

File Entity Revisions Module 7.x.2.x

Kenneth Lancaster
drupal.org/u/kenneth-lancaster
kennethlancaster.com

Rick Tilley
drupal.org/u/pcrats33
pcrats.com

Co-maintainers wanted

Module History and Overview

The File Entity Revisions module is a Drupal module that allows for file entities to have the capability of revisions in the same manner that Drupal allows for revisions of content types. The 7.x.1.x branch was a great leap forward in helping to bring the capability of file versioning to Drupal.

Overview of the 7.x.1.x Branch

As described by Eric Duran, the author of the original 7.x.1.x branch, along with the revisions capability the module included support for:

- Revision Scheduler Module
- Revision Revert Operation
- Revision Delete Operation
- Revision Publish Operation (File Admin integration)
- Revision Unpublish Operation (File Admin integration)

We are not planning any future support for the 7.x.1.x branch. All support and features are built and focused towards the 7.x.2.x branch only.

Overview of the 7.x.2.x Branch

The File Entity Revisions 7.x.2.x module greatly expands on the original 7.x.1.x branch with a lot of custom rewriting, added sub-module, and many new features including:

- Moderation of files through integration of State Machine and Workflow modules.
- Better stream wrapper integration for public and private files.
- Only 1 revision at a time is allowed to be published residing in public:// or private:// or custom://
- All revisions of unpublished files reside in revisions:// and have special private access permissions
- File path set with taxonomy term attached to the file entity by the term reference field
- ~~Domain Access integration to build first folder in file structure based on domain name~~
 - Code commented out, can still be implemented.
- Persistent file name for the file's published version where the file name is set by the filename field
- Persistent file path, only changes if selected taxonomy and/or filename is changed in the file form field
- Revisions Workflow tab featuring file moderation history, comments, links to view previous versions of the file, revision operations, and domain assignment (if using Domain Access and Domain Access Entity) in an easy to understand graphic interface

Future goals of this branch are:

- Token Integration
- Thumbnails of files on the Revisions Workflow tab
- Workbench Integration (at least in some form, even if it's just visually)

Requirements

File Entity Revisions enables Drupal to store different versions of a file by utilizing these required modules:

- File Entity - makes File Types fieldable, allowing us to attach fields to files.
- File Admin - extends File Entity to add administrative options to file types.
- Workflow - defines the actual workflow which is attached to the file as a field.
- State Machine - used by Workflow to store the various moderation states and settings.
- Access Content Bridge - helper module for content access interplay.

If you are running Domain Access:



- Domain Access Entity - defines entity access per domains for file types.

Initial Module Setup

Follow your preferred method to install and enable the following modules.

File Entity and File Administration Modules

Enable the following modules for File Entity and File Admin:

Name	Description	Links
<div></div> File administration options (file_admin)	▶ Adds file administration options to publish, promote, and make st...	<div>Permissions</div>
<div></div> File entity (file_entity)	▶ Extends Drupal file entities to be fieldable and viewable.	<div>Permissions ▾</div>



File Entity and File Administration Modules

Enable the following modules for Access Control Bridge:

Name	Description	Links
<div><input checked="" type="checkbox"/></div> Access Control Bridge (acb)	▶ Bridge module linking Access Control modules so their access per...	





Workflow Modules

Enable the following modules for Workflow:

Name	Description	Links
<div><input checked="" type="checkbox"/></div> Workflow access (workflow_access)	▶ Content access control based on workflows and roles. Depends on the node_access system, so only works for entities of ...	
<div><input checked="" type="checkbox"/></div> Workflow Actions (workflow_vbo)	▶ Provides actions that can be associated to triggers, or used as VBO-action. Provided actions are 'set to next state' and 'set...	
<div></div> Workflow API (workflow)	▶ Workflow API. (Enable Workflow Node or Workflow Field to add arbitrary workflows to entities.)	<div>Help ▾</div>
<div><input checked="" type="checkbox"/></div> Workflow Clean Up (workflow_cleanup)	▶ Cleans up Workflow cruft.	<div>Configure</div>
<div></div> Workflow Field (workflowfield)	▶ Defines a Workflow field, widget and formatter. (Do not enable Workflow Field and Workflow Node together.)	
<div><input checked="" type="checkbox"/></div> Workflow Fields (workflow_fields)	▶ Adds per-field settings to each workflow state.	
<div><input checked="" type="checkbox"/></div> Workflow Node (workflownode)	▶ Adds a Workflow to nodes. (Use only if you come from a version below 7.x-2.x. If not, use 'Workflow Field' instead).	
<div><input checked="" type="checkbox"/></div> Workflow Node Search API (workflow_search_api)	▶ Adds Workflow Node's state information to Search API index.	
<div><input checked="" type="checkbox"/></div> Workflow Revert (workflow_revert)	▶ Adds a 'Revert' link to the first workflow history row.	<div>Permissions</div>
<div><input checked="" type="checkbox"/></div> Workflow Rules (workflow_rules)	▶ Provides additional, workflow-specific Rules integration.	<div>Configure</div>
<div><input checked="" type="checkbox"/></div> Workflow Trigger (workflow_actions)	▶ Enables actions to be fired upon a Workflow State change.	
<div><input checked="" type="checkbox"/></div> Workflow UI (workflow_admin_ui)	▶ Provides administrative UI for workflow.	<div>Permissions ▾</div>
<div><input checked="" type="checkbox"/></div> Workflow views (workflow_views)	▶ Provides views integration for workflows.	<div>Permissions</div>

State Machine Modules

Enable the following modules for State Machine:

Name	Description	Links
<input checked="" type="checkbox"/>  State Flow (state_flow)	▶ A base implementation of the State Machine	Permissions
 State Flow Entity (state_flow_entity)	▶ The base for an entity workflow	
 State Flow Schedule (state_flow_schedule)	▶ Scheduling of publishing, unpublishing with State Flow	
 State Machine (state_machine)	▶ Provides a Finite State Machine for Drupal entities.	

File Entity Revisions Modules

Enable the following modules for File Entity Revisions:


Name	Description	Links
<input checked="" type="checkbox"/> File entity Revisions (file_entity_revisions)	▶ Extends file entity to be revision-able .	Permissions

File Entity Revisions also includes the sub-module Uploading Taxonomy Paths:

Name	Description	Links
<input checked="" type="checkbox"/> Uploading Taxonomy Paths (uploading_tax_paths)	▶ Custom paths for uploading documents - based on a taxonomy	

Domain Access Entity

This is required ONLY if you are using Domain Access:

Name	Description	Links
 Domain Access Entity (domain_entity)	▶ Defines Domain access for entities	Permissions

Map Out Your Workflow

File Entity Revisions will allow you to apply a Workflow to a File Type. Before you start creating workflows in Drupal and applying them to files, you will want to think about how you want workflow to handle files, how many states will be in the transition process, and if you want to create and use different file types to have their own workflow. Depending on your needs, you may want to create different file types so that you can have file type specific fields.

Once data has been saved for a specific workflow, you cannot add or remove the available states in that workflow. For this reason, it is very important that you map out the workflow states before applying the workflow to any uploaded files. Your various states with the workflow will also define when an item is accessible to the public.

Example States Labels and Resulting Actions:	<u>Not Accessible to the Public</u> Creation Draft Review Final Review 508 Check	<u>Accessible to the Public</u> Published
--	---	--

Most people think of accessible to the public in terms of the default “Published” and “Unpublished” used for making nodes live on a website. With moderation states you can use any term as the label to “publish” an item. For this demonstration we will create a File Type and Workflow for Permits. You can create as many moderation states as you feel necessary, however for this demonstration we will only use a few moderation states.

Permit State Labels and Resulting Actions:	<u>Not Accessible to the Public</u> Creation Permit Draft	<u>Accessible to the Public</u> Permit Published
--	---	---

Putting the Pieces Together

The following sections run through the whole process of creating a workflow for file entity moderation with true file revisions. Please keep in mind this set-up is utilizing several different modules and there are always variations in a set-up. The modules and their configurations below are what was needed to build file moderation workflow with the File Entity Revisions module. If there are issues or suggested enhancements with modules outside of File Entity Revisions module they would need to be addressed with those modules.

