

# **ACEP Data Catalog User Guides**

Jenae Matson

2024-08-05

# Table of contents

<b>Welcome</b>	<b>4</b>
<b>1 Getting Started</b>	<b>5</b>
Making an Account . . . . .	5
Searching for Datasets . . . . .	6
Submitting a Comment/Complaint . . . . .	7
<b>2 Researcher</b>	<b>8</b>
Posting a Dataset . . . . .	8
Tagging a Dataset . . . . .	10
Editing/Deleting a Dataset . . . . .	11
Adding a Dataset to a Group . . . . .	12
Creating a New Group . . . . .	13
<b>3 Admin</b>	<b>15</b>
Giving User Permissions . . . . .	15
Reviewing and Publishing a Dataset . . . . .	16
Deleting Datasets/Groups/Organizations . . . . .	17
Purge Deleted Datasets/Groups/Organizations . . . . .	17
Restore Deleted Datasets . . . . .	18
Restoring a Deleted User . . . . .	20
<b>Metadata Walkthrough</b>	<b>21</b>
Dataset Fields . . . . .	21
Title* . . . . .	21
Description* . . . . .	21
Keywords . . . . .	21
Contact Points* . . . . .	21
Publisher* . . . . .	22
License* . . . . .	22
Organization* . . . . .	22
Visibility . . . . .	22
Landing Page . . . . .	22
Release Date* . . . . .	22
Modification Date* . . . . .	22
Version . . . . .	22

Version Notes . . . . .	23
Identifier . . . . .	23
Provenance . . . . .	23
Type . . . . .	23
Temporal Coverage . . . . .	23
Spatial Coverage . . . . .	23
Language . . . . .	25
Documentation . . . . .	25
Conforms to . . . . .	25
Is referenced by . . . . .	25
URI . . . . .	25
Resource Fields . . . . .	25
Name* . . . . .	25
Description* . . . . .	25
Format . . . . .	25
GitHub Repo . . . . .	26
Status . . . . .	26
License . . . . .	26
Access URL . . . . .	26
Download URL . . . . .	26
Release Date* . . . . .	26
Modification Date* . . . . .	26
Language . . . . .	26
Documentation . . . . .	26
Conforms to . . . . .	27
Access Services . . . . .	27
URI . . . . .	27
<b>For Developers</b> . . . . .	<b>28</b>
Data Sources Overview . . . . .	29
Developing the Data Catalog . . . . .	30
Basic Docker Commands . . . . .	30
Creating a Local Instance . . . . .	31
Create a New Extension . . . . .	33
Install an Extension . . . . .	33
Updating the Main Site . . . . .	33
Extensions . . . . .	34
Currently Installed . . . . .	34
Adding Alternate Schemas with ckanext-scheming . . . . .	39
Attempted Extensions . . . . .	39

# Welcome

This documentation provides some helpful tutorials for working with the ACEP Data Catalog.

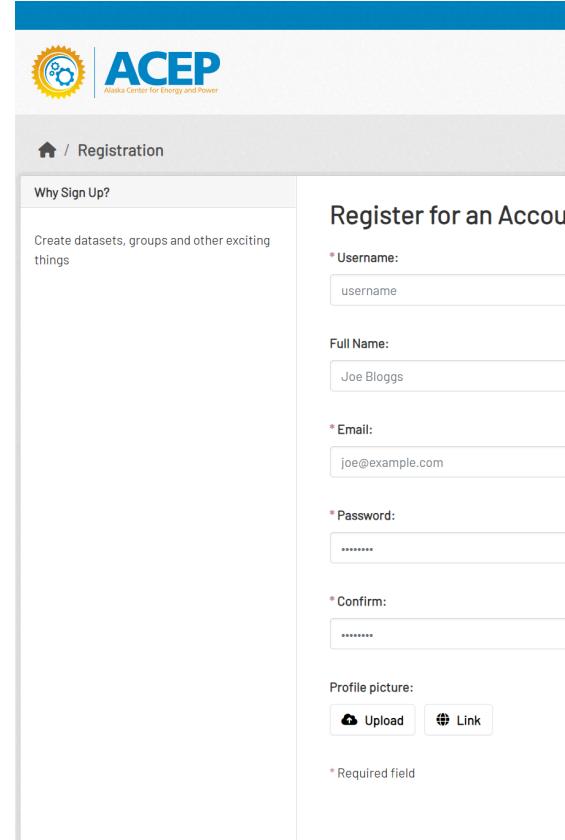
The screenshot shows the ACEP Data Catalog homepage. At the top, there is a navigation bar with links for Log In, Register, Datasets, Organizations, Groups, About, FAQ, and a search bar. The main header says "Welcome to the ACEP Data Catalog" and features a large image of solar panels and a wind turbine. To the right of the main content area is a sidebar with a word cloud of energy-related terms like "energy", "alaska", "solar", "wind", etc. Below the sidebar is a section titled "Inventory of Alaska Energy Datasets and Systems" with a link to "More Alaska Energy Data resources". At the bottom, there is a "Search Datasets" bar with a placeholder "E.g. Tanana River Test Site" and a search icon. Below the search bar are six categories: "Beneficial and Equitable Electrification" (0 datasets), "Energy Statistics" (5 datasets), "Marine Energy" (0 datasets), "Power Systems Integration" (0 datasets), "Solar Technologies" (0 datasets), and "Tanana River Test Site" (4 datasets). Each category has a corresponding icon.

# 1 Getting Started

Guides for anyone using the data catalog.

## Making an Account

1. Click the **Register** button in the upper right corner of the screen.



The screenshot shows the ACEP (Alaska Center for Energy and Power) registration page. At the top, there's a logo for ACEP and a subtext "Alaska Center for Energy and Power". Below the logo, the page title is "Registration". On the left, there's a sidebar with a "Why Sign Up?" section containing the text "Create datasets, groups and other exciting things". The main form area has several input fields:

- "Username": A text input field containing "username".
- "Full Name": A text input field containing "Joe Bloggs".
- "Email": A text input field containing "joe@example.com".
- "Password": A password input field containing "\*\*\*\*\*".
- "Confirm": A password confirmation input field containing "\*\*\*\*\*".
- "Profile picture": A section with "Upload" and "Link" buttons.

