NRC OpenLCA Collaboration Server: Reference Manual

Volume: 1
Report No.: 1
Date: May, 2022
Authors: Portia Masibi, Miyuru Kannangara

Energy, Mining and Environment Research Center

National Research Council Canada
Conseil national de recherches Canada

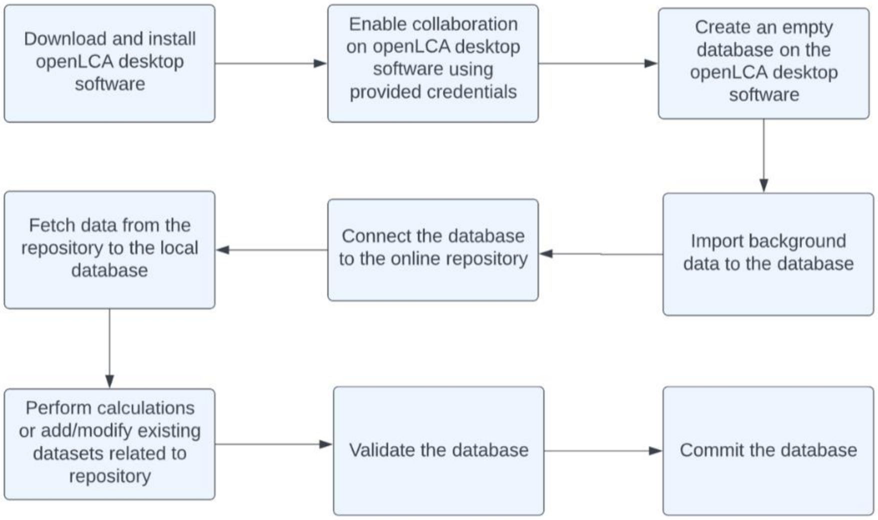
Document Change Log

Revision	Changes	Author	Approver	Release Date
0.1		Portia Masibi	Miyuru Kannangara	25-05-2022
1.0		Portia Masibi	Miyuru Kannangara	19-07-2022
1.1		Miyuru Kannangara	Miyuru Kannangara	01-11-2022

Executive Summary

NRC develops and maintains life cycle inventory (LCI) databases that can be used in life cycle assessment in various sectors. In order to facilitate sharing, access and collaborative development of LCI data, NRC has implemented openLCA collaboration server (CS) to host its LCI databases. This document outlines the access and use of NRC openLCA CS. It assumes the readers are already familiar with the general use of openLCA software.

Main steps that need to be followed for accessing and using the NRC openLCA CS are explained below. First, users should request access credentials that can obtained from NRC administrator. Users are provided with the varying level of access privileges based on specific needs and requirements.


Download and install openLCA desktop software
Enable collaboration on openLCA desktop software using provided credentials
Create an empty database on the openLCA desktop software
Import background data to the database
Connect the database to the online repository
Fetch data from the repository to the local database
Perform calculations or add/modify existing datasets related to repository
Validate the database
Commit the database

In CS, LCI data are organized as repositories. Users can directly download the LCI data or connect their local openLCA software to the CS. In the latter case, users need to configure the openLCA software installed in their respective computers to access NRC openLCA CS. Also, the local openLCA database of each user need to be configured and connected with relevant online repositories for collaborative development of LCIs. The LCI download, modification and upload using CS follows a linear workflow to avoid inconsistencies. This manual provides guidance for accessing, configuring and using NRC openLCA CS in detail.

Table of Contents

List of Figures

1. Introduction

With the general advancement of Life Cycle Assessments (LCA), collaborative work on LCA studies has become increasingly common. Collaboration can be in the same team, organisation, or even international teams. Currently, NRC uses openLCA software platform (can be downloaded at https://www.openlca.org/) to conduct LCA and maintain NRC's life cycle inventory (LCI) databases. In order to facilitate remote collaboration in LCA studies, NRC is using openLCA Collaboration Server (openLCA CS) platform. This document serves as a guide for setting up and using NRC openLCA Collaboration Server for users.

OpenLCA Collaboration Server is a server application that complements openLCA application and was developed to facilitate exchange and synchronisation of LCA data (e.g. flows processes, product systems or entire LCA models) between different users who work from different computers, enabling distributed, collaborative LCA modelling. OpenLCA users can work simultaneously on an LCA study while tracking each other's changes along synchronised databases.