Create the Workflow

Once you mapped out your workflow, go to Administration > Configuration > Workflow and add a Workflow:

Home » Administration » Configuration » Workflow

Workflows

This page allows you to maintain Workflows. Once a workflow is created, you can maintain your entity type and add a Field of type 'Workflow'.
Powered by [Drupal](#)

[+ Add workflow](#) [+ Import workflow](#) [+ Access settings](#) [+ Clean up workflow](#)

Label	Status	Operations
None.		


[Save](#)

You can create a general workflow to handle common file types, or you can create specific workflows to be used only for a specific file type. For a better demonstration, we will title the workflow to be used for specific file type. On the Add Workflow page, simply type in the name for your workflow and hit save:

Home » Administration » Configuration » Workflow » Workflows

Edit permit document workflow

[Edit](#) [States](#) [Transitions](#) [Labels](#) [Permission summary](#)

 The workflow *permit_document_workflow* was created. Please maintain the states and transitions.

Powered by [Drupal](#)

Label *

Machine name: `permit_document_workflow` [\[Edit\]](#)




The human-readable name of this content type.

[Save](#)

Now you will want to create the various states within the workflow. Creation is the initial state which occurs when you first create an entity and is a default state. The entity remains in this state until it is initially saved. Note: once you have saved any entity with this workflow, you cannot add or delete any of these states with drastic consequences. Plan your workflow first!

permit document workflow

[Edit](#) [States](#) [Transitions](#) [Labels](#) [Permission summary](#)Powered by [Drupal](#)


State	Active	Reassign	Count	Operations
 (creation)	Machine name: (creation) 		0	Actions
 Permit Draft	Machine name: permit_draft [Edit] 	<input type="text"/>	0	Actions
 Permit Review	Machine name: permit_review [Edit] 	<input type="text"/>	0	Actions
 Permit Published	Machine name: permit_published [Edit] 	<input type="text"/>	0	Actions
 <input type="text"/>			0	Actions

[Save Changes](#)

The Transitions tab is actually where you set the permissions for users moving content along the workflow path. Depending on the user role and their permission settings here, an entity can be set to any of a number of states upon save.

permit document workflow

Edit States **Transitions** Labels Permission summary

 The workflow was updated.

You are currently viewing the possible transitions to and from workflow states. The state is shown in the left column; the state to be moved to is to the right. For each transition, check the box next to the role(s) that may initiate the transition. For example, if only the "production editor" role may move a node from Review state to the Published state, check the box next to "production editor". The author role is built in and refers to the user who authored the node.

If not all roles are in the list, please review which roles may 'participate in workflows' on the [Permissions page](#).

Powered by [Drupal](#)

From / To →	Permit Draft	Permit Review	Permit Published
(creation)	<input checked="" type="checkbox"/> (author) <input checked="" type="checkbox"/> Contributor <input checked="" type="checkbox"/> Moderator <input checked="" type="checkbox"/> Publisher <input checked="" type="checkbox"/> Site Architect	<input checked="" type="checkbox"/> (author) <input checked="" type="checkbox"/> Contributor <input checked="" type="checkbox"/> Moderator <input checked="" type="checkbox"/> Publisher <input checked="" type="checkbox"/> Site Architect	<input type="checkbox"/> (author) <input type="checkbox"/> Contributor <input type="checkbox"/> Moderator <input checked="" type="checkbox"/> Publisher <input checked="" type="checkbox"/> Site Architect
Permit Draft		<input checked="" type="checkbox"/> (author) <input checked="" type="checkbox"/> Contributor <input checked="" type="checkbox"/> Moderator <input checked="" type="checkbox"/> Publisher <input checked="" type="checkbox"/> Site Architect	<input type="checkbox"/> (author) <input type="checkbox"/> Contributor <input type="checkbox"/> Moderator <input checked="" type="checkbox"/> Publisher <input checked="" type="checkbox"/> Site Architect
Permit Review	<input checked="" type="checkbox"/> (author) <input checked="" type="checkbox"/> Contributor <input checked="" type="checkbox"/> Moderator <input checked="" type="checkbox"/> Publisher <input checked="" type="checkbox"/> Site Architect		<input type="checkbox"/> (author) <input type="checkbox"/> Contributor <input type="checkbox"/> Moderator <input checked="" type="checkbox"/> Publisher <input checked="" type="checkbox"/> Site Architect
Permit Published	<input type="checkbox"/> (author) <input type="checkbox"/> Contributor <input checked="" type="checkbox"/> Moderator <input checked="" type="checkbox"/> Publisher <input checked="" type="checkbox"/> Site Architect	<input type="checkbox"/> (author) <input type="checkbox"/> Contributor <input checked="" type="checkbox"/> Moderator <input checked="" type="checkbox"/> Publisher <input checked="" type="checkbox"/> Site Architect	

Save

Once you have set the Transitions you can change the Transition labels and review the Permission summary on the other tabs. Reviewing those tabs is not necessary at this point for the demonstration and we can now move on to creating a file type for which to apply the newly created workflow.

Create the File Type

Once you created your workflow you can create your file type. Your file type will require you to add a Workflow field and a Term Reference field. If you are also running Domain Access you will need to configure the settings for the Domain field which is actually set elsewhere (discussed further below in Domain Access Entity Configuration for Domain Access). Go to Admin>Structure>File-Types add a new file type.

+ Add file type

Name	Operations					Status
Image <i>An Image file is a still visual.</i>	edit file type	manage fields	manage display	manage file display	disable	Default
Video <i>A Video file is a moving visual recording.</i>	edit file type	manage fields	manage display	manage file display	disable	Default
Audio <i>An Audio file is a sound recording.</i>	edit file type	manage fields	manage display	manage file display	disable	Default
Document <i>A Document file is written information.</i>	edit file type	manage fields	manage display	manage file display	disable	Default

For this demonstration we will create a File Type for Permits. This file type will only utilize Adobe pdf and Word document mimetypes. For the publishing options “Published” should stay unchecked and “Create new revision” should be checked. However, keep in mind that once this file type is created, the File Entity Revisions module takes over the publishing and whether this box is checked or not will have no effect.

For our purposes we are leaving the display owner and date information unchecked as well. However, you can set it to suit your requirements. You will also want to adjust the display options for your own requirements as well.

Name *

Permit Documents

Machine name: permit_documents [\[Edit\]](#)

This is the human readable name of the file type.

Description

These files are for various Permit document files. Once you have uploaded your file more field options and meta information will be available.

This is the description of the file type.

Mimetypes

application/msword
application/pdf

Enter one mimetype per line.

▸ Mimetype List

Submission form settings

Publishing options

Display settings

Publishing options

Default options

☐ Published

☐ Promoted

☐ Sticky at top of lists

☒ Create new revision

Users with the *Administer files* permission will be able to override these options.

Save

Now that the new document type has been created, we will add the Workflow field to it in order to attach the Workflow moderation. Note, that once the File Entity Revisions module is enabled and set for a file type by selecting Create new revision here, the Published checkbox option above is then controlled behind the scenes by the module and it makes no difference if the option here for Published is selected or not.

Permit Documents

[Edit](#)
[Manage fields](#)
[Manage dependencies](#)
[Manage display](#)
[Manage file display](#)

Powered by [Drupal](#)

[Show row weights](#)

Label	Machine name	Field type	Widget	Operations
+ File name	filename	File name		
+ File	preview	File preview		
+ Add new field				
<input type="text" value="it Document Workflow"/> Label	field_permit_document_workflow [Edit]	<input type="text" value="Workflow"/> Type of data to store.	<input type="text" value="Workflow"/> Form element to edit the data.	

Once you hit save and the field is added we will adjust the settings according to the following screenshot. Be sure to select the appropriate Workflow that you just set up. For how to display the states, choose "Select list" and then be sure to select the box for "Hide the widget on Entity form" and do not select the "Workflow history" checkbox.

