

# Users and Permissions

- [Creating User](#)
- [Managing Permissions](#)

# Creating User

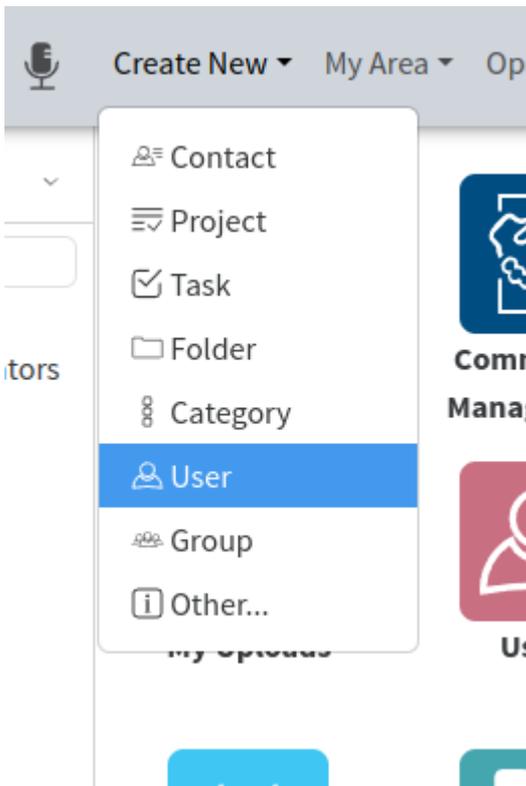
Like every Server-clients oriented software Profuz LAPIS allows to manage users with different level of privileges. Adding new user can be done from main Create New drop-down menu. Same is valid for adding new Groups. Each user can be defined to be member of different groups e.g.

- Subtitling QC
- Subtitling Editing
- Subtitling Approval

Accessing permission rights for each task, subtask or other data can be managed by entering inside the location. Go to Action → Edit → Permissions. Click to edit per specific permission request.

# Creating User

To create an User go to **Create New - User**



The new window appear where the details about User have to be defined.

Only Username is mandatory.

Created by Administrator on 15/11/2024, 21:57:07

GP George Peterson (Draft-User) 🔒 🔗 ⋮

Standard edit ▾ Collaborators: Administrator 

Username \*

Password

Repeat Password

Groups  
 ✕ 🔍

- Subtitling Clients 🔍
- Subcontractors 🔍

Contact Book  
 🔍

User's Folder  
 🔍

↶ Cancel Create

By pressing Create, the User is created.

More details can be added

Created by Administrator on 15/11/2024, 21:59:24

GP George Peterson (User) 🔒 🔗 🔗 ⋮

 George Peterson ✉️ Send Reset Password Email 🔑 Change Password

**Overview** Details Main Contact All Contacts Calendar Activities

Username  
George Peterson

Time Zone

Edit Contact Details

GP George Peterson (User) Created by Administrator on 15/11/2024, 21:59:24

**GP** George Peterson Send Reset Password Email Change Password

Overview **Details** Main Contact All Contacts Calendar Activities

**Properties**

Username  
George Peterson

Groups  
**Managers**

Contacts

Permissions  
[View Permissions](#)

User's Folder

To connect an Existing contact to the User:

- By quick editing

GP George Peterson (User) Last modified by Administrator on 15/11/2024, 22:01:29

**GP** George Peterson Send Reset Password Email Change Password

Overview **Details** Main Contact All Contacts Calendar Activities

**Properties**

Username  
George Peterson

Groups  
**Managers**

**Contacts**

georg

George Peterson

Yota Georgakopoulou

User's Folder

- By Actions - Edit

# GP George Peterson (Draft-User)



Standard edit ▾

Collaborators: Administrator

Username \*

George Peterson

Password

Password

Repeat Password

Groups

Managers × Filter...

Contacts

ged ×

George Peterson

Yota Georgakopoulou

User's Folder

Filter...

Activities

Types

Cancel

Save

The details of the contact now appear in the Main Contact tab

# GP George Peterson (User)



## George Peterson

Send Reset Password Email Change Password

[Overview](#) [Details](#) **[Main Contact](#)** [All Contacts](#) [Calendar](#) [Activities](#)

Edit Contact Details



Full Name  
George Peterson

Web sites

Phone Numbers  
**+555-123-4567-890**

Emails  
**george@example.net**

Properties

Description  
Loves automobiles and dogs. Has its own pet.