The different cases where the openLCA Collaboration Server is important are:

- Teams working on the same LCA models, wanting to synchronise data.
- A single user working on an LCA model from different devices, and wanting to synchronise data.
- A personal documentation backup system with change control where one can comment blocks of changes to document your work.
- Sharing of data sets, LCA models and entire databases and their updates without losing a database's integrity and enabling the track of minor changes.
- Publication of data, where others can view or download publicly available data on the LCA Collaboration Sever an d import them directly into openLCA without registration.
- Creation of consistent and verified databases with the option to check previous versions.

2. Accessing NRC openLCA collaboration server

The openLCA Collaboration Server (openLCA CS) is an open-source tool that is available for free. It has been configured and implemented on an NRC server. This allows users to use it directly from the website or in the openLCA desktop application without having to install or maintain the servers. This section explains the steps needed to access the LCA CS either suing website or desktop software.

2.1 Website

Users can access the openLCA CS directly from the website without the need to download and install the openLCA application. However, users can only view and download datasets from the website. The steps to access it from the website are:

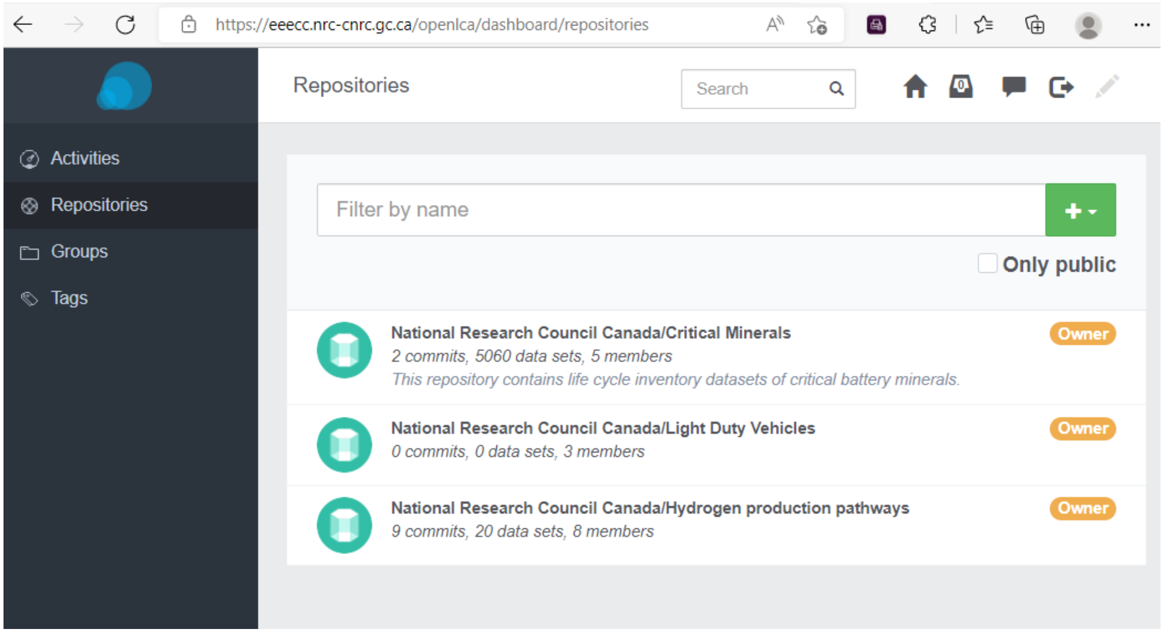
1. Firstly, credentials should be provided for users by NRC server administrator.
2. Users can then visit the NRC openLCA collaboration server directly at the url: https://eeecc.nrc-cnrc.gc.ca/openlca
3. The log in icon is at the top right corner. Log-in with given credentials.
4. Then use the password reset process to generate your own password your account. To do this, go to *Profile*, and fill in the *Password* and *Password (repeated)* section. Click on *Save*. This will save the new password.
5. For any technical challenges or questions on accessing the server, users can contact the server administrator at: Cyrille.Deces-Petit@nrc-cnrc.gc.ca

On the website, users can select and browse available repositories and download the repositories to use on their local openLCA desktop application. The access to relevant repositories are configured by the administrator as needed.

OpenLCA uses repositories and databases to store manage data and their difference is important to distinguish.

Repositories: Refers to online collections of LCA data that exists in openLCA CS. It consists of grouped datasets that mirror the local databases.

Databases: Refers to local databases of the openLCA desktop client that contain existing local LCA datasets. A part of the database or the complete database can be connected to online repositories.