We are hiding the workflow widget on the entity form and we are not using the Workflow history tab on the file because we have created a customized Revision Workflow tab for displaying both the both the file history and controlling all moderation states. So enabling these other options may confuse the UX/UI for the end user when dealing with file types.

You can of course change this to suit your needs as you may want to have the Workflow State widget available on the file create/edit pages to make publishing new revisions a one step process. However, the History tab is not necessary as all of the information that is in the History tab is also found on the Revisions Workflow tab.

Permit Document Workflow

Edit Field settings Widget type Conditional states Delete

Powered by [Drupal](#)

Field settings

These settings apply to the *Permit Document Workflow* field everywhere it is used. These settings impact the way that data is stored in the database and cannot be changed once data has been created.

Workflow type *

permit document workflow ▼

Choose the Workflow type. Maintain workflows [here](#).

Workflow widget

Set some global properties of the widgets for this workflow. Some can be altered per widget instance.

How to show the available states

Select list ▼

The Widget shows all available states. Decide which is the best way to show them. ('Action buttons' do not work on Comment form.)

☒ Hide the widget on Entity form.

Using Workflow Field, the widget is always shown when editing an Entity. Set this checkbox in case you only want to change the status on the Workflow History tab or on the Node View. (This checkbox is only needed because Drupal core does not have a widget.)

☐ Use the workflow name as the title of the workflow form

The workflow section of the editing form is in its own fieldset. Checking the box will add the workflow name as the title of workflow section of the editing form.

☐ Allow scheduling of workflow transitions.

Workflow transitions may be scheduled to a moment in the future. Soon after the desired moment, the transition is executed by Cron. This may be hidden by settings in widgets, formatters or permissions.

☐ Show a timezone when scheduling a transition.

Allow adding a comment to workflow transitions

optional ▼

On the Workflow form, a Comment form can be included so that the person making the state change can record reasons for doing so. The comment is then included in the node's workflow history. This may be altered by settings in widgets, formatters or permissions.

☒ Log informational watchdog messages when a transition is executed (a state value is changed)

Optionally log transition state changes to watchdog.

Workflow history

☐ Use the workflow history, and show it on a separate tab.

Every state change is recorded in table (workflow_node_history). If checked and user has proper permission, a tab 'Workflow' is shown on the entity view page, which gives access to the History of the workflow. If you have multiple workflows per bundle, better disable this feature, and use, clone & adapt the Views display 'Workflow history per Entity'.

Now you will need to add a taxonomy to the file type using Term Reference field. Taxonomy is a common feature to use in Drupal and how you structure your taxonomy is always different from website to website, so I will not run through the taxonomy creation process. However, you need to understand that the taxonomy which you attach to a file type is used to help place the file in a folder and thus build the file URL structure.

You can attach as many term reference fields to a file type to fit your site needs. However, only 1 term reference field can be used per file type by the File Entity Revisions module to build file structure. You can title your Term Reference field whatever you wish. For this demonstration we are breaking up our files by division terms.

Add new field

ocument Division

Label

field_permit_document_division [\[Edit\]](#)

Term reference ▼

Type of data to store.

Select list ▼

Form element to edit the data.

Once you have saved the field, and Drupal takes you to the screen for more defined field settings, you will want to scroll down to the checkbox for “Use field’s value to determine upload sub-path. Once you select that box, another form field option will open up where you can set the path using %1 for the taxonomy term value. Do not include a trailing slash. We are also hoping to integrate token support here in the future.

☒ Use field's value to determine upload sub-path.

The taxonomy name will be converted to a path name that can be used for upload placement whenever a new file is uploaded.

— Uploading Taxonomy Paths Settings —

Path to store uploaded files

permits/%1

Enter a path, use %1 to represent the taxonomy value

Also, for the taxonomy reference field be sure to set the option for “Number of values” to only 1 value. In testing, allowing more than one value still worked, but the first value in the list is what will get used. And it may not always be the same term that shows up first in the array, so this will cause your file URL’s to keep changing confusing the user of what to expect. So only one term value should be used.

— Permit Document Division field settings —

These settings apply to the *Permit Document Division* field everywhere it is used. Because the field already has data, some settings can no longer be changed.

Number of values

1

Maximum number of values users can enter for this field.

Vocabulary *

Divisions

The vocabulary which supplies the options for this field.

Configuring Uploading Taxonomy Paths and File Entity Revisions

At this point our file type now has the necessary taxonomy attached to it for file placement and URL structure and these settings can also be seen on the Uploading Tax Paths configuration overview screen.

Home » Administration » Configuration » Workflow

Uploading Tax Paths

Powered by Drupal

— File types and their configurations. —

Permit Documents (permit_documents) field_permit_document_division permits/%1

In order to configure a default path, you will need to edit a taxonomy field attached to a file type and check the box that says: Use field's value to determine upload sub-path.

Also, at this point our file type now has a workflow attached to it enabling file moderation. We now need to go to the File Entity Revisions configuration page to select the various states that are for publishing and un-publishing content.

Be sure to match the correct field to the file type, which is the reason I used a specific naming structure in this demonstration.

Also, for this demonstration, I used a limited amount of transition states to keep it simple. Your needs may be very different for your workflow, but the same rules here apply. You will see your file type machine name listed and under that is where you will select the workflow field.

For the “Publish fallback action”, you want to select the transition state that will actually trigger the file to being published. Note: only one 1 revision can be published at a time.

For the “Unpublish fallback action”, you will want to select the transition state that a file revision will be placed in upon another revision getting published.


So, when any change is made to which version of a file is published, the Unpublish fallback action kicks in for the previously published version. In our demonstration we chose Permit Draft as the fallback.

In an example of how this might work, a current revision of a file may be deemed to need an edit, and a previous version, may need to go live to replace it (revert the file) while changes are being made. In this scenario, when a file has a currently published version, and another older or newly uploaded version is then set to Publish, what was the currently published version will then get placed into the state as defined by the Unpublish fallback action.

The option for “Check all states that are valid published (or unpublished) states” is pretty self-explanatory. However, I should note that if you create multiple workflows and use the same name for various states (or if you clone a workflow), then you will see all multiple states with the same name. This can get very confusing for deciding which fallback action to choose unless you are able to match up the state id number.

Home » Administration » Configuration » Workflow

File Entity Revisions

 This form is domain-sensitive, be sure you select the proper domain before saving.

Powered by [Drupal](#)

File Entity Revisions needs to understand how Workflow is set up.

permit_documents

Permit Document Workflow [field_permit_document_workflow] ▼

Select the active workflow field.

Publish fallback action

(4) Permit Published ▼

Select the default published state.

Unpublish fallback action

(2) Permit Draft ▼

Select the default unpublished state.

Check all states that are valid published states

- ☐ (1) (creation)
- ☐ (2) Permit Draft
- ☐ (3) Permit Review
- ☒ (4) Permit Published

Check all states that are valid unpublished states

- ☒ (1) (creation)
- ☒ (2) Permit Draft
- ☒ (3) Permit Review
- ☐ (4) Permit Published

Now that all the right components are in place and various settings have been configured. The last step we need to do is configure Access and the Triggers and Actions for the various workflow transitions. You will find that using a specific naming structure for multiple file types and workflows will greatly help when configuring these settings. To get to the page you can go to any of your created workflows and click on the “Actions” link in the far right side.

Home » Administration » Configuration » Workflow

Workflows

This page allows you to maintain Workflows. Once a workflow is created, you can maintain your entity type and add a Field of type 'Workflow'.
Powered by [Drupal](#)

[+ Add workflow](#) [+ Import workflow](#) [+ Access settings](#) [+ Clean up workflow](#)

Label	Status	Operations
permit document workflow (Machine name: permit_document_workflow)	Default	edit clone export access actions

This will open up the various options on this page which are fairly self-explanatory. You are able to set Actions for the Trigger of a state transition. The more transitions you have, the more settings you will have to configure.