A note at the bottom right of the form says "\* Required field".

2. Fill out your information, using your UA email if you have one.
3. In order to see the Internal Use datasets or post datasets, you will need permissions granted to your account. If you are an ACEP employee, contact Liz or another admin.

# Searching for Datasets

There are many ways to search for datasets in the catalog.

- **Search Bar**

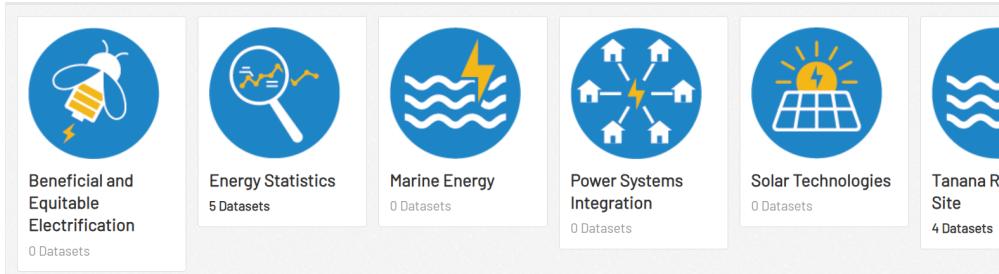
Use key words to find datasets, just like Google. On the left sidebar of the search results page, you can filter by tags and other categories to get more specific results.

The screenshot shows a search results page for the term "Tanana". The left sidebar contains filters for Organizations (ACEP Open Data - 3), Groups (Tanana River Test Site - 3), Tags (Alaska - 3, CEC - 3, Code - 3, Energy - 3, Hydrokinetic - 3, Marine - 3, MATLAB - 3, MHK - 3, Nenana - 3, Power - 3), Formats (none found), and Licenses (none found). The main search bar has "Tanana" entered, and the results are ordered by Relevance. Three datasets are listed:

- Water Horse Hydroelectric Harvester Single Oscillator Field Testing Data,...**  
Raw and processed timeseries data generated during field testing of a single oscillating hydrofoil Water Horse prototype at PMEC (Pacific Marine Energy Center) Tanana River Test...  
Tags: Alaska, CEC, Code, Current Energy Converter, Energy, Field Test Content Model, Hydrokinetic, MATLAB, MHK, Marine, Nenana, Oscillating Hydrofoil, Power, Pre-Processed Data, Processed Data, Raw Data, River, Tanana River, Technology
- Water Horse Hydroelectric Harvester Dual Oscillator Field Testing Data,...**  
Raw and processed timeseries data generated during field testing of a single oscillating hydrofoil Water Horse prototype at PMEC (Pacific Marine Energy Center) Tanana River Test...  
Tags: Alaska, CEC, Code, Current Energy Converter, Energy, Field Test Content Model, Hydrokinetic, MATLAB, MHK, Marine, Nenana, Oscillating Hydrofoil, Power, Pre-Processed Data, Processed Data, Raw Data, River, Tanana River, Technology
- Simulink Model of a New Energy EVG-005 5kW Hydropower Turbine**  
Simulink model for a New Energy 5kW hydropower turbine. ADCP data ("ds\_streamwise\_7\_13.nc") and DC voltage, DC current, and rotor rotation observed from the New Energy EVG-005...

- **Groups**

Browse through the groups that can be found at the **Groups** tab at the top of the screen



or listed on the home page.

- **Organizations**

Sort by larger categories by looking the **Organization** tab at the top of the screen.

- The **ACEP Open Data** organization holds datasets created and owned by ACEP’s research teams.
- The **ACEP Internal Use** organization holds datasets that ACEP does not own. These are useful resources for ACEP researchers. **You must be an ACEP employee to gain access to these datasets.** Contact Liz to get the necessary permissions for your account.

## **Submitting a Comment/Complaint**

If you find a problem with a dataset or have a comment or suggestion for the data catalog, please contact the ACEP data team at [uaf-acep-dcm-support@alaska.edu](mailto:uaf-acep-dcm-support@alaska.edu).

## 2 Researcher

Guides for researchers and those posting datasets to the catalog.

### Posting a Dataset

There are two ways to access the **Add Dataset** button:

- Click on the **Datasets** tab at the top of the screen
  - Click on the **Organizations** tab at the top of the screen and select the organization you want to add a dataset to:
    - Choose **ACEP Open Data** if you are posting your own data or data that ACEP owns.
    - Choose **ACEP Internal Use** if you are posting a useful dataset that ACEP does not own.
1. Click on the **Add Dataset** button above the search bar.
  2. On the first page, fill out the metadata fields for your dataset.
    - For examples and more information about the metadata fields, see the [Metadata](#)

The screenshot shows the ACEP CKAN Dataset creation interface. On the left, there's a sidebar with a question 'What are datasets?' and a detailed answer about CKAN Datasets. The main area has a green progress bar at the top with two steps: '1 Create Dataset' and '2 Add data'. Below the progress bar are several input fields: 'Title' (with placeholder 'eg. A descriptive title'), 'URL' (with placeholder 'eg. portal.lab.acep.uaf.edu/dataset/<dataset>'), 'Description' (with placeholder 'eg. Some useful notes about the data' and a note 'You can use Markdown formatting here'), 'Tags' (with placeholder 'eg. economy, mental health, government'), 'License' (a dropdown menu with placeholder 'Please select the license' and a note 'License definitions and additional information can be found at [opendefinition.org](#)'), 'Organization' (a dropdown menu with placeholder 'ACEP Internal Use'), and 'Visibility' (a dropdown menu). The URL field has an 'Edit' button next to it.

### Walkthrough.

- Double check the organization field.
    - Choose **ACEP Open Data** if you are posting your own data or data that ACEP owns.
    - Choose **ACEP Internal Use** if you are posting a useful dataset that ACEP does not own.
  - If you are a researcher, your dataset's visibility will automatically be set to private. An admin will review your dataset and make it public.
3. Once you have completed the metadata fields, click the **Next: Add Data** button at the bottom of the form.

**What's a resource?**

A resource can be any file or link to a file containing useful data.