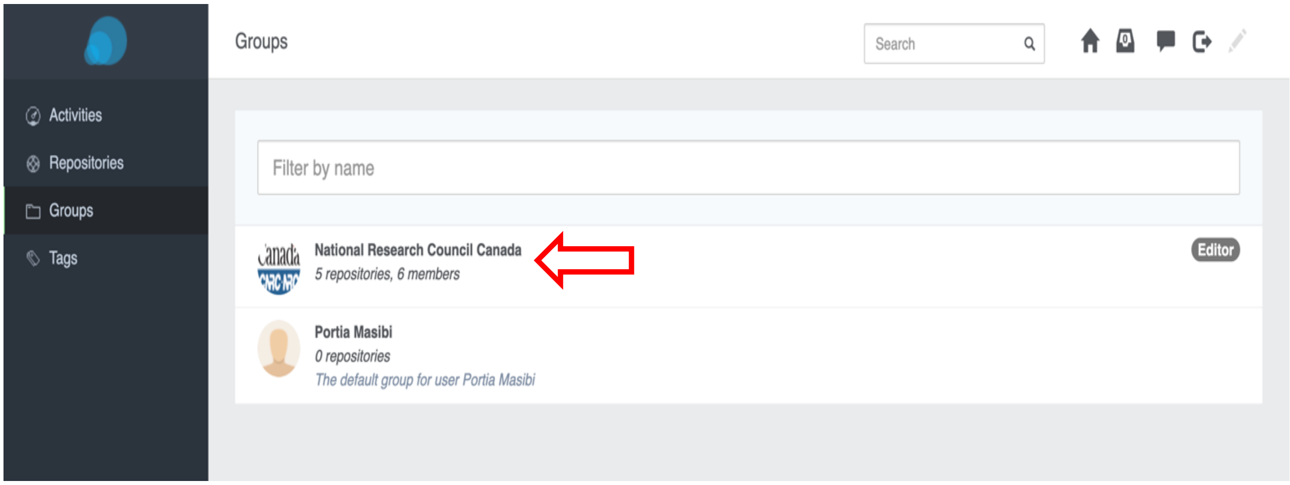
National Research Council Canada/Critical Minerals
2 commits, 5060 data sets, 5 members
This repository contains life cycle inventory datasets of critical battery minerals.
Owner

National Research Council Canada/Light Duty Vehicles
0 commits, 0 data sets, 3 members
Owner

National Research Council Canada/Hydrogen production pathways
9 commits, 20 data sets, 8 members
Owner

Figure 1: An example list of repositories available on the openLCA collaboration server

6. In addition to repositories, users should make sure they are added to the relevant groups that facilitates communication among users of a repository. This can be found under the Groups section (Figure 2).



National Research Council Canada
5 repositories, 6 members
Editor

Portia Masibi
0 repositories
The default group for user Portia Masibi

Figure 2: Access to National Research Council Canada Repositories

2.2 OpenLCA Desktop Client

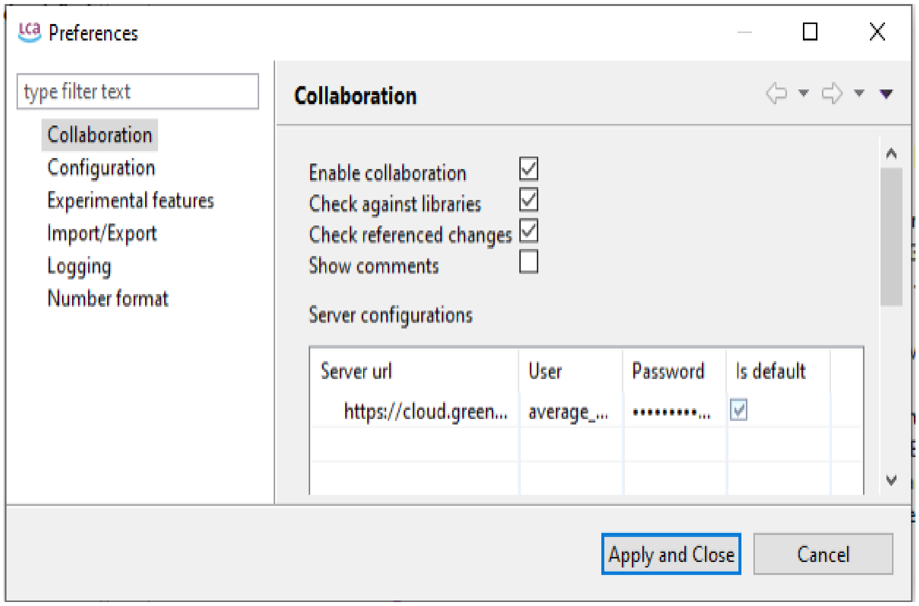
Using the openLCA desktop software is the recommended way of access the CS as it provides the user the ability to download LCI data, perform impact assessment calculations and modify the existing data and share the modification with the working group.