Home » Administration » Structure » Triggers

Triggers

[Comment](#) [Node](#) [System](#) [Taxonomy](#) [User](#) [Workflow API](#)

Action *Unpublish File Entity* has been unassigned.

Triggers are events on your site, such as new content being added or a user logging in. The Trigger module associates these triggers with actions (functional tasks), such as unpublishing content containing certain keywords or e-mailing an administrator. The [Actions settings page](#) contains a list of existing actions and provides the ability to create and configure advanced actions (actions requiring configuration, such as an e-mail address or a list of banned words).

There is a tab on this page for each module that defines triggers. On this tab you can assign actions to run when triggers from the [Workflow API module](#) happen.

Powered by [Drupal](#)

Trigger: When file *permit_documents* moves *permit document workflow* from (creation) to *Permit Draft*

Choose an action

Assign

Trigger: When file *permit_documents* moves *permit document workflow* from (creation) to *Permit Review*

Choose an action

Assign

For any resulting state change where the file moves from a published state to an unpublished state, choose the action to Unpublish the File Entity, and obviously for the state change that should trigger publishing the file, for example from review to publish, choose the action for Publish File Entity as seen below. After you select the appropriate action, click Assign.

Trigger: When file *permit_documents* moves *permit document workflow* from *Permit Draft* to *Permit Published*

Publish File Entity

Assign

Once you have assigned an action it should appear like the screenshot below.

Trigger: When file *permit_documents* moves *permit document workflow* from *Permit Draft* to *Permit Published*

Name	Operation
Publish File Entity	unassign

Choose an action

If a state change results in no actual change in public accessibility to the file, for example, moving from Creation to Draft, Draft to Draft, Draft to Review, or Publish to Publish, do not apply an action. The configuration should look like the screenshot below.

Trigger: When file *permit_documents* moves *permit document workflow* from *(creation)* to *Permit Review*

Choose an action

NOTE: There is one exception to the above statement. For any state change of published to published, you will want to add a publish action as seen below. This will eliminate a couple annoying bugs if you are actually using the Workflow widget in the Create/New Draft form.

Trigger: When file *permit_documents* moves *permit document workflow* from *Permit Published* to *Permit Published*

Name	Operation
Publish File Entity	unassign

Choose an action

With Actions now defined for the file type, return to the page for your workflow and follow the link for Access.

Home » Administration » Configuration » Workflow

Workflows

This page allows you to maintain Workflows. Once a workflow is created, you can maintain your entity type and add a Field of type 'Workflow'.
 Powered by [Drupal](#)

[+ Add workflow](#) [+ Import workflow](#) [+ Access settings](#) [+ Clean up workflow](#)

Label	Status	Operations				
permit document workflow (Machine name: permit_document_workflow)	Default	edit	clone	export	access	actions

Once you select the "access" link, on the following page you will want to select who has viewing, editing, and deleting rights to the file entities while in the different transition states. The options are fairly self-explanatory and how you select these options of course depends on your workflow.

permit document workflow Access

WARNING: Use of the 'Edit any', 'Edit own', and even 'View published content' permissions for the content type may override these access settings. You may need to [alter the priority of the Workflow access module](#).

Powered by [Drupal](#)

Permit Draft

Roles who can view posts in this state

- ☒ (author)
- ☒ Contributor
- ☒ Moderator
- ☒ Publisher
- ☒ Site Architect

Roles who can edit posts in this state

- ☒ (author)
- ☒ Contributor
- ☒ Moderator
- ☒ Publisher
- ☒ Site Architect

Roles who can delete posts in this state

- ☒ (author)
- ☒ Contributor
- ☒ Moderator
- ☒ Publisher
- ☒ Site Architect

Permit Review

Roles who can view posts in this state

- ☒ (author)
- ☒ Contributor
- ☒ Moderator
- ☒ Publisher
- ☒ Site Architect

Roles who can edit posts in this state

- ☐ (author)
- ☐ Contributor
- ☒ Moderator
- ☒ Publisher
- ☒ Site Architect

Roles who can delete posts in this state

- ☐ (author)
- ☐ Contributor
- ☒ Moderator
- ☒ Publisher
- ☒ Site Architect

Configuring the Drupal Permissions

Also be sure to visit the default Drupal permissions page for roles and adjust permissions there for the file type. You will also want to ensure the Access Content Bridge module is installed and enabled to help with any possible access conflicts. Keep in mind that you will need to adjust and configure the permissions for your new custom file type, every time that you create a new file type. On the Permissions page look for the File entity category.

Permission	anonymous user	authenticated user	Contributor	Moderator	Publisher	Site Architect
Permit Documents: Edit own files	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Permit Documents: Edit any files	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Permit Documents: Delete own files	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Permit Documents: Delete any files	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Permit Documents: Download own files	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Permit Documents: Download any files	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

While we are on the topic of the Drupal permissions page, here's a quick overview of some of the other various permissions you will/may need to adjust. Obviously, depending on your set-up, this will most likely not be an exhaustive list of the permissions you will need to adjust. However, these are the likely permissions you will need to look at affecting any file uploading, editing, moderation, viewing, and other similar actions.

Permission	anonymous user	authenticated user	Contributor	Moderator	Publisher	Site Architect
File administration options						
View any unpublished file details For viewing file details, not for downloading files.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
View own unpublished file details For viewing file details, not for downloading files.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
File entity						
Bypass file access control View, edit and delete all files regardless of permission restrictions. <i>Warning: Give to trusted roles only; this permission has security implications.</i>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Administer file types <i>Warning: Give to trusted roles only; this permission has security implications.</i>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Administer files <i>Warning: Give to trusted roles only; this permission has security implications.</i>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Add and upload new files	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
View own private files Includes the following stream wrappers: <i>Private files</i> .	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
View own files	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
View private files <i>Warning: Give to trusted roles only; this permission has security implications.</i>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
View files Includes the following stream wrappers: <i>Public files, File revisions that aren't currently published.</i>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

* Notice that in the View files permission, it says, "Includes the following stream wrappers: *Public files, File revisions that aren't currently published.*" Just ignore this, the new branch works with the permissions differently, and this View files permission only applies to the Public stream wrappers and not to File Revision stream wrappers.

File entity Revisions						
Administer File Entity Revisions Configure File Entity Revisions to work with Workflow.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
View Unpublished File Revisions Moderators will need to be able to view unpublished file revisions. Currently viewing private files only works on the current revision, this allows them to view other revisions as well.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
State Flow						
Manage content workflow Manage the content workflow pages and operations.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Administer content revisions	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Workflow API						
Schedule workflow transitions Schedule workflow transitions.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Show workflow state change on node view Show workflow state change form on node viewing.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Participate in workflows Role is enabled for transitions on the workflow admin pages.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Edit comment in workflow transitions Edit comment of Logged transitions via a Views link.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Workflow Revert						
Revert workflow Allow user to revert workflow state.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Workflow UI						
Administer workflow Administer workflow configurations.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Workflow views						
Access workflow summary views Access workflow summary views.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Under File Entity Revisions permissions be sure to only assign this permission Administer File Entity Revisions to roles you want to have access to changing configuration settings.

The issue of permission is brought up because often during our testing, issues we ran into had more to do with a simple permission setting and this gets even more complicated when using Domain Access. So you may need to do quite a bit of configuration testing to match your workflow design.

Configuring File System Paths

You will also need to visit the Drupal File System paths page. On this page you will now also see a new setting for Revisions file system path. This is where all the file revisions are stored. You can of course change your file structure to suit your needs and follow best practices for security. Here we simplified this for easier demonstration.

Home » Administration » Configuration » Media

File system

Powered by [Drupal](#)

Public file system path

A local file system path where public files will be stored. This directory must exist and be writable by Drupal. This directory must be relative to the Drupal installation directory and be accessible over the web.

Private file system path

An existing local file system path for storing private files. It should be writable by Drupal and not accessible over the web. See the online handbook for [more information about securing private files](#).

Temporary directory

A local file system path where temporary files will be stored. This directory should not be accessible over the web.

Default download method

☒ Public local files served by the webserver.