**Data:**

**Name:**

eg. January 2011 Gold Prices

**Description:**

Some useful notes about the data

You can use Markdown formatting here

**Format:**

eg. CSV, XML or JSON

This will be guessed automatically. Leave blank if you wish

4. On the next page, add the data resources to your dataset.
  - Some CSV files (maximum ~100-200MB) can be uploaded directly to the catalog
  - Alternatively, enter a link to where the data is stored, such as a GitHub repository or Google Drive.
5. If you have more resources to add, click the **Save & add another** button at the bottom of the form. Otherwise click **Finish** to post your dataset.

## Tagging a Dataset

TBD

## Editing/Deleting a Dataset

1. Navigate to the dataset page and click on the **Manage** button in the upper right.

The screenshot shows the ACEP dataset page for "Alaska Railroad Features". At the top right, there is a red circle highlighting the "Manage" button. The page includes sections for Followers (0), Organization (ACEP Internal Use), Data and Resources (listing four geojson files: Crossings.geojson, MP.geojson, Railroad\_Boundary.geojson, and Track\_Centerline.geojson, each with an "Explore" button), and Metadata (a table with two columns: Field and Value). A sidebar on the left shows a house icon with a lock and the text "ACEP Internal Use".

2. This displays the **Edit metadata** page where you can change the metadata of the dataset. After making changes, click the **Update Dataset** button at the bottom of the form.

- To delete the dataset, click the **Delete** button at the bottom of the form.

NOTE: Deleting a dataset does not remove it completely from the database. The url of the deleted dataset will not be able to be reused until it has been purged by a sysadmin.

3. To edit the resources or add more resources, click on the **Resources** tab.

- To delete a resource, select it from the list and then click the **Delete** button at the bottom of the page.

## Adding a Dataset to a Group

1. Navigate to the dataset page and click on the **Groups** tab.
2. Select a group from the dropdown menu and click the **Add to group** button.
3. To remove the dataset from a group, hover your cursor over the group and click the **Re-**

The screenshot shows the ACEP Internal Use website interface. At the top, there's a navigation bar with links for Datasets, Organizations, Groups, About, and FAQ, along with a search bar. The main content area shows the 'Alaska Railroad Features' organization profile. On the left, there's a sidebar with a house icon and a lock icon. The main panel has tabs for Dataset and Groups. Under the Groups tab, there's a list of groups, with one group, 'Power Systems Integration', circled in red. Next to this group is a red 'Remove' button.

move button.

## Creating a New Group

Groups are a good way to group together datasets that are connected. You may want to create groups for common research themes, funding organizations, or projects.

1. Click on the **Groups** tab at the top of the screen.

The screenshot shows a web browser window with the ACEP (Alaska Center for Energy and Power) logo at the top left. The top navigation bar includes links for Datasets, Organizations, Groups (which is highlighted in blue), About, and FAQ. Below the navigation, a breadcrumb trail shows the user is at the home page / Groups / Create a Group. The main content area has a title 'Create a Group'. On the left, there is a sidebar with a section titled 'What are Groups?' containing text about CKAN Groups. The main form area contains fields for 'Name' (set to 'My Group'), 'Description' (with placeholder text 'A little information about my group...'), and 'Image' (with 'Upload' and 'Link' buttons). A note at the bottom of the form states '\* Required field'.

2. Click the **Add Group** button.
3. Enter the information for the group. Find a logo or simple image to upload to represent the group.
4. Once you have entered the information, click the **Create Group** button at the bottom of the form.

# 3 Admin

Guides for admins managing the data catalog.

## Giving User Permissions

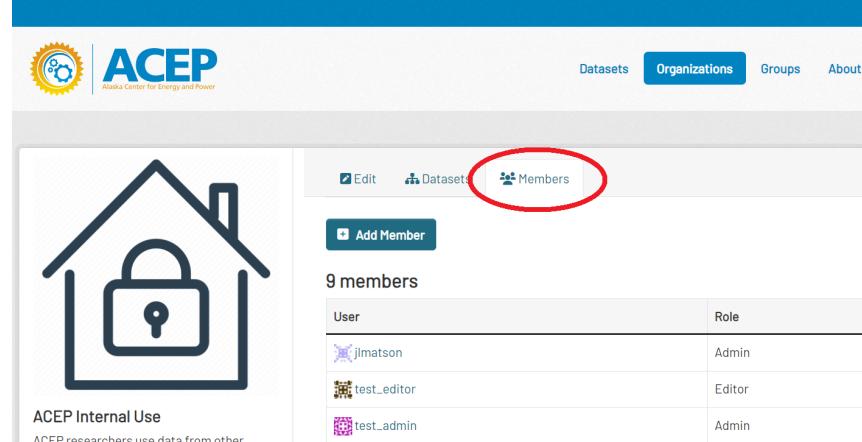
Users have roles in each organization that give them different permissions.

The roles are:

- **Member:** can see private datasets in the organization
- **Editor:** can add private datasets to the organization and delete datasets from the organization
- **Admin:** can change users' roles in an organization and publish datasets from private to public

To give a user a role in an organization:

1. Click on the **Organizations** tab at the top of the screen and click on the organization you want to add the user to.
2. Click on the **Manage** button in the upper right.



The screenshot shows the ACEP (Alaska Center for Energy and Power) data catalog interface. At the top, there is a navigation bar with links for Datasets, Organizations (which is highlighted in blue), Groups, and About. Below the navigation bar, there is a large icon of a house with a padlock inside it, labeled "ACEP Internal Use" and "ACEP researchers use data from other". To the right of the icon, there is a "Members" tab, which is circled in red. Below the tab, there is a "Add Member" button and a table showing "9 members". The table has columns for "User" and "Role". The data in the table is as follows:

User	Role
jmatson	Admin
test_editor	Editor
test_admin	Admin

3. Click on the **Members** tab at the top.

- To add a new member to the organization, click the **Add Member** button. Enter their username, select their new role from the dropdown, and click the **Add Member** button at the bottom of the form.

**Add Member**

Existing User : If you wish to add an existing user, search for their username below.

Username:

Role: Member

- To change an existing member's role, click the wrench next to their user name.

test_editor	Editor	
-------------	--------	--

- Select their new role from the dropdown menu and click **Update Member**
- To remove the user from the organization completely, click the **Delete** button.