Configuration of desktop client

When accessing openLCA CS from the desktop, first a user needs the openLCA application on their desktop. The openLCA CS can then be accessed by linking it to a user's openLCA desktop application. This allows sharing and collaboration of with local datasets. The steps for this are:

1. First, the user must make sure they have access to the openLCA CS by following the steps in Section 2.1.
2. Download OpenLCA application from https://www/openlca.org/download/
3. In the top left corner of your local openLCA, click on *File* then *Settings/Preferences.*
4. A Preferences window should pop up (Figure 3), in this window check *Enable collaboration, Check against libraries* and *Check referenced changes.*
5. On the same window, right click on the server configuration table and click *create new.*
6. Under the Server URL use the NRC server https://eeecc.nrc-cnrc.gc.ca/openlca
7. Enter the username and password used for logging into the collaboration server. The username required can be found on the collaboration server website in the Profile/Account section.
8. Set this Server URL as *default* and select *Apply and Close.* You should now have access to the NRC collaboration server. Users can add multiple collaboration servers with their respective credentials.

If an error arises, check if you have not accidently included any extra spaces or similar in the server URL, password or user name sections. If the problem persists, contact the server administrator.



Enable collaboration ☑
Check against libraries ☑
Check referenced changes ☑
Show comments ☐

Server configurations

Server url	User	Password	Is default
https://cloud.green...	average_...	•••••••••...	☑

Figure 3: Setting up openLCA CS on desktop application

Check against libraries checks if the user can commit specific data to that library. Server admins and other high user roles can decide to choose certain data in the repository that should not be committed further.

Check referenced changes suggests new commits to the library that might otherwise cause an error. For example, when a user adds a new flow to an existing process, where the flow is not yet in the openLCA CS.

3. Setting-up the local database and connecting with the online repository

In the openLCA, a database is comprised of multiple categories of data including product systems, processes, flows, indicators and parameters and background data, as summarized below. A much more detailed explanation can be found in the openLCA user manual.

- Product Systems: Product systems are life cycle models of a product as in ISO 14040.
- Processes: A process is an activity that transforms an input into an output. This stores data which is for processes that can either be a unit process or system process.

- Flows: This data stores all flows, flow types and reference flow properties. The flow types could be a product, an elementary flow or a waste flow. It is also possible to see where the flows are used in different processes.
- Indicators and Parameters: This data gives us information on the impact assessment methods used, social indicators, global parameters and the data quality systems used.
- Background Data: In this section we find all the background data to our datasets, in flow properties we can have access to economic and technical flow properties, in unit groups we have conversions of units, currencies used, actors who are contributors to the model, sources of information and locations where this information was sourced from.

It is critical to organize the foreground data and background data in user's local database as shown in Figure 4. Typically, the LCI datasets provided in NRC CS repositories are gate-to-gate datasets, which require supporting background LCI datasets for cradle-to-gate or cradle-to-grave life cycle impact assessments (LCIA). In order to facilitate the connectivity with commercially and publically available LCI databases, NRC LCI data shares a common reference flow system with such databases. The reference flow system used is indicated in the documentation of each NRC LCI dataset.