☐ Private local files served by Drupal.

This setting is used as the preferred download method. The use of public files is more efficient, but does not provide any access control.

Revisions file system path

An existing local file system path for storing file revisions. It should be writable by Drupal and not accessible over the web.

Save configuration

Configuring Domain Access Entity (only) when using Domain Access

If you are using Domain Access read this section too, otherwise you can skip to the Upload and Add a New File section. Domain Access is not required and Domain Access Entity is only required if using Domain Access.

In Domain Access settings you will need to configure the Entities tab for files. Once configured this settings will automatically place the necessary domain access field onto the file type. If you do not properly configure this while running Domain Access you will get errors when uploading files.

NOTE: There is a bug with Domain Access Entity. Selecting “Send to affiliates” does not actually select all the domains for the file. You will need to individually select each domain for the file type.

Domains

Domain list Settings Batch updating Content defaults **Entities** User defaults

Powered by [Drupal](#)

Choose which entities are under Domain Access control, and choose domain entity widget behavior of bundles

✓ **File (file)**

☒ un-check for disabling domain access on File (file), all entities domain assignment values will be deleted.

▸ **Permit Documents**

Choose which behavior must be used with the bundle Permit Documents ('permit_documents')

User choose affiliate, with a default value (form widget on the entity creation form) ▼

default domain value(s) for the bundle: Permit Documents ('permit_documents') *

☐ Current domain

☐ Send to all affiliates

☒ dep-internet

▸ **Image**

▸ **Video**

▸ **Audio**

▸ **Document**

☐ Activate domain access on Taxonomy term (taxonomy_term)

☐ Activate domain access on Field collection item (field_collection_item)

☐ Activate domain access on Comment (comment)

Save

This module is supposed to help in determining the access to an entity based on the domain. However, at this time I want to emphasize, that if you are running Domain Access, any public file in the system is accessible through all domains and this has more to do with behind the scenes access. This is explained better below.

This works with the users' access by assigning users to a domain. So if a user is assigned to a specific domain, when that user is on a file list page, they will be able to see the files to which they have access to as determined by the domains assigned to the user and the file entity. **From our testing, this applies to the superadmins too!**

Also as mentioned, you should realize that public files are accessible by any of the domains. However, you will only be able to access the page housing the file, from the actual domain it is assigned, and to make it more confusing you can edit from any domain the user is assigned to. Other permissions apply too, but here's a basic breakdown:

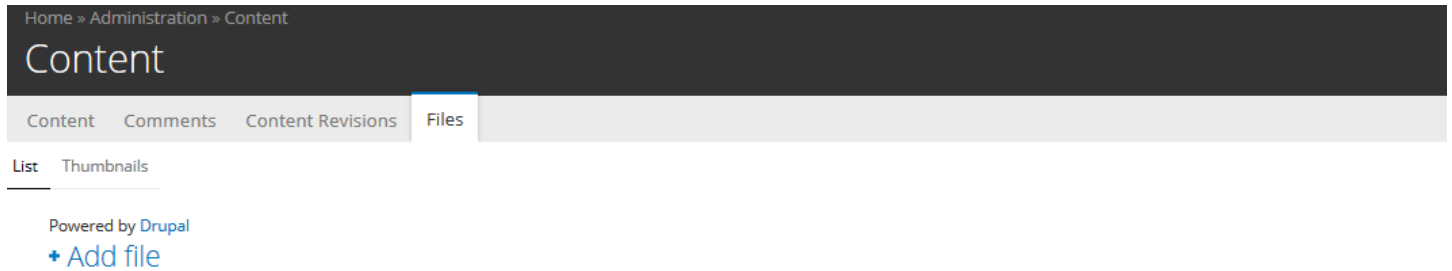
User has access to domain-X and domain-Y, but no access to domain-Z. File pdf is assigned to domain-Y and domain-Z.

Condition for user	Condition for file assigned to domain-Y and domain-Z	Domain-X	Domain-Y	Domain-Z
Public and User X and Y	Once published, is file accessible through this domain path?	yes	yes	yes
Public and User X and Y	Once published, does front-end file page exist/accessible?	no	yes	yes
User X and Y	Published or not, does file appear in the admin file list?	yes	yes	No Access
User X and Y	Published or not, can the user access file edit page?	yes	yes	No Access

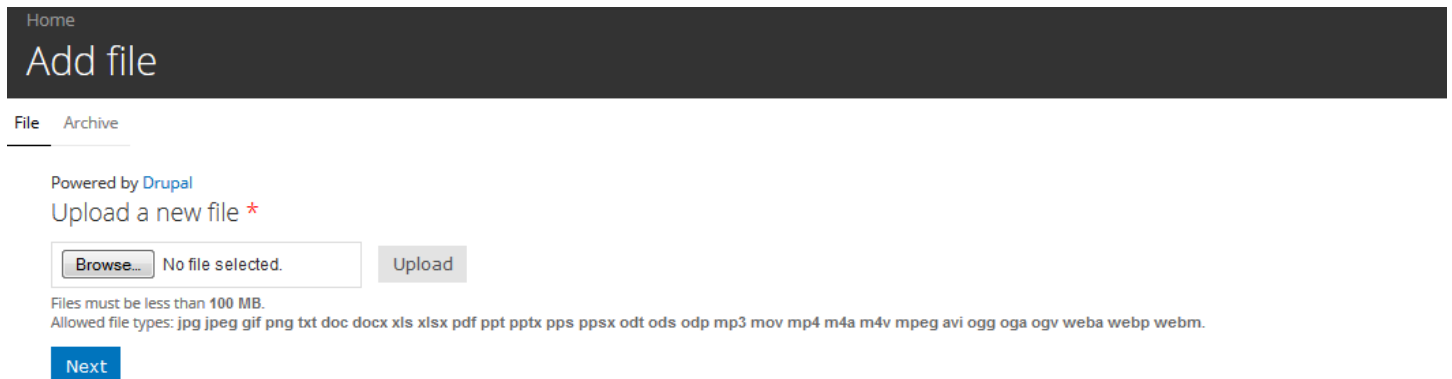
All of this has to do with the Domain Access Entity module and not File Entity Revisions. So any changes or enhancements would need to come from that module. However, it is important to understand how it interacts in this whole setup when running Domain Access.

Upload and Add a File

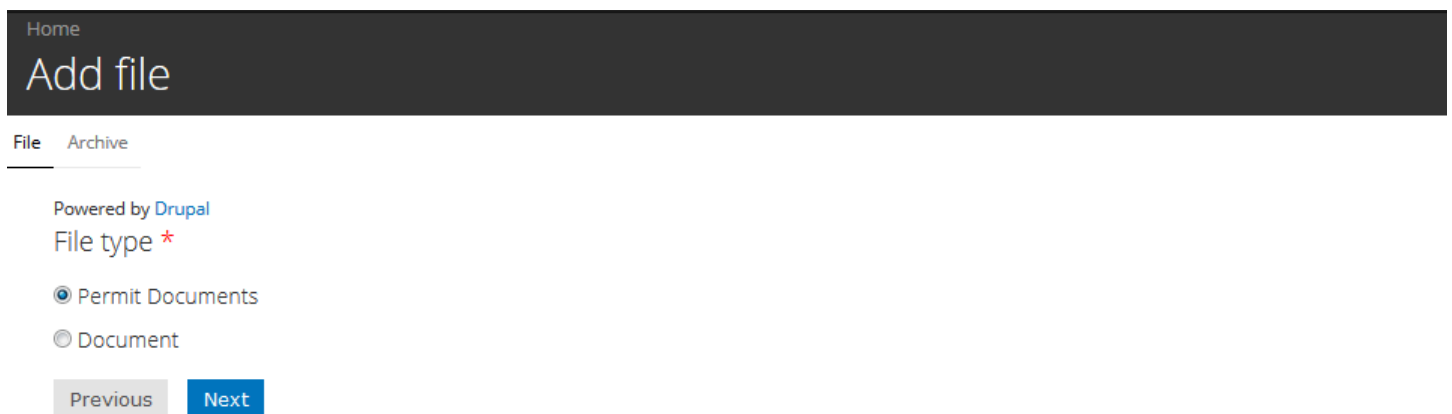
At this point, all necessary settings should be configured and everything is set for file entity revisions and moderation. For this demonstration, we will add a pdf file through the “Add file” link on the page for Files under the Content menu item. Note: if you are running Domain Access, you will need to also add and configure a Domain Access Entity field.