**Edit Member**

test\_editor

Role: Editor

Delete Update Member

## Reviewing and Publishing a Dataset

1. Navigate to the dataset you would like to publish.
2. Review the metadata and ensure the files/links are correct and function properly.
3. To publish, click on the **Manage** button in the top right of the dataset page.

4. In the metadata fields, find the **Visibility** field.

The screenshot shows a form for editing a dataset. At the top, there is a dropdown menu labeled "Organization" with "ACEP Open Data" selected. Below it is a section labeled "Visibility" with three options: "Private" (selected), "Private" (highlighted in blue), and "Public". There is also a "Source" field containing the URL "https://arcg.is/4j5b50".

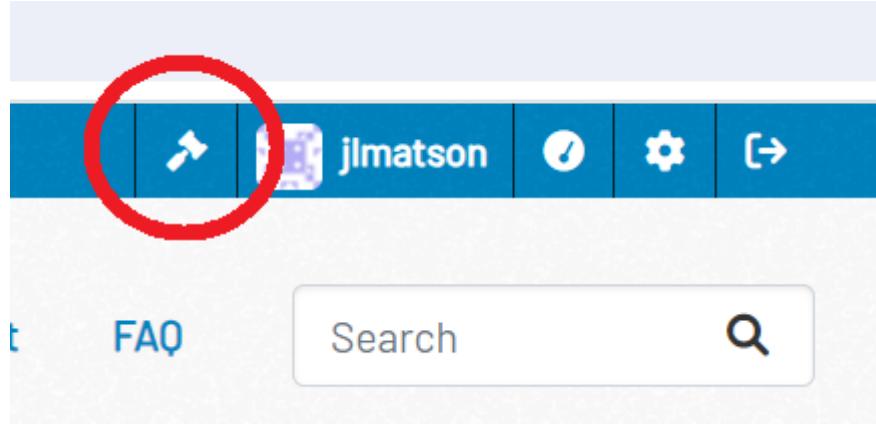
5. Change the **Visibility** field from **Private** to **Public**, and click the **Update Dataset** button at the bottom of the form.

## Deleting Datasets/Groups/Organizations

1. To delete a dataset, group, or organization, navigate to its page and click the **Manage** button in the upper right.
2. Click the **Delete** button at the bottom of the form to delete.
3. **Deleting a dataset, group, or organization does not remove it from the database.** After deleting, a sysadmin can restore the item or purge it to remove it from the database. If not purged, the url for the deleted item cannot be reused.

## Purge Deleted Datasets/Groups/Organizations

1. Navigate to sysadmin settings by clicking the hammer icon at the top of the page.



2. Click on the **Trash** tab. This page will list all the currently deleted datasets, groups, and

The screenshot shows the ACEP web application interface. At the top, there is a navigation bar with links for Datasets, Organizations, Groups, About, FAQ, and a search bar. Below the navigation bar, there is a header for the 'Trash' section. The 'Trash' tab is highlighted with a red circle. On the left, there is a sidebar with sections for Deleted datasets, Deleted organizations, and Deleted groups. Each section has a list of items and a 'Purge' button. At the bottom right of the trash section, there is a large red 'Purge all' button.

organizations.

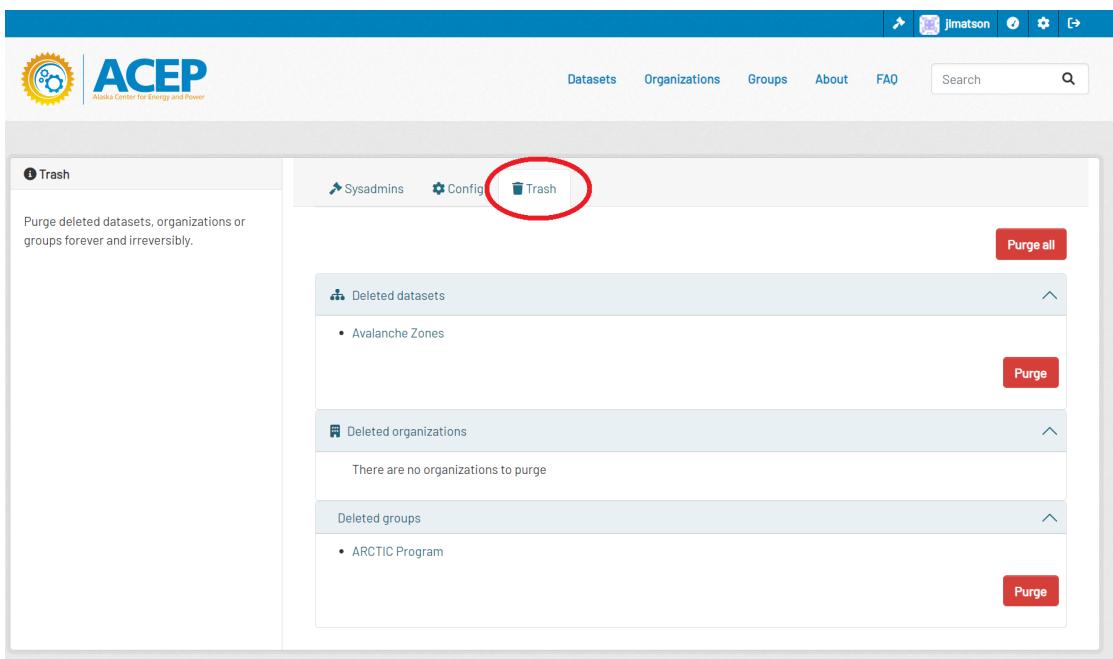
3. You can purge all deleted items using the **Purge all** button at the top of the form. To purge only one group of items (datasets, group, or organizations) use the **Purge** button in that section.

## Restore Deleted Datasets

1. Navigate to sysadmin settings by clicking the hammer icon at the top of the page.

The screenshot shows the ACEP web application interface. At the top, there is a navigation bar with links for Datasets, Organizations, Groups, About, FAQ, and a search bar. Below the navigation bar, there is a header for the 'Sysadmins' section. A red circle highlights the hammer icon in the top-left corner of the header, which is used to access sysadmin settings. The rest of the page content is mostly obscured by a light gray overlay.