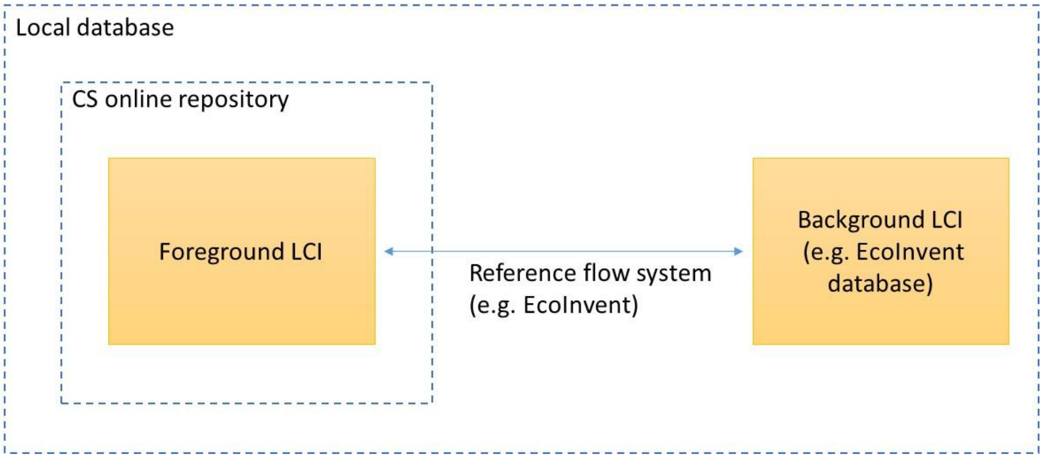


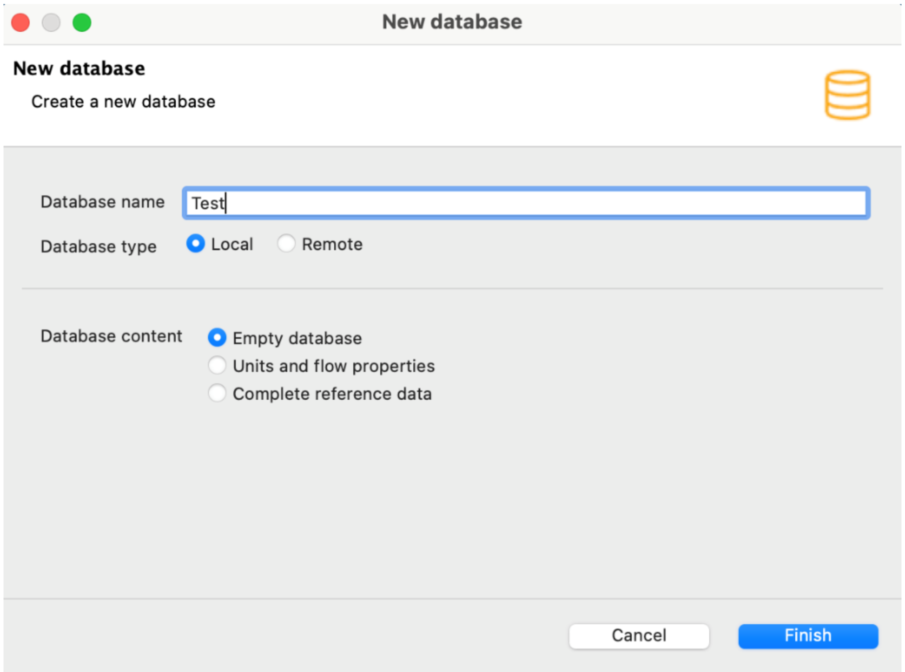
Figure 4: Organization of the LCI datasets in the local database.

Hence, each user needs to create a local database and import necessary background LCI datasets required for LCIA calculations.

3.1 Creating a local database

Creating and connecting local database is described below:

1. First, open your openLCA application.
2. Right click on the *Navigation* bar and Select *new database.*
3. A window pops up where we give the database a name, choose the type of the database and content of the database. For this example we named the database Test (Figure 5).
4. Click on *Local* for database type and *empty database* for the database content. After that select *Finish*.



Database name: Test

Database type: ● Local ○ Remote

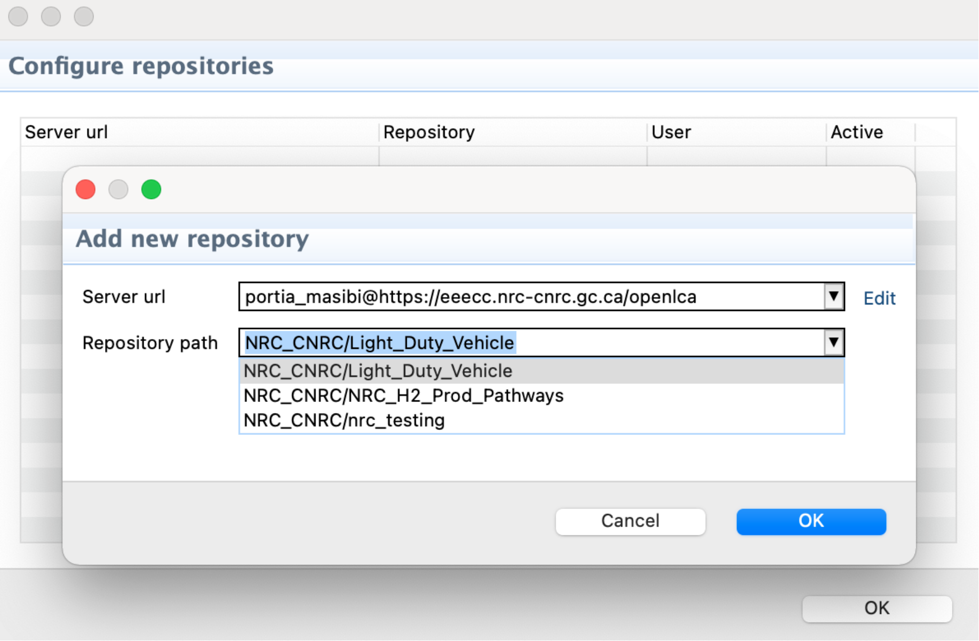
Database content:
● Empty database
○ Units and flow properties
○ Complete reference data

Figure 5: Creating a local empty database

A ***local database*** is located in the local computer while an ***online repository*** is located in a server.