Once you click add file you are directed to the upload page.



Browse to your file, upload and click “Next”, this will take you to a new page to choose what file type you are uploading. For example, if you have created and set up multiple file types that can handle pdf’s, you will see all of those options here.



After you make your selection and click next, the following option is for selecting if this is to be a Public or Private file.

Home

Add file

File Archive

Powered by Drupal

Destination *

☒ Public local files served by the webserver.

☐ Private local files served by Drupal.

Previous

Next

The following page is where you will select the taxonomy term that will affect the file path. You can also change the name of the file at this point if you need to. If you have attached any other fields to this file type, you will also see those here. When you have entered all the data necessary, click “Save”.

Home

Add file

File Archive

Powered by Drupal

Name *

sample.pdf

Permit Document Division *

Administrative Services

Domain *

☐ Send to all affiliates

☒ dep-internet

User information
Associated with Anonymous

Workflow
Not published

Workflow

☐ Promoted

☐ Sticky at the top of lists

Previous

Save

Once you have saved, the file now leaves the “creation” state and immediately moves into the first state set in your workflow configuration, which in our case will be “Permit Draft”. Find the file in the file list in the administration page and click on the link to go to the file page or the file edit screen.


<input type="checkbox"/>	Title	Type	Size	Author	Updated	Used in	Published	Operations
<input type="checkbox"/>	pdf-sample.pdf	Application Documents	7.76 KB	cms_admin	04/30/2015 - 09:38	0 places	not published	Edit Delete
<input type="checkbox"/>	summary.pdf	Different Domain Document	100.47 KB	cms_admin	04/30/2015 - 08:47	0 places	published	Edit Delete

File Entity Revisions modifies the tabs on the file page to look like the following screenshot. Since files are now following a workflow with revisions, Edit has been changed to New Draft.

dep-internet

Home

Home

Search this site... 


Navigation

- [Add content](#)
- [Add file](#)
- [Workflow](#)
- [Recent content](#)

sample.pdf

[View Current](#) [New Draft](#) [Usage](#) [Delete File](#) [Revisions Workflow](#)

Permit Document Workflow

Permit Published  [sample.pdf](#)

Permit Document Division

[Administrative Services](#)

Powered by [Drupal](#)


With proper permissions you will also be able to access this tab and others on the page that houses the files operations. You may now click on the Revisions Workflow tab to view the transition history and to change the moderation state.

Home » [sample.pdf](#)

Revisions for *sample.pdf*

[View Current](#) [New Draft](#) [Usage](#) [Delete File](#) [Revisions Workflow](#)

Powered by [Drupal](#)

Revision ID	File Info	Moderated Transitions	Revision Operations										
26 Current Revision	View	<table><thead><tr><th>Time</th><th>Previous State</th><th>Next State</th><th>User</th><th>Comment</th></tr></thead><tbody><tr><td>Fri, 04/24/2015 - 09:29</td><td>(creation)</td><td>Permit Draft</td><td>cms_admin</td><td></td></tr></tbody></table>	Time	Previous State	Next State	User	Comment	Fri, 04/24/2015 - 09:29	(creation)	Permit Draft	cms_admin		<div>Current Workflow State: Permit Draft</div> <div>Change Workflow State:</div> <div>Permit Draft </div> <div>Workflow comment</div> <div><div></div><div>A comment to put in the workflow log.</div><div>Update workflow</div></div>
Time	Previous State	Next State	User	Comment									
Fri, 04/24/2015 - 09:29	(creation)	Permit Draft	cms_admin										

At this point, the file is not published and thus not accessible to the public. Files in these states are housed in the revisions folder as set by your configuration for the revisions stream wrapper from earlier. Links to view the actual file are only accessible to those users with the proper permissions. When you are ready to publish, you can enter in a comment, change the workflow state to your published state, and click "Update workflow" as seen below.

Home » sample.pdf

Revisions for *sample.pdf*

View Current New Draft Usage Delete File Revisions Workflow

Powered by Drupal

Revision ID File Info Moderated Transitions

26
Current Revision [View](#)

Time	Previous State	Next State	User	Comment
Fri, 04/24/2015 - 09:29	(creation)	Permit Draft	cms_admin	

Revision Operations

Current Workflow State: **Permit Draft**

Change Workflow State:

Permit Published ▾

Workflow comment

This is now being published.

A comment to put in the workflow log.

Update workflow

With workflow updated, you will see the file is now published and is accessible to the public as indicated by the “Accessible” message below the revision ID.

To clarify Current Revision is the file revision that is currently under moderation, and does not necessarily indicate the most recent revision (explained more below). Accessible indicates the file version which is published, utilizing the persistent file URL and is accessible to the public.

Home » sample.pdf

Revisions for *sample.pdf*

View Current New Draft Usage Delete File Revisions Workflow

Powered by Drupal

Revision ID File Info Moderated Transitions

26
Current
Revision
Accessible [View](#)

Time	Previous State	Next State	User	Comment
Fri, 04/24/2015 - 10:16	Permit Draft	Permit Published	cms_admin	This is now being published.
Fri, 04/24/2015 - 09:29	(creation)	Permit Draft	cms_admin	

Revision Operations

Current Workflow State: **Permit Published**

Change Workflow State:

Permit Published ▾

Workflow comment

A comment to put in the workflow log.

Update workflow

The tab labels have also been modified for better UX to more easily understand the file operations.

- View Current lets you view the current file version under moderation (may not be the published version).
- New Draft creates a new draft revision of the current revision under moderation.
- Delete File will completely delete the file and all the revisions of that file.
- Revisions Workflow is the tab seen above and controls all the moderation of the file.

For sites with Domain Access, in the File Info column the file's assigned domains will also be listed as seen in the below screenshot.

Revision ID	File Info	Moderated Transitions	Revision Operations															
8	View (wiki-dev-depinternet, admin-dev-depinternet, dev-depinternet)	<table border="1"><thead><tr><th>Time</th><th>Previous State</th><th>Next State</th><th>User</th><th>Comment</th></tr></thead><tbody><tr><td>Thu, 04/30/2015 - 09:38</td><td>General Documents Published</td><td>General Documents Draft</td><td>cms_admin</td><td></td></tr><tr><td>Thu, 04/30/2015 - 09:37</td><td>General Documents Published</td><td>General Documents Draft</td><td>cms_admin</td><td></td></tr></tbody></table>	Time	Previous State	Next State	User	Comment	Thu, 04/30/2015 - 09:38	General Documents Published	General Documents Draft	cms_admin		Thu, 04/30/2015 - 09:37	General Documents Published	General Documents Draft	cms_admin		<p>Current Workflow State: General Documents Draft</p> <p>Change Workflow State: General Documents Draft</p> <p>Workflow comment</p> <p>A comment to put in the workflow log.</p> <p>Update workflow</p>
Time	Previous State	Next State	User	Comment														
Thu, 04/30/2015 - 09:38	General Documents Published	General Documents Draft	cms_admin															
Thu, 04/30/2015 - 09:37	General Documents Published	General Documents Draft	cms_admin															

A change only in the moderation state remains connected only to its file revision that the state change affects as seen above. A change in any fields associated with a file and/or newer versions of a file being uploaded to replace the older versions, will create a new file revision as seen below. You will see these changes grouped by row.

For example, in the below screenshot, revision ID 73 is the Current Revision and is also the published (publicly accessible version). Once a New Draft has been clicked, and a new file version uploaded to replace the current version, the Revisions Workflow tab will appear separating out the files by ID.