2. Click on the **Trash** tab. This page will list all the currently deleted datasets, groups, and organizations.



3. Click on the dataset you would like to restore, and click the **Manage** button in the upper right.

Visibility:

Private

State:

Deleted

Source:

<http://example.com/dataset.json>

4. In the metadata fields find the **State** field.
5. Change the **State** field from **Deleted** to **Active**, and then click the **Update Dataset** button at the bottom of the form.

## Restoring a Deleted User

When a user is deleted from the website, their information remains in the database with the `state` field set to `deleted`. To reactivate the user, you must set this field to `active` in the database.

1. Enter the `acep-db-cont` docker container

- `> docker exec -it acep-db-cont /bin/bash`

2. Access the postgres database

- `> psql -U postgres`

3. List all the databases and connect to the `ckandb` database.

- `> \l`
- `> \c ckandb`

4. List all the tables and list the columns of the user table.

- `> \d`
- `> \d user`

5. List all the users in the user table.

- `> SELECT * FROM public.user;`

6. Find the deleted user with the username [username].

- `> SELECT id, name, email, state FROM public.user WHERE name = '[username]';`

7. Update the user's state field.

- `> UPDATE public.user SET state = 'active' WHERE name = '[username]';`

8. Find the user again and ensure that the state field is set to active.

- `> SELECT id, name, email, state FROM public.user WHERE name = '[username]';`

# Metadata Walkthrough

A detailed walkthrough of the metadata fields available.

## Dataset Fields

\*required field

### Title\*

A descriptive title for the dataset.

### Description\*

A free-form description of the dataset.

### Keywords

Keywords that describe the dataset.

Standardized vocabulary TBD

### Contact Points\*

Individuals connected to the creation or curation of the dataset. An email address is not required. This includes the following roles:

- **Creator:** primarily responsible for the content of the dataset
- **Contributor:** made contributions to the dataset
- **Editor:** helped prepare the dataset for publication

**Publisher\***

The entity responsible for making the dataset available.

ACEP is the default value. If the publisher is another entity, or if you would like to enter a custom value for one of the fields, select Other and enter information into the provided text box.

**License\***

A legal document under which the dataset is made available.

**Organization\***

The organization the dataset falls under, either **ACEP Internal Use** or **ACEP Open Data**.

**Visibility**

The visibility of the dataset: **private** for internal use or under-review datasets and **public** for reviewed/published datasets.

**Landing Page**

A webpage where one can gain access to the dataset or more information about the dataset.

**Release Date\***

Enter the date of the dataset's entry to the data catalog.

**Modification Date\***

Enter the most recent date of change to the dataset entry.

**Version**

An indicator of the version of the dataset.

## **Version Notes**

A description of changes between the current and previous version of the dataset.

## **Identifier**

A unique identifier of the dataset.

## **Provenance**

The origin and history of the dataset.

Include the filename of a README or other narrative document that describes the origin of the data and the processing steps employed to get to this posted version. Optionally include changes of ownership, custody, or responsibility. If brief, this information can be included in this free text field instead of in a separate document.

More info: <https://www.go-fair.org/fair-principles/r1-2-metadata-associated-detailed-provenance/>

## **Type**

The nature or genre of the dataset.

## **Temporal Coverage**

The temporal period(s) covered by the dataset. Enter the begin and end dates for the temporal period(s) covered by the dataset. If there is a period of missing or suspect data in the dataset, you can enter it as a separate temporal coverage and make a comment in the Notes section to explain the issue.

Warning: If there is no temporal coverage to enter, the field must be deleted.

## **Spatial Coverage**

The geographical area(s) covered by the dataset. These values should be entered in **GeoJSON format**.

Examples:

## **Geometry**

```
{
  "type": "Polygon",
  "coordinates": [
    [
      [
        [100.0, 0.0],
        [101.0, 0.0],
        [101.0, 1.0],
        [100.0, 1.0],
        [100.0, 0.0]
      ]
    ]
  }
}

{
  "type": "LineString",
  "coordinates": [
    [-101.744384, 39.321550],
    [-101.552124, 39.330048],
    [-101.403808, 39.330048],
    [-101.332703, 39.364032],
    [-101.041259, 39.368279],
    [-100.976562, 39.305091],
    [-100.914062, 39.245016],
    [-100.843658, 39.164141]
  ]
}
```

## Bounding Box

```
{
  "type": "Polygon",
  "coordinates": [
    [
      [
        [-80.190, 25.774],
        [-66.118, 18.466],
        [-64.757, 32.321],
        [-80.190, 25.774]
      ]
    ]
  }
}
```

## Centroid

```
{  
  "type": "Point",  
  "coordinates": [-72.323, 25.354]  
}
```

## **Language**

The language(s) of the dataset.

## **Documentation**

A page or document about the dataset.

## **Conforms to**

An established standard to which the dataset conforms.

## **Is referenced by**

A related resource that references, cited, or otherwise points to the dataset.

## **URI**

## **Resource Fields**

### **Name\***

A descriptive name for the resource.

### **Description\***

A free-form description of the resource.

### **Format**

The file format of the resource. This can be detected automatically.

### **GitHub Repo**

If the resource is a link to a GitHub repository, select True. This will allow the addition of a GitHub repository view that displays the repository's statistics.

### **Status**

The current stage of the resource in the workflow.

### **License**

A legal document under which the resource is made available.

### **Access URL**

A URL where one can access the resource.

### **Download URL**

A URL where one can download the resource.

### **Release Date\***

The date of formal issuance/publication of the resource.

### **Modification Date\***

The most recent date of change/update/modification to the resource entry.

### **Language**

The language(s) of the metadata or textual values of the resource.

### **Documentation**

A page or document about the resource.

## **Conforms to**

An established standard to which the resource conforms.

## **Access Services**

A data service that gives access to the resource.