This will create a database with processes, flows, indicators, and parameters. These will be further explained in Section 4. By right-clicking on the database, this gives a user option to either delete the database, rename it, copy it etc.

5. To connect to the repository, right click on the database, then click on *repository* and # *configure*.
6. Right click on the empty table and select *add to the repository*. If the openLCA CS is set up correctly, the NRC Server URL should show up along with all the Repository paths in the NRC server, this can be seen in Figure 6. Chose the repository you want to connect to the database and then click *OK.*

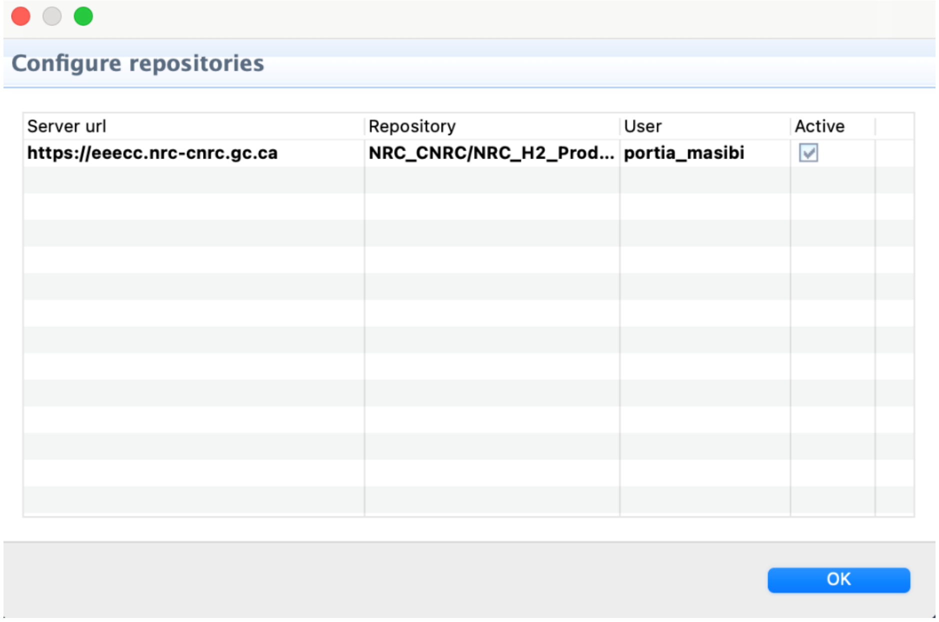


Server url: portia_masibi@https://eeecc.nrc-cnrc.gc.ca/openlca

Repository path: NRC_CNRC/Light_Duty_Vehicle

Figure 6: Configuring a local database to an online repository

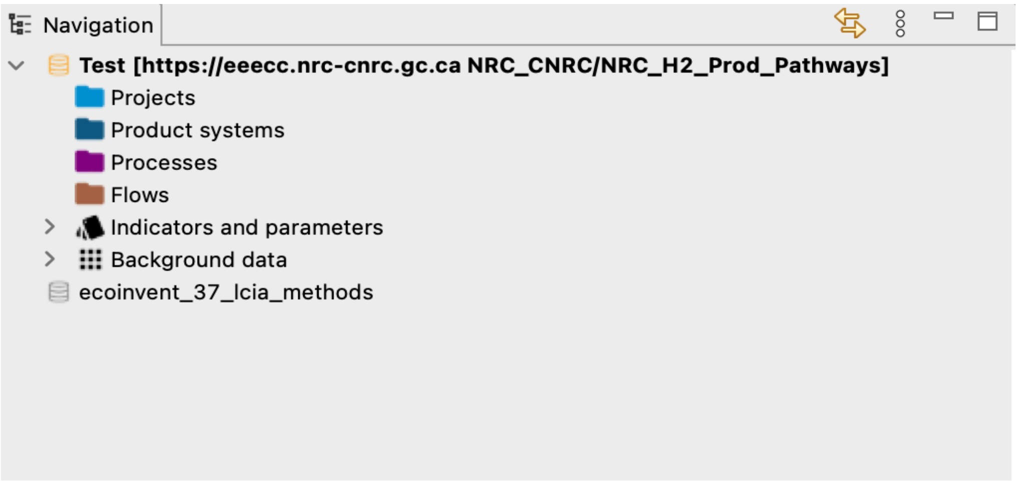
7. Set the database as *Active* and click on *OK.* Only one repository can be active at a time