Revisions for *removesharedmailbox.pdf*

View Current New Draft Usage Delete File Revisions Workflow

Powered by Drupal

Revision ID	File Info	Moderated Transitions	Revision Operations																									
75	View (sub-dev-depinternet, dep-internet, All Sites)	<table border="1"><thead><tr><th>Time</th><th>Previous State</th><th>Next State</th><th>User</th><th>Comment</th></tr></thead><tbody><tr><td>Wed, 05/06/2015 - 07:34</td><td>General Documents Published</td><td>General Documents Draft</td><td>cms_admin</td><td></td></tr></tbody></table>	Time	Previous State	Next State	User	Comment	Wed, 05/06/2015 - 07:34	General Documents Published	General Documents Draft	cms_admin		Make Current Revision Delete this Revision															
Time	Previous State	Next State	User	Comment																								
Wed, 05/06/2015 - 07:34	General Documents Published	General Documents Draft	cms_admin																									
73	View (sub-dev-depinternet, dep-internet, All Sites) Current Revision Accessible	<table border="1"><thead><tr><th>Time</th><th>Previous State</th><th>Next State</th><th>User</th><th>Comment</th></tr></thead><tbody><tr><td>Wed, 05/06/2015 - 07:34</td><td>General Documents Draft</td><td>General Documents Published</td><td>cms_admin</td><td></td></tr><tr><td>Wed, 05/06/2015 - 07:34</td><td>General Documents Review</td><td>General Documents Published</td><td>cms_admin</td><td>Here's a comment for when we published the file.</td></tr><tr><td>Wed, 05/06/2015 - 07:34</td><td>General Documents Draft</td><td>General Documents Review</td><td>cms_admin</td><td>This is a comment when changing the state.</td></tr><tr><td>Wed, 04/29/2015 - 17:04</td><td>(creation)</td><td>General Documents Draft</td><td>cms_admin</td><td></td></tr></tbody></table>	Time	Previous State	Next State	User	Comment	Wed, 05/06/2015 - 07:34	General Documents Draft	General Documents Published	cms_admin		Wed, 05/06/2015 - 07:34	General Documents Review	General Documents Published	cms_admin	Here's a comment for when we published the file.	Wed, 05/06/2015 - 07:34	General Documents Draft	General Documents Review	cms_admin	This is a comment when changing the state.	Wed, 04/29/2015 - 17:04	(creation)	General Documents Draft	cms_admin		<p>Current Workflow State: General Documents Published</p> <p>Change Workflow State: General Documents Published</p> <p>Workflow comment</p> <p>A comment to put in the workflow log.</p> <p>Update workflow</p>
Time	Previous State	Next State	User	Comment																								
Wed, 05/06/2015 - 07:34	General Documents Draft	General Documents Published	cms_admin																									
Wed, 05/06/2015 - 07:34	General Documents Review	General Documents Published	cms_admin	Here's a comment for when we published the file.																								
Wed, 05/06/2015 - 07:34	General Documents Draft	General Documents Review	cms_admin	This is a comment when changing the state.																								
Wed, 04/29/2015 - 17:04	(creation)	General Documents Draft	cms_admin																									

In the above example revision ID 75 is the latest version uploaded. However, revision 73 is still the current version. To then make the newly uploaded file version 75 the current and then published (and publicly accessible) version, you must first click on the link for "Make Current Version" found under Revision Operations. This will take you to a screen to confirm this action seen below.

Home » removesharedmailbox.pdf » Revisions Workflow

Are you sure you want to make the revision from *Wed, 05/06/2015 - 07:34* the current revision?

Powered by Drupal

[Make Current](#) [Cancel](#)

Click the button for Make Current, and this will now make file version 75, the current version of the file as seen below.

Home » [removesharedmailbox.pdf](#)

Revisions for *removesharedmailbox.pdf*

View Current New Draft Usage Delete File **Revisions Workflow**

✓

- File: revisions//applications/Florida-Coastal-Office/sample_15_0.pdf has been unpublished!
- application_documents *removesharedmailbox.pdf* has been reverted back to the revision from *Wed, 05/06/2015 - 07:46*.

Powered by [Drupal](#)

Revision ID

File Info

Moderated Transitions

Revision Operations

75

Current
Revision

[View](#)
(sub-dev-depinternet,
dep-internet, All Sites)

Time	Previous State	Next State	User	Comment
Wed, 05/06/2015 - 07:46	General Documents Published	General Documents Draft	cms_admin	
Wed, 05/06/2015 - 07:34	General Documents Published	General Documents Draft	cms_admin	

Current Workflow State:
General Documents Draft

Change Workflow State:

General Documents Draft

Workflow comment

A comment to put in the workflow log.

Update workflow

73

Accessible

[View](#)
(sub-dev-depinternet,
dep-internet, All Sites)

Time	Previous State	Next State	User	Comment
Wed, 05/06/2015 - 07:34	General Documents Draft	General Documents Published	cms_admin	
Wed, 05/06/2015 - 07:34	General Documents Review	General Documents Published	cms_admin	Here's a comment for when we published the file.
Wed, 05/06/2015 - 07:34	General Documents Draft	General Documents Review	cms_admin	This is a comment when changing the state.
Wed, 04/29/2015 - 17:04	(creation)	General Documents Draft	cms_admin	

Make Current Revision

Delete this Revision

What has to be made very clear is at this point, is file version 75 is now the current version of the file that is in moderation and that file version 73 is still the published version (publicly accessible) of the file. So the persistent file URL still points to the file version 73 and not to the file version 75. You can follow the View links to see this in action.

Now to make the new version of the file (revision ID 75) the published version, you simply change the workflow state of that version (revision ID 75) to a published state and add an optional workflow comment if you like.

Revisions for *removesharedmailbox.pdf*[View Current](#) [New Draft](#) [Usage](#) [Delete File](#) [Revisions Workflow](#)

- ✓ File: revisions//applications/Florida-Coastal-Office/sample_15_0.pdf has been unpublished!
- application_documents removesharedmailbox.pdf has been reverted back to the revision from Wed, 05/06/2015 - 07:46.

Powered by [Drupal](#)

Revision ID	File Info	Moderated Transitions	Revision Operations																									
75 Current Revision	View (sub-dev-depinternet, dep-internet, All Sites)	<table><thead><tr><th>Time</th><th>Previous State</th><th>Next State</th><th>User</th><th>Comment</th></tr></thead><tbody><tr><td>Wed, 05/06/2015 - 07:46</td><td>General Documents Published</td><td>General Documents Draft</td><td>cms_admin</td><td></td></tr><tr><td>Wed, 05/06/2015 - 07:34</td><td>General Documents Published</td><td>General Documents Draft</td><td>cms_admin</td><td></td></tr></tbody></table>	Time	Previous State	Next State	User	Comment	Wed, 05/06/2015 - 07:46	General Documents Published	General Documents Draft	cms_admin		Wed, 05/06/2015 - 07:34	General Documents Published	General Documents Draft	cms_admin		<p>Current Workflow State: General Documents Draft</p> <p>Change Workflow State:</p> <p>General Documents Published ▼</p> <p>Workflow comment</p> <p>Now changing the state to publish version 75, which will then also unpublish version 73 and switch the persistent file URL to point to file version 75.</p> <p>A comment to put in the workflow log.</p> <p>Update workflow</p>										
Time	Previous State	Next State	User	Comment																								
Wed, 05/06/2015 - 07:46	General Documents Published	General Documents Draft	cms_admin																									
Wed, 05/06/2015 - 07:34	General Documents Published	General Documents Draft	cms_admin																									
73 Accessible	View (sub-dev-depinternet, dep-internet, All Sites)	<table><thead><tr><th>Time</th><th>Previous State</th><th>Next State</th><th>User</th><th>Comment</th></tr></thead><tbody><tr><td>Wed, 05/06/2015 - 07:34</td><td>General Documents Draft</td><td>General Documents Published</td><td>cms_admin</td><td></td></tr><tr><td>Wed, 05/06/2015 - 07:34</td><td>General Documents Review</td><td>General Documents Published</td><td>cms_admin</td><td>Here's a comment for when we published the file.</td></tr><tr><td>Wed, 05/06/2015 - 07:34</td><td>General Documents Draft</td><td>General Documents Review</td><td>cms_admin</td><td>This is a comment when changing the state.</td></tr><tr><td>Wed, 04/29/2015 - 17:04</td><td>(creation)</td><td>General Documents Draft</td><td>cms_admin</td><td></td></tr></tbody></table>	Time	Previous State	Next State	User	Comment	Wed, 05/06/2015 - 07:34	General Documents Draft	General Documents Published	cms_admin		Wed, 05/06/2015 - 07:34	General Documents Review	General Documents Published	cms_admin	Here's a comment for when we published the file.	Wed, 05/06/2015 - 07:34	General Documents Draft	General Documents Review	cms_admin	This is a comment when changing the state.	Wed, 04/29/2015 - 17:04	(creation)	General Documents Draft	cms_admin		<p>Make Current Revision Delete this Revision</p>
Time	Previous State	Next State	User	Comment																								
Wed, 05/06/2015 - 07:34	General Documents Draft	General Documents Published	cms_admin																									
Wed, 05/06/2015 - 07:34	General Documents Review	General Documents Published	cms_admin	Here's a comment for when we published the file.																								
Wed, 05/06/2015 - 07:34	General Documents Draft	General Documents Review	cms_admin	This is a comment when changing the state.																								
Wed, 04/29/2015 - 17:04	(creation)	General Documents Draft	cms_admin																									