## **URI**

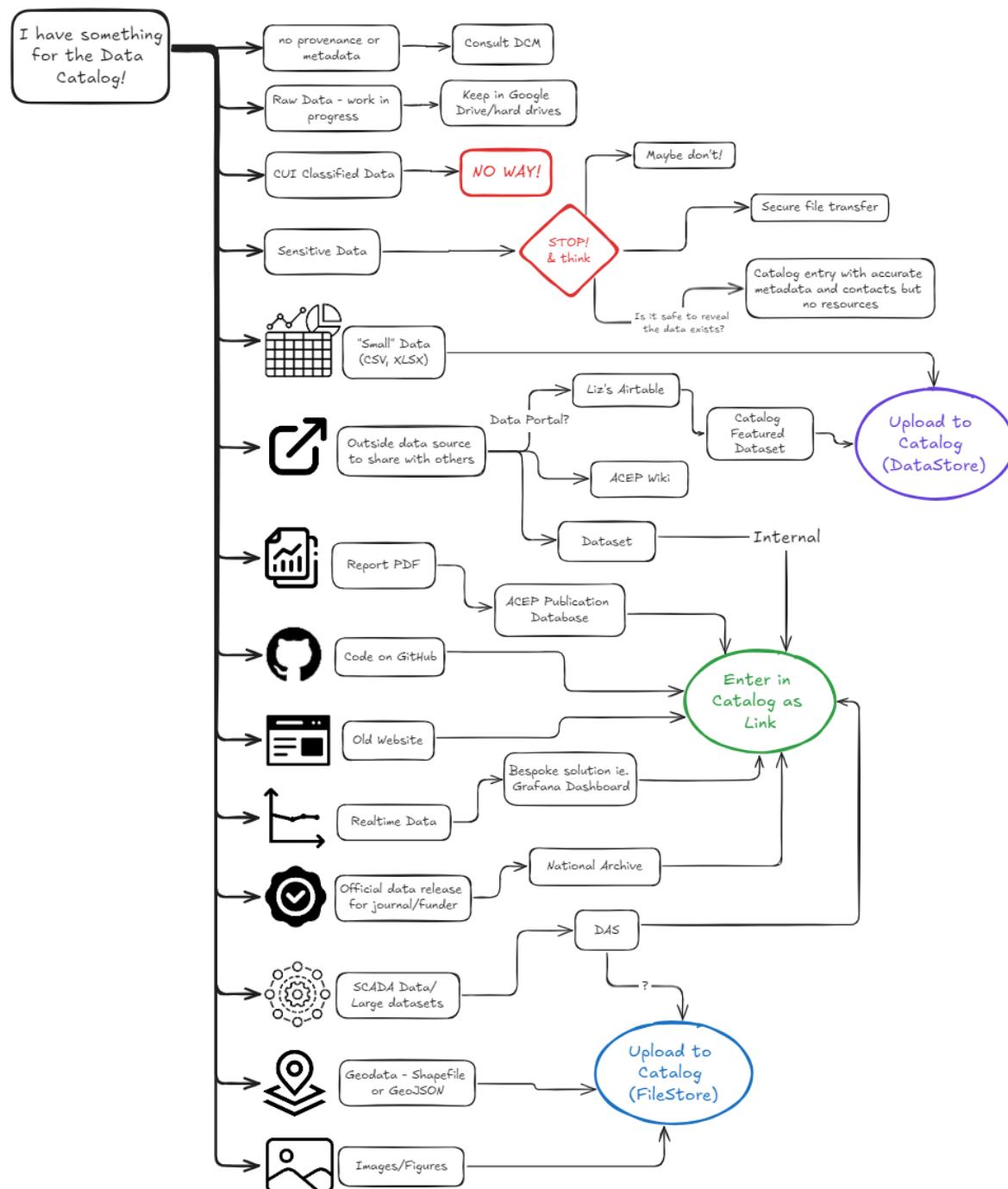
# For Developers

Documentation of the development of the ACEP Data Catalog.

GitHub repository for the main Data Catalog site: <https://github.com/UAF-RCS/acepportal-ckan>

For more information and guides, visit the official [CKAN Documentation](#)

## Data Sources Overview



## Developing the Data Catalog

The ACEP Data Catalog is run on a VM hosted by RCS. Extensions can be updated by pushing to the aceportal-ckan GitHub repository. After pushing, changes take ~30 min to update on the main site.

### Basic Docker Commands

List all running containers:

- `docker ps -a`

There are 5 containers that run the data catalog

- `acep-ckan-cont`
- `acep-db-cont`
- `acep-redis-cont`
- `acep-solr-cont`
- `acep-datapusher-cont`

### Spin up Application

- `docker compose up`

This will turn the terminal into an output stream for the docker containers.

TIP: I recommend keeping two terminals open: one for the output stream so you can see errors, and another to run other commands in

### Rebuild and spin up containers:

- `docker compose up -d --build`

Run this command after installing a new extension and adding it to the `.env` file.

### Go into container:

- `docker exec -it [container_name] /bin/bash`

Or if bash is not installed in the container:

- `docker exec -it [container_name] /bin/sh`

Or if in a bash terminal:

- `docker exec -it [container_name] bash`

### **Restart a container:**

- `docker restart [container_name]`

Restart the `acep-ckan-cont` container after making changes to non-HTML files. Changes in html files can be seen by refreshing the webpage.

### **Take Down Application**

- `docker compose down`

### **Clean up Project**

- `docker compose down --rmi all -v --remove-orphans`

This removes all containers, images, and volumes associated with a project. **Only do this if you want to clean up your environment and reset the containers.**

## **Creating a Local Instance**

Creating a local version of the data catalog is a useful tool for developing and testing new features.

1. Install Docker: <https://www.docker.com/get-started/>
2. Clone the ACEP CKAN repository from Github: <https://github.com/UAF-RCS/acepportal-ckan.git>
3. Create the `.env` file inside the main `acepportal-ckan` folder. Copy the contents from the `.env.example` file.
4. Specify the location of the source files, storage files, backups, etc. in the `.env` file. You will move those files to these locations in the next steps. For example:

```
# CKAN Mounts Directory
CKAN_EXTENSIONS_MOUNT=./ckan-extension
SRC_EXTENSIONS_PATH=/srv/app/src_extensions
CKAN_SOURCE_MOUNT=./ckan-src/src
CKAN_STORAGE_MOUNT=./ckan-src/storage
CKAN_INI_MOUNT=./ckan-src/ckan.ini
```

5. To create a replica of the current main Data Catalog, copy over the source files, storage files, `ckan.ini` file, and database backups from the VM. These files are located on the VM inside `/opt/ckan/backups`. Use `scp` to copy the files onto your machine. These backups are created everyday: replace `[date]` with the most recent date in the format `yyyymmdd`.