Server url	Repository	User	Active
https://eeecc.nrc-cnrc.gc.ca	NRC_CNRC/NRC_H2_Prod...	portia_masibi	☑

Figure 7: Activating an online repository on a local desktop

8. OpenLCA will connect the repository to the database and rebuild the index of the selected database. A connected database looks like the one shown below on Figure 8 where the URL is shown next to the database name



- Test [https://eeecc.nrc-cnrc.gc.ca NRC_CNRC/NRC_H2_Prod_Pathways]
  - Projects
  - Product systems
  - Processes
  - Flows
  - Indicators and parameters
  - Background data
- ecoinvent_37_lcia_methods

Figure 8: How a local database connected to an online repository looks like

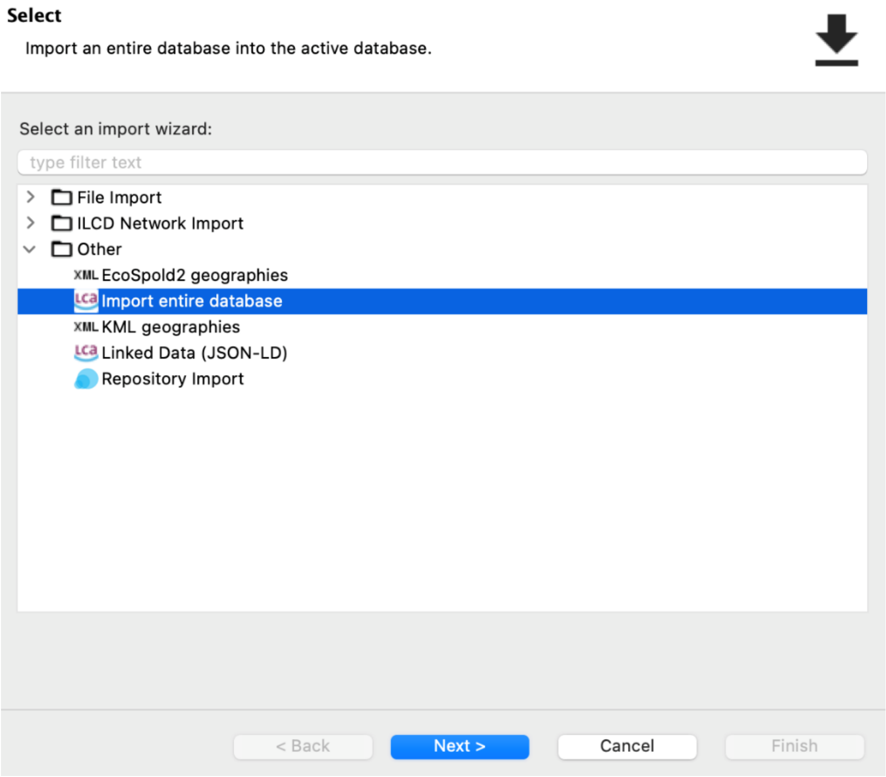
3.2 Importing background databases

Both commercial and free openLCA databases can be found at https://nexus.openlca.org/. When commercial databases such as EcoInvent have been used as background LCI for NRC repository, the users will be required to have their own subscriptions. Commercial background databases are not a part of NRC CS repositories.

Once background database is obtained, follow the steps below to import it into the local database created in section 3.1.

a. On your local computer right click on the database linked to the repository and select *import*. In this example, we used the Test database we created.

b. A window for selecting the database will pop up (Figure 9). Under *Other*, select *Import entire database*, ILCD or JSON-LD based on the format of the downloaded background database.