Now click Update workflow. File version 75 immediately becomes the published version and is now accessible to the public as indicated by the “Accessible” message in the revision ID column for file version ID 75. The persistent URL that was pointing to version 73, now points to version 75 and file version 75 is no longer accessible to the public but is now only accessible to those with editing (or similar) permissions for this file/file type. Changes are seen below.

Revisions for *removesharedmailbox.pdf*[View Current](#) [New Draft](#) [Usage](#) [Delete File](#) [Revisions Workflow](#)

- ✓ File: public//applications/Florida-Coastal-Office/removesharedmailbox.pdf has been published!

Powered by [Drupal](#)

Revision ID	File Info	Moderated Transitions	Revision Operations																									
75 Current Revision Accessible	View (sub-dev-depinternet, dep-internet, All Sites)	<table><thead><tr><th>Time</th><th>Previous State</th><th>Next State</th><th>User</th><th>Comment</th></tr></thead><tbody><tr><td>Wed, 05/06/2015 - 07:57</td><td>General Documents Draft</td><td>General Documents Published</td><td>cms_admin</td><td>Now changing the state to publish version 75, which will then also unpublish version 73 and switch the persistent file URL to point to file version 75.</td></tr><tr><td>Wed, 05/06/2015 - 07:46</td><td>General Documents Published</td><td>General Documents Draft</td><td>cms_admin</td><td></td></tr><tr><td>Wed, 05/06/2015 - 07:34</td><td>General Documents Published</td><td>General Documents Draft</td><td>cms_admin</td><td></td></tr></tbody></table>	Time	Previous State	Next State	User	Comment	Wed, 05/06/2015 - 07:57	General Documents Draft	General Documents Published	cms_admin	Now changing the state to publish version 75, which will then also unpublish version 73 and switch the persistent file URL to point to file version 75.	Wed, 05/06/2015 - 07:46	General Documents Published	General Documents Draft	cms_admin		Wed, 05/06/2015 - 07:34	General Documents Published	General Documents Draft	cms_admin		<p>Current Workflow State: General Documents Published</p> <p>Change Workflow State:</p> <p>General Documents Published ▼</p> <p>Workflow comment</p> <p>A comment to put in the workflow log.</p> <p>Update workflow</p>					
Time	Previous State	Next State	User	Comment																								
Wed, 05/06/2015 - 07:57	General Documents Draft	General Documents Published	cms_admin	Now changing the state to publish version 75, which will then also unpublish version 73 and switch the persistent file URL to point to file version 75.																								
Wed, 05/06/2015 - 07:46	General Documents Published	General Documents Draft	cms_admin																									
Wed, 05/06/2015 - 07:34	General Documents Published	General Documents Draft	cms_admin																									
73	View (sub-dev-depinternet, dep-internet, All Sites)	<table><thead><tr><th>Time</th><th>Previous State</th><th>Next State</th><th>User</th><th>Comment</th></tr></thead><tbody><tr><td>Wed, 05/06/2015 - 07:34</td><td>General Documents Draft</td><td>General Documents Published</td><td>cms_admin</td><td></td></tr><tr><td>Wed, 05/06/2015 - 07:34</td><td>General Documents Review</td><td>General Documents Published</td><td>cms_admin</td><td>Here's a comment for when we published the file.</td></tr><tr><td>Wed, 05/06/2015 - 07:34</td><td>General Documents Draft</td><td>General Documents Review</td><td>cms_admin</td><td>This is a comment when changing the state.</td></tr><tr><td>Wed, 04/29/2015 - 17:04</td><td>(creation)</td><td>General Documents Draft</td><td>cms_admin</td><td></td></tr></tbody></table>	Time	Previous State	Next State	User	Comment	Wed, 05/06/2015 - 07:34	General Documents Draft	General Documents Published	cms_admin		Wed, 05/06/2015 - 07:34	General Documents Review	General Documents Published	cms_admin	Here's a comment for when we published the file.	Wed, 05/06/2015 - 07:34	General Documents Draft	General Documents Review	cms_admin	This is a comment when changing the state.	Wed, 04/29/2015 - 17:04	(creation)	General Documents Draft	cms_admin		<p>Make Current Revision Delete this Revision</p>
Time	Previous State	Next State	User	Comment																								
Wed, 05/06/2015 - 07:34	General Documents Draft	General Documents Published	cms_admin																									
Wed, 05/06/2015 - 07:34	General Documents Review	General Documents Published	cms_admin	Here's a comment for when we published the file.																								
Wed, 05/06/2015 - 07:34	General Documents Draft	General Documents Review	cms_admin	This is a comment when changing the state.																								
Wed, 04/29/2015 - 17:04	(creation)	General Documents Draft	cms_admin																									

Current Issues or Bugs

Over view of some of the possible issues or any bugs needing review and possible code changes. So far nothing here is a major issue and deals more with user experience:

No way to see field changes on files

When a New Draft is created, once saved there is no way to see any changes made to the fields attached to the file. For example if a various text field was edited, after saving we cannot view that revision to see those changes unless it is made the current revision. Possible solutions:

Utilizing required moderation comments here would be good. However we have turned off the Workflow Widget on the workflow field for file types (selected Hide the widget on Entity form). If we leave that widget in, we do have the ability to see the Workflow Comment form, and could make that a required field upon saving any new draft.

Maybe look at including the Workflow comment field on a file's New Draft page without including the moderation state selector? Could be hidden with css, but would it be better done in another way? Either only pulling/placing the Workflow Comment field without the selector, or pull the whole widget while making the workflow selector inaccessible (grayed out)?

Look at incorporating the Diff module???

Or simply place the Workflow selector in its entirety on the file create/edit pages.

Empty file moderation comment when changing Current Revision

When changing to a previous version of a file (revert) or making a newly uploaded file the current, a file history is automatically created which inherits the current versions last history as its initial history and also leaves the Workflow comment blank. It would be nice if we can automatically insert a default comment here that says "File changed to Current Revision". See the empty comment box below for where it should appear.

Revision ID

File Info

Moderated Transitions

Revision Operations

82
Current
Revision

[View](#)
(sub-dev-depinternet,
dep-internet, All Sites)

Time	Previous State	Next State	User	Comment
Wed, 05/06/2015 - 11:36	General Documents Published	General Documents Draft	cms_admin	
Wed, 05/06/2015 - 11:36	General Documents Published	General Documents Draft	cms_admin	Saving a new draft to demonstrate the empty comment in history.

Current Workflow State:
General Documents Draft

Change Workflow State:
General Documents Draft

Workflow comment *

A comment to put in the workflow log.

Update workflow