Company

When contact is attached it may have also an image of the user

**GP** George Peterson (User)

---



George Peterson

Overview Details **Main Contact** All Contacts Calendar

---

Full Name  
George Peterson

George Peterson (User) Last modified by Administrator on 15/11/2024, 22:03:

---



George Peterson

Send Reset Password Email Change Password

Overview Details **Main Contact** All Contacts Calendar Activities

Edit Contact Details



Full Name  
George Peterson

Web sites

Phone Numbers  
**+555-123-4567-890**

Emails  
**george@example.net**

Properties

Description  
Loves automobiles and dogs. Has its own pet.

Companv

# Managing Permissions

## Understanding Permissions logic

Permissions in LAPIS are per object. They control the common operations that are allowed or denied on an object for particular Users and Groups. The most important ones are:

- **read** - allows/denies an object to be accessible
- **write** - allows/denies an object to be modifiable
- **delete** - allows/denies object to be delete-able
- **download** - allows/denies an object to be downloadable
- **owner** - allows/denies all operations

Each object in the system has **explicitly set** and **implicit** permissions. Explicit permissions are set to each object and can be changed manually, while **implicit** permissions are applied to an object through connected to it other objects (for example from a Folder object in which it is present) or other rules (for example an Assignee property gives permissions to its Task). Combination of explicit and implicit permissions gives the effective permissions of an object. Explicit permissions overwrite the Implicit permissions. **Deny** of an operation has priority in effective (combined) permissions.

When an object is created in the system it receives the default explicit permissions as set for its type. This is configured in the “Default Instance Permissions” property in the corresponding Type or if not set there, from the first ancestor Type in which it is defined.

## Define Default Instance Permissions for a Type

You have to be part of Administrators group to be able to perform these tasks. Go to the menu Administration/Types and find the type you want to manage.

- Permissions
- Types**
- Permissions Presets

ainer)

Last modified by [Administrator](#) on 12/01/2024, 06:31:46

## Asset (Type)



### Properties

Name

Asset

Display Name

Asset

Base Types

 **Object**

Java Type

Properties

 **Name (Symbol)**,  **Contents (File-StreamableContent)**,  **Title (String)**,  **Description (String)**

Instance Thumbnail

Additional Types Properties

Is Versioned by Default

Default Instance Permissions



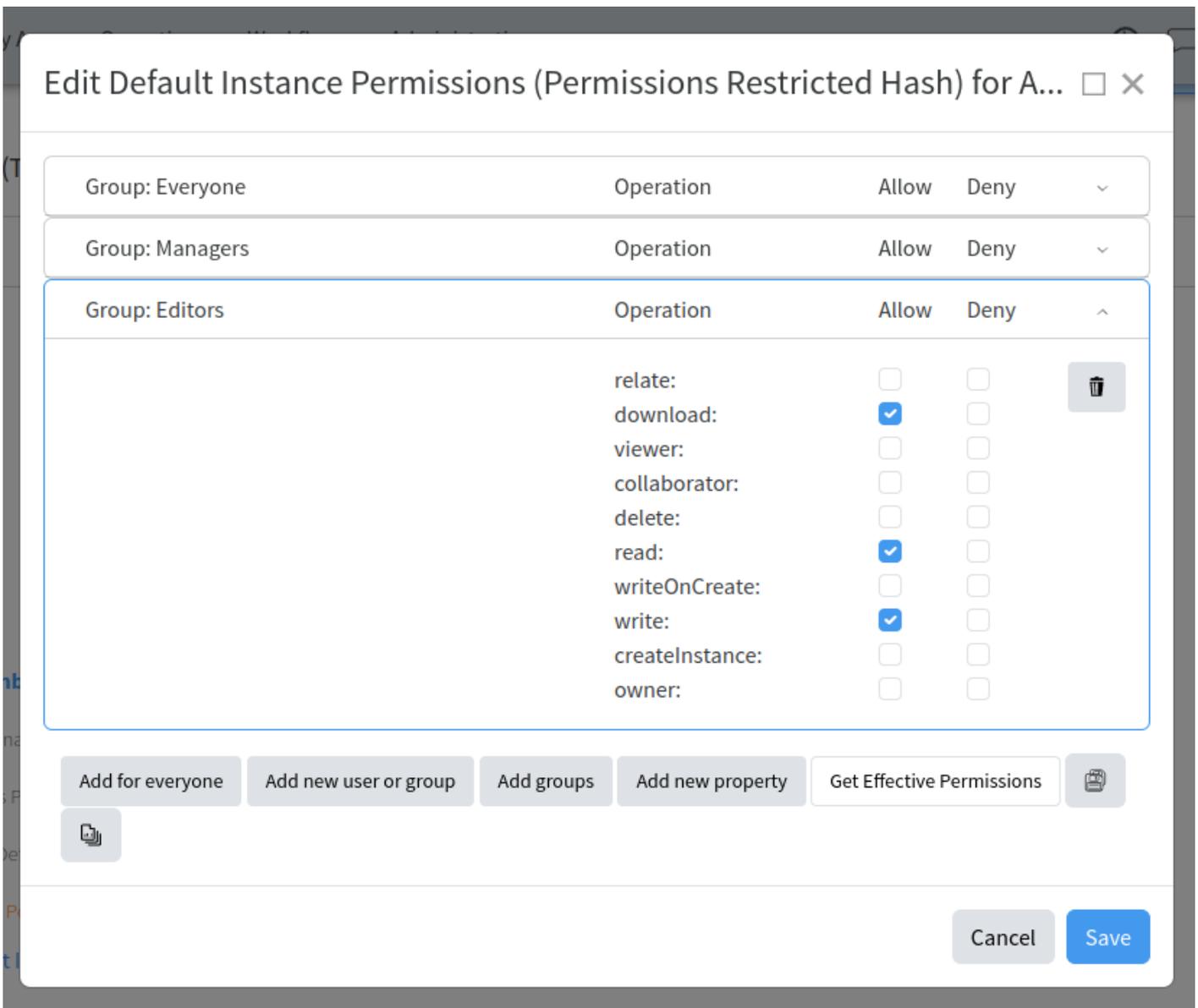
[Edit Default Instance Permissions](#)