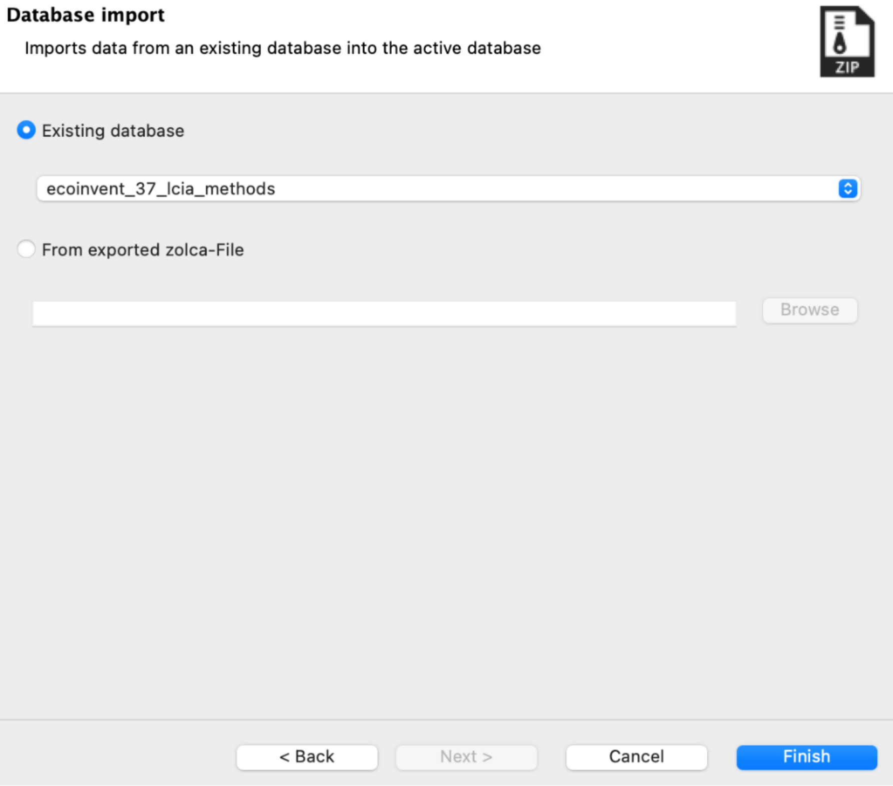
Select

Import an entire database into the active database.

Select an import wizard:

Figure 9: Importing an entire database into an active database

c. This will give the user two options (Figure 10). To export a database that exists on the openLCA application or from a user's personal computer (exported zolca-file). As we had already imported the database into openLCA. Click on *existing database* and select *Finish.*



Database import

Imports data from an existing database into the active database

● Existing database

ecoinvent_37_lcia_methods

○ From exported zolca-File

Figure 10: Importing an existing database on openLCA or exporting a file from computer

At this stage, the local database is complete with required background data and it is connected with the CS online repository.

<u>Note</u>

If the background database is large in size or a commercial proprietary database and no changes in background data is anticipated, it is not necessary to have the background database as a part of the repository. In that case, the repository needs to be **untracked** at this point before continuing to next step. Please refer section 5.1 (Data management) for further details.

4. Fetching and committing data

Fetching and committing represent download and upload from and to an online repository, respectively.

4.1 Fetching data

When connected to an openLCA CS, datasets can be fetched from a connected repository.

1. First, right-click on a local database
2. Select *Repository*, then *Fetch*. The figure for this steps is shown in Figure 11.

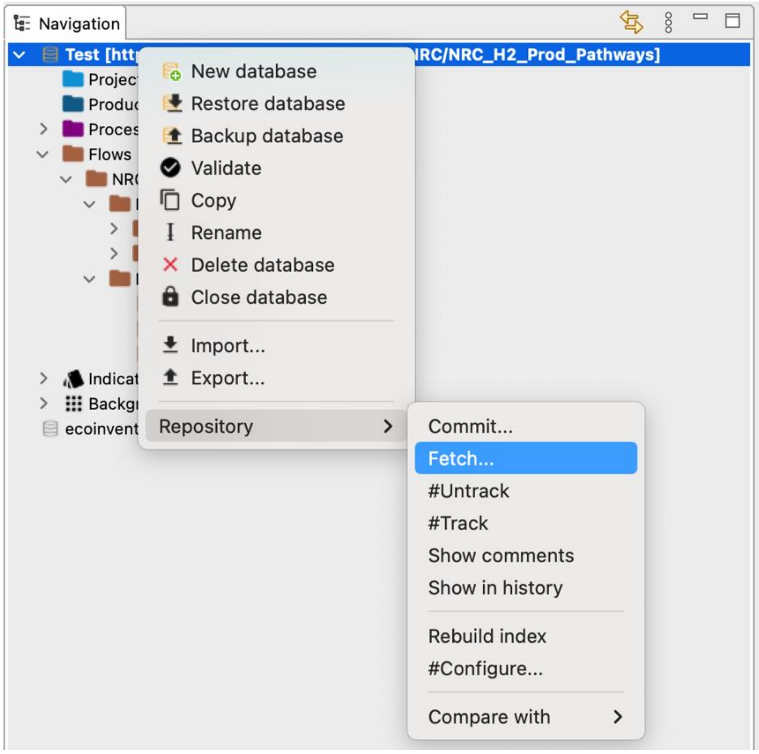
