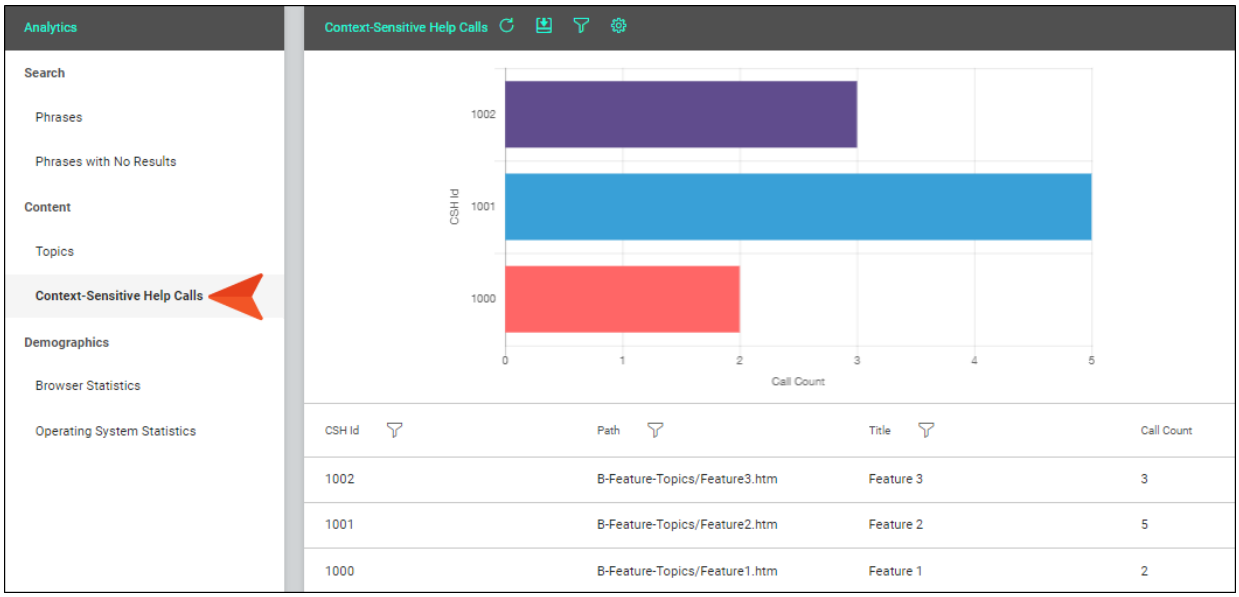


Content

- **Topics** Shows all topics in the output that have been viewed at least once. *This is useful because it indicates the topics where you might want to focus much of your attention.*