Inside of `acepportal-ckan/ckan-src` run the following

- `scp user@portal.lab.acep.uaf.edu:/opt/ckan/backups/app_[date].tar.bz2`
- `scp user@portal.lab.acep.uaf.edu:/opt/ckan/backups/app_storage_[date].tar.bz2`
- `scp user@portal.lab.acep.uaf.edu:/opt/ckan/acepportal-ckan/ckan-src/ckan.ini`

6. Use tar to decompress the source and storage tar files

- `tar -jxvf app_[date].tar.bz2`
- `tar -jxvf app_storage_[date].tar.bz2`

Decompressing the `app_storage` tar file should create a folder called `ckan` containing the folders `resources`, `storage`, and `webassets`. Rename the `ckan` folder to `storage`. This should result in the directory structure specified in `ckan-src/README.txt`

7. Create a backups folder alongside the `acepportal-ckan` repository on your machine. Specify the name in the `BACKUP_TO` setting in the `.env` file.

```
# Backups
BACKUP_TO=.../.../[backups folder name]
```

8. Run the following commands inside the backups folder to copy over the database and datastore.

- `scp user@portal.lab.acep.uaf.edu:/opt/ckan/backups/ckandb_[date].tar`
- `scp user@portal.lab.acep.uaf.edu:/opt/ckan/backups/datastore_[date].tar`
- `scp user@portal.lab.acep.uaf.edu:/opt/ckan/backups/resource_[date].tar`

9. Inside of the `ckan.ini` file, set the `ckan.site_url` setting to the localhost url as so:

```
ckan.site_url = http://127.0.0.1:5000
```

10. Build the containers using,

- `docker compose up`

11. Once the containers are up, use the `import_database.sh` bash script to import the database.

- `bash import_database.sh`

12. Rebuild the CKAN search index.

- `docker exec -it acep-ckan-cont /bin/bash`
- `cd /srv/app`
- `ckan search-index rebuild`

## Create a New Extension

1. Enter the acep-ckan-cont Docker container
  - `docker exec -it acep-ckan-cont /bin/bash` and run the following command
  - `ckan generate extension -o /srv/app/src/ckan-extension` This will create an extension in the `ckan-extension` folder which can be edited outside of the container.
2. Add the extension name to the `CKAN_PLUGINS` list in the `.env` file.
3. Run `docker compose up -d --build ckan`

## Install an Extension

1. Ensure that the extension supports CKAN 2.10.4 and Python 3.10. Clone the extension repository into the `ckan-extension` folder.
2. Ensure that all dependencies for the extension are listed in `requirements.txt` or a similar file.
3. Add the extension name to the `CKAN_PLUGINS` list in the `.env` file.
4. Run `docker compose up -d --build ckan`

## Updating the Main Site

To add a feature from your local instance to the main Data Catalog,

1. Push the files to the `acepportal-ckan` GitHub repository.
  - If you are adding a new extension, you must **delete the .git folder** before adding/committing/pushing.
2. Wait about 30 min. for the changes to be pulled to VM.
3. If you have added a new extension, SSH into the VM and add the extension name to the `.env` file.
  - `ssh user@portal.lab.acep.uaf.edu`
  - `cd /opt/ckan/acepportal-ckan`
  - `vi .env`
4. After installing new extensions or making other changes, you may need to restart the `acep-ckan-cont` container to make them take effect. Inside the VM, run
  - `docker restart acep-ckan-cont`

## Extensions

### Currently Installed

<https://github.com/UAF-RCS/aceportal-ckan/tree/main/ckan-extension>

#### **ckanext-customtheme**

**Author:** Jenae Matson

**Purpose:** Add custom theming and features for the CKAN instance, including

- ACEP logos, colors, and fonts
- Home page layout, images, and featured dataset
- Changed font weight of Register button
- Added tags to search page display
- HTML file for About page text
- Removed social media links from dataset/resources pages
- Added support contact info to dataset sidebar
- Added default blank option to add-to-group dropdown menu

**Configuration Settings:**

- `ckan.customtheme.featured_dataset = alaska-energy-inventory`

#### **ckanext-dcat**

**Link:** <https://github.com/ckan/ckanext-dcat>

**Purpose:** Rework metadata to conform to DCAT standard.

**Configuration Settings:**

- `ckanext.dcat.rdf.profiles = euro_dcat_ap_3`

**Modifications:**

- The file `schemas/acep_dcat_fields.yaml` was created to define the metadata fields for the catalog.
- The file `templates/scheming/form_snippets/publisher.html` was created to define the dynamic dropdown menu in the Publisher metadata field.

## **ckanext-faqpage**

**Author:** Jenae Matson

**Purpose:** Create an FAQ page linked in the masthead with collapsible boxes for questions and answers.

## **ckanext-geoview**

**Link:** <https://github.com/ckan/ckanext-geoview>

**Purpose:** Created resource views for geojson and other geo-data file types. We have implemented the OpenLayers Viewer.

**Configuration Settings:**

- ckanext.geoview.ol\_viewer.default\_feature\_hoveron = true

## **ckanext-githubrepopreview**

**Link:** <https://github.com/DataShades/ckanext-githubrepopreview>

**Purpose:** Provide a view for GitHub repository resources.

**Modifications:** This extension was created for an older version of CKAN, so the following changes were made to make it work with version 2.10:

- In the file `plugin.py`, replace the line `from lib import parse` with the following

```
from urllib.parse import urlparse

def parse(input_url, some_flag):
    parsed_info = {}
    parsed_url = urlparse(input_url)
    domain = parsed_url.netloc
    path_parts = parsed_url.path.strip('/').split('/')

    parsed_info['domain'] = domain
    parsed_info['owner'] = path_parts[0] if len(path_parts) > 0 else None
    parsed_info['repo'] = path_parts[1] if len(path_parts) > 1 else None

    return parsed_info
```

- In the file `templates/githubrepo.html`, delete the following lines

```

{%- block styles %}
    {% resource g.main_css[6:] %}

{%- endblock %}

{%- block scripts %}
    {% resource 'base/main' %}
    {% resource 'base/ckan' %}
    {% if g.tracking_enabled %}
        {% resource 'base/tracking.js' %}
    {% endif %}
{%- endblock -%}

```

### **ckanext-ldap**

**Link:** <https://github.com/NaturalHistoryMuseum/ckanext-ldap>

**Purpose:** Allows users to login using their UA credentials (a temporary solution while the official SSO is being implemented).