Properties View Placement

Name: wide

Contents: wide

Edit the “Default Instance Permissions” property, add users, groups or Everyone and allow or deny the operations you want for each of those:



Note that those permissions will be applied to the objects created from now on. All existing objects will keep their existing permissions.

## Using Folders, Categories and Tags to manage permissions

Folders, Categories and Tags are all configured to provide inherited permissions to objects they contain. It is enough to set or change permissions to a folder in order to easily propagate them to multiple objects. Note that objects' explicit permissions will overwrite those set in a folder.

# Change explicit permissions of a particular object

Click on Edit Permissions button in the object header:

The screenshot shows a file management interface for a video file named "Turbo20 Trailer.mp4" (video/mp4). The file was created by "system" on 18/06/2024 at 12:53:30. The interface includes several action buttons: "Create Subtitle Suite", "Make Asset", "Transcribe with Microsoft", and "Full Delete". A video player is embedded, showing a "DREAMWORKS ANIMATION SKG" logo. The sidebar on the right contains a "File Copies" section with three entries of the same file, each 4.44 MB. Below that is a "Transformations" section with one entry: "transcribe\_Turbo20 Trailer.srt" (346 B, application/x-subrip). At the bottom, there is a "Properties" section with "Start Timecode" and "Timecode Keyframes" options. A table with columns "Time" and "Timecode" is shown, containing "No Entries".

then click on "Edit Permissions":

The screenshot shows a dialog box titled "Permissions (Permissions Restricted Hash)". The dialog has a close button (X) in the top right corner. A button labeled "Edit Permissions" is highlighted with a blue border. At the bottom right, there are two buttons: "Cancel" and "Save".

and set the permission you want to be applied to this object:

Edit Permissions (Permissions Restricted Hash) for Turbo20 Trailer.m... □ ×

Group: Managers	Operation	Allow	Deny	▼
Property: owners	Operation	Allow	Deny	▼
User: Storage Demo User	Operation	Allow	Deny	^
	relate:	<input type="checkbox"/>	<input type="checkbox"/>	
	download:	<input type="checkbox"/>	<input type="checkbox"/>	
	viewer:	<input type="checkbox"/>	<input type="checkbox"/>	
	collaborator:	<input type="checkbox"/>	<input type="checkbox"/>	
	delete:	<input type="checkbox"/>	<input type="checkbox"/>	
	read:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	writeOnCreate:	<input type="checkbox"/>	<input type="checkbox"/>	
	write:	<input type="checkbox"/>	<input type="checkbox"/>	
	createInstance:	<input type="checkbox"/>	<input type="checkbox"/>	
	owner:	<input type="checkbox"/>	<input type="checkbox"/>	

Add for everyone Add new user or group Add groups Add new property Get Effective Permissions

Cancel Save