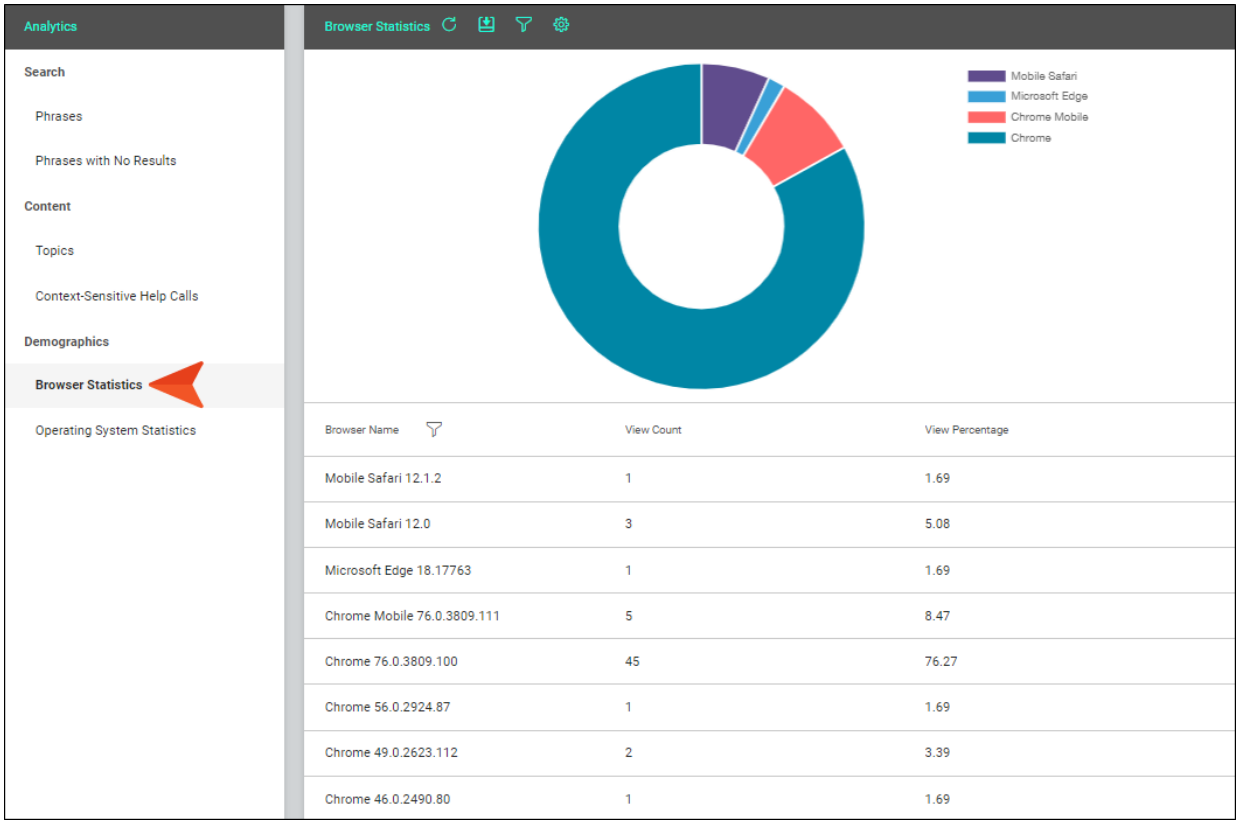


- **Context-Sensitive Help Calls** Shows context-sensitive Help identifiers that have been invoked at least once in the output. *This is useful because it indicates which areas of your product's interface are most active, and perhaps even unclear to users.*