**Configuration Settings:**

- ckanext.ldap.uri = ldaps://auth.alaska.edu
- ckanext.ldap.auth.password = s3arch@ccount!
- ckanext.ldap.base\_dn = ou=userAccounts,dc=ua,dc=ad,dc=alaska,dc=edu
- ckanext.ldap.search.filter = (sAMAccountName={login})
- ckanext.ldap.auth.dn = cn=rcs-ad-read,ou=RCS,ou=UAF,dc=ua,dc=ad,dc=alaska,dc=edu
- ckanext.ldap.username = sAMAccountName
- ckanext.ldap.fullname = displayName
- ckanext.ldap.email = mail
- ckanext.ldap.ckan\_fallback = True

**Modifications:** Theng from RCS has made some modifications, including improving user creation to handle existing accounts: <https://github.com/UAF-RCS/aceportal-ckan/tree/main/ckan-extension/ckanext-ldap>.

## ckanext-package-group-permissions

**Link:** <https://github.com/salsadigitalauorg/ckanext-package-group-permissions>

**Purpose:** Allows all editors and admins to add datasets to any group, without having to be added as members to each group.

Note: This extension has now been updated by the creators. The following modifications are no longer necessary.

**Modifications:** This extension was created and works with CKAN 2.9. This instance is version 2.10, so the extension requires some small modifications to work. The following changes were made to the original extension:

- In the file `plugin.py`, change the `member_create` function to the following

```
def member_create(self, next_auth, context, data_dict):  
    """  
    This code is largely borrowed from /src/ckan/ckan/logic/auth/create.py  
    With a modification to allow users to add datasets to any group  
    :param context:  
    :param data_dict:  
    :return:  
    """  
  
    group = logic_auth.get_group_object(context, data_dict)  
  
    authorized = False  
    if not group.is_organization and data_dict.get('object_type') == 'package':  
        authorized = helpers.user_has_admin_access(include_editor_access=True)  
  
    if not authorized:  
        # Fallback to the default CKAN behaviour  
        return next_auth(context, data_dict)  
    else:  
        return {'success': True}
```

- In the the file `templates/package/group_list.html`, add the line `{ h.csrf_input() }` to the beginning of the two post forms, as follows

```
{% if groups %}  
<form class="add-to-group" method="post">  
    {{ h.csrf_input() }}  
    ...  
</form>  
{% endif %}
```

```

{% if c.pkg_dict.groups %}
<form method="post">
    {{ h.csrf_input() }}
    ...
{% endif %}

```

## **ckanext-pdfview**

**Link:** <https://github.com/ckan/ckanext-pdfview>

**Purpose:** Provide a view for pdf resources.

## **ckanext-restrictpublish**

**Author:** Jenae Matson

**Purpose:** Restrict the ability to change the visibility of a dataset to admins only. Datasets posted by editors default to private.

## **ckanext-scheming**

**Link:** <https://github.com/ckan/ckanext-scheming>

**Purpose:** Allows for the creation of alternate metadata templates (schemas) defined by .yaml or .json files.

**Configuration Settings:**

- scheming.dataset\_schemas ckanext.dcat.schemas:acep\_dcat\_fields.yaml
- scheming.presets = ckanext.scheming:presets.json ckanext.dcat.schemas:presets.yaml
- scheming.dataset\_fallback = false

**Modifications:** Some of the automatically calculated resource fields were manually re-added to be displayed. In the file `templates/scheming/package/resource_read.html`, below `{%- block resource_license -%}` add the following

```

{%- block resource_size -%}
<tr>
    <th scope="row">{{ _('Size') }}</th>
    <td>{{ res.size or _('unknown') }} bytes</td>
</tr>
{%- endblock -%}

```

```

{%- block resource_datastore -%}
<tr>
    <th scope="row">{{ _('Datastore active') }}</th>
    <td>{{ res.datastore_active or _('unknown') }}</td>
</tr>
{%- endblock -%}

```

## **ckanext-xloader**

**Link:** <https://github.com/ckan/ckanext-xloader>

**Purpose:** Improve data uploading, including increasing allowed file sizes.

**Configuration Settings:**

- ckanext.xloader.jobs\_db.uri=postgresql://ckandbuser:ckandbpassword@db/ckandb

## **Adding Alternate Schemas with ckanext-scheming**

1. Create a .yaml or .json file in the folder `ckanext-scheming/ckanext/scheming` to define the metadata schema. See extension documentation for more information and examples.
2. In `ckan.ini`, add your schema(s) to the `scheming.dataset_schemas` config option. For example:

```
scheming.dataset_schemas = ckanext.scheming:arctic_dataset.json ckanext.scheming:geo_dataset
```

3. The new dataset creation form is located at a url defined by the schema type name. For example, the creation form for datasets of type `arctic-dataset` is located at `/arctic-dataset/new`. You can define a new Add Dataset button using this new url.

## **Attempted Extensions**

### **ckanext-spatial**

**Link:** <https://github.com/ckan/ckanext-spatial>

**Purpose:** This extension adds the ability to search for datasets on a map widget, as well as a dataset extent map widget on the dataset page, provided correct geospatial metadata.

**Problems:** This extension is not currently installed due to the following,

- Configuring map tiles for ckanext-spatial caused the map tiles for ckanext-geoview to disappear.
- Datasets with the required spatial metadata were not searchable on the map search widget, although the dataset extent widet worked correctly.

### **ckanext-oidc-pkce**

**Link:** <https://github.com/DataShades/ckanext-oidc-pkce/tree/master>

**Purpose:** This extension allows for users to be authenticated through an external application when they login.

**Problems:** Ideally users on the ACEP Data Catalog would be able to login using their UA login credentials through Google Authentication. This extension installs correctly, but does not seem to support Google Authentication.