

## Documents: Permissions explained

This article details the permissions available in the document application and explains how to set these to inherit across directory levels.

An [application administrator](#) of People can create [People Roles and Groups](#) and add users to these.

An application administrator of Documents can then specify Users/Roles/Groups in permissions across folders and files to give or restrict access.

This differing permissions structure across users and the document directory will determine each user's experience and ensure that they can only interact with the appropriate content to the correct degree.

An application administrator of Documents can change permissions on the admin side of the application as detailed [here](#)

An end user (with this ability) can change permissions from the front end of the application by following the steps [here](#)

## Available Permissions

Permissions that can be set on folders and files are the same:

A screenshot of the permissions interface for 'Sample Document 9.pdf'. The interface includes a back arrow, the document name, a 'Full screen' button, and a 'Version 1' indicator. There is an unchecked checkbox for 'Inherit rights from parent folder'. Below this is a search bar with 'Start typing to add...' and 'Browse' and 'Remove' buttons. A dropdown menu is open, showing 'All registered Owner' and 'Role: Administrators'. A list of permissions follows: 'View' (checked), 'Create Draft' (unchecked), 'Edit & Approve' (checked), 'Move/delete' (checked), 'Edit Metadata' (checked), and 'Edit permissions' (unchecked). At the bottom, there are 'View effective permissions...' and 'Apply permissions' and 'Close' buttons.

- **View** – Users can see this content listed in the application, if a document they can [download it](#).
- **Create Draft** – This user can only submit draft documents rather than directly upload them. Other users with 'Approve' permissions can publish these drafts.
  - If you're not using an approval process, this permission can be left blank for all. If you are, it's required to give a user "View" and "Create Draft" permissions only, then give another group/role all other permissions, allowing them to publish draft documents submitted.
- **Edit & Approve** – Users can add documents to a folder, create sub-folders in the parent folder, as well as approve other users' drafts (if drafts are being utilised).
- **Move/Delete** – Users can add folders and documents to the clipboard and [relocate these](#) to another area of the directory, as well as delete content.
  - The trashcan retains 'deleted' items for 30 days in case mistakes are made, and content can be retrieved or permanently deleted from the trashcan. End users can only see content they moved to the trash can, administrators of documents can see everyone's.
- **Edit Metadata** – Users can edit the associated [metadata](#) of folders/documents e.g. properties such as titles, tags, author, review date, etc.
- **Edit Permissions** – Users can alter the permissions on a folder or document, giving or restricting access to all of the above.
  - Only application administrators can give/remove this permission from the [admin side](#).

**Please note:** "View" and "Edit & approve" permissions only will allow users to upload files and new versions but "Edit metadata" permissions will be needed too if they should also be able to edit document properties e.g title, tags, metadata

## General logic to understand when assigning permissions

In Documents, access is given to the top-level (parent folder) by inputting permissions groups/roles/users.

The permissions groups/roles/users put into the parent will impact which groups can be picked in its content and/or subfolders.

Logically, users that are not defined in the parent cannot access the content in it - or be chosen for any of its sub level content - they need to be included in the parent first.

Subsequently, the system will only offer groups/roles/users that have been defined in the parent when setting permissions for its content or subfolders/files.

If a group/role/user is not appearing in the subfolder permissions, it will be because they have not been included in the parent or have been left out at some point in the folder levels above the one being modified.

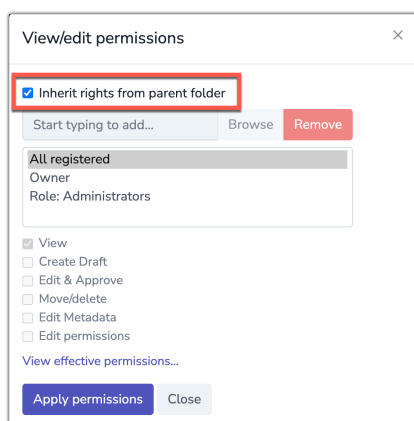
To troubleshoot, the permissions at each level above the content you are trying to update can be checked to see where the group/role/user has been left out, with this is rectified by adding them in and saving.

## Setting 'inherit' permissions

### Folder level

To speed up creating folder structure, Claromentis has an 'inherit' checkbox option for permissions for subfolders.

This will pull through the permissions set on the folder above and apply this to the subfolder.



Inheriting can be turned off by deselecting the checkbox and custom permissions added into new folders instead.

**Please note:** When not inheriting only users/roles/groups defined in the parent folder can be selected as Documents logic requires users/roles/groups to be defined higher in the directory for use in the lower levels.

## Document level

Document permissions also offer an 'inherit' option.

This means the permissions for the folder the document is being put into will also be applied to the file.

The screenshot shows the 'Add new Document' form. The 'Permissions' section is highlighted with a red box. It contains the following elements:

- Inherit rights from parent folder
- Start typing to add...
- All registered
- Owner
- Role: Administrators
- View
- Create Draft
- Edit & Approve
- Move/Delete
- Edit Metadata
- Edit permissions
- [View effective permissions...](#)

Other sections of the form include:

- Files:** Choose file, or select from pre-uploaded...
- Tags:** Input field, popular tags: intranet, claromentis, microsoft, excel, social
- Review Date:** Calendar icon
- Author:** Select user dropdown, Browse button
- Related Documents:** Add new, Select document
- Description:** Text area
- Version comment:** Text area
- Notifications:**  None,  Select recipients
- Status:**  Mark as draft
- 

The 'inherit' option can be deselected here too so custom permissions can be set for documents that differ from the folder permissions.

This allows different users/roles/groups to see different files listed when in the same folder based on their permissions.

**Please note:** When not inheriting only users/roles/groups defined in the parent folder can be selected as Documents logic requires users/roles/groups to be defined higher in the directory for use in the lower levels.