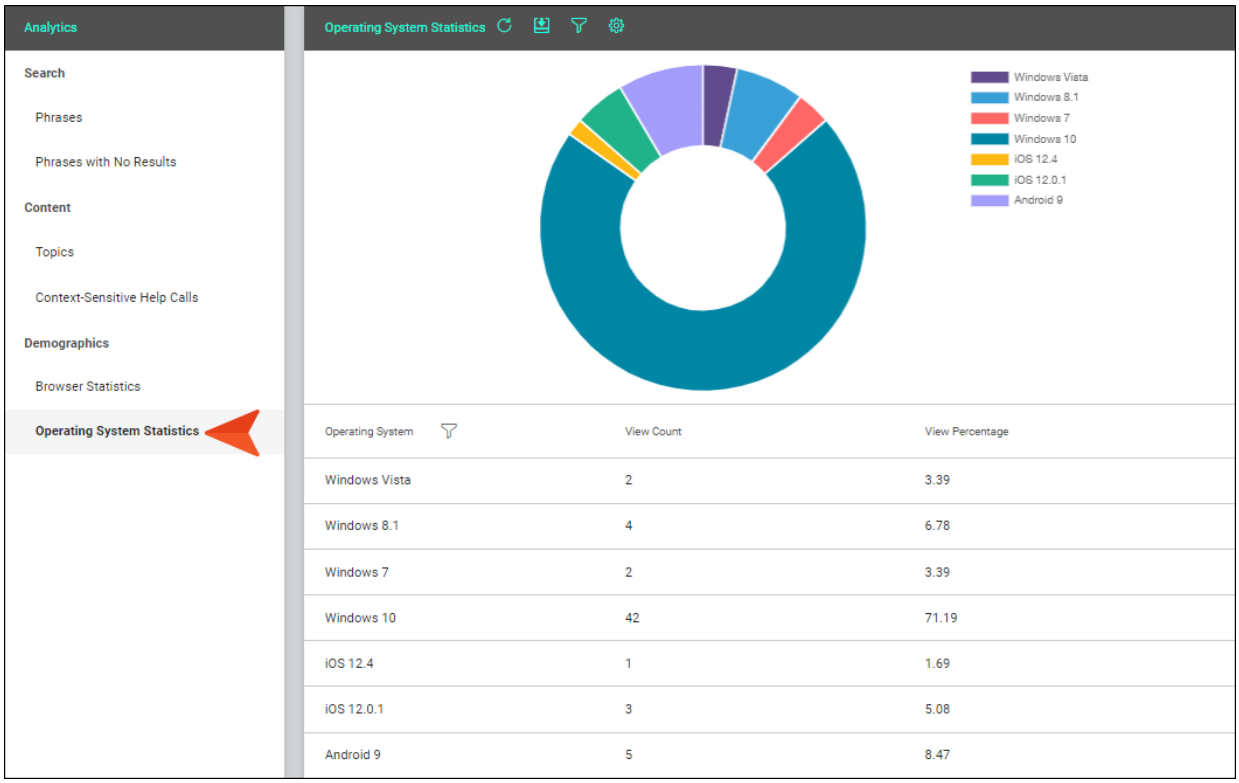


Demographics

- **Browser Statistics** Shows which browsers are being used to view your online output. The graph shows a broad view of the browsers, and the grid breaks down the information into specific browser versions. *This is useful because it indicates which browsers you should focus on supporting.*



- **Operating System Statistics** Shows which operating systems are being used to view your online output. *This is useful because it indicates which operating systems you should focus on supporting. It might even be used to determine whether your company's software needs to continue supporting certain operating systems.*



NOTE The graph at the top is limited to the first 10 items. The grid below shows all items.

Host Output Anywhere

Central analytics works on Flare projects uploaded to Central, or you can host output on your own servers. If you host the output outside of Central, you still need to use Central with a key to view the analytics data. Also, the server where the output is hosted must be able to communicate with Central (e.g., not be behind a firewall).

Creating Analytics Keys

Analytics keys are associated with Flare HTML5 targets. You can create and edit these keys either in Flare or on Central. To do this, you must have the permission on Central to "Manage Output Analytics Keys".

How to Create an Analytics Key in Central

1. On the left side of the interface, click Analytics.
2. In the toolbar click  .
3. Enter a name for the key and an optional description, then click Save.

Associating Analytics Keys With Targets

If you create a key in Flare, it will become associated with your target as you do this. But if you create the key in Central, you need to then open the Flare target and associate that key with the target.

How to Associate an Analytics Key With a Target

1. In Flare, open an HTML5 target.
2. Select the **Analytics** tab.
3. From the **Provider** field, select **Central**.
4. If you are not yet logged in to Central, click **Login**. Then complete the fields in the dialog and click **OK**.
5. From the **Central License** field, choose your license.
6. At the bottom of the **Key** area, click the drop-down and select the name of the key.
7. Click  to save your work.
8. Build output for the target on Central to see the analytics results.

Managing Analytics Keys

On the Analytics page, you can edit or delete keys.

How to Edit an Analytics Key

1. On the left side of the interface, click **Analytics**.
2. Click the check box next to the key you want to edit.
3. In the toolbar click .
4. Use the fields to edit the name and/or description, then click **Save**.

How to Delete Analytics Keys

1. On the left side of the interface, click **Analytics**.
2. In the grid, click the check box next to each key you want to delete. Or click the top check box to select all keys.
3. In the toolbar click .
4. In the dialog, type `delete` in the field under the listed keys.
5. Click **Delete**.

Downloading Analytics

In the toolbar of the Analytics page, you can click to download the data to a comma-separated value (CSV) file.

> **NOTE** If the project *is not bound to Central*, you can view the analytics information on any Central license where you are using the analytics key. However, if the project *is bound to Central*, you can view its analytics only on the Central license where the project is uploaded.

PDFs

The following PDFs are available for download from the online Help.

Getting Started Guide

Authoring Guide

License Management and Purchasing Guide

Projects and Builds Guide

Security Whitepaper

Sites Guide

Tasks Guide

Users and Teams Guide

What's New Guide

Widgets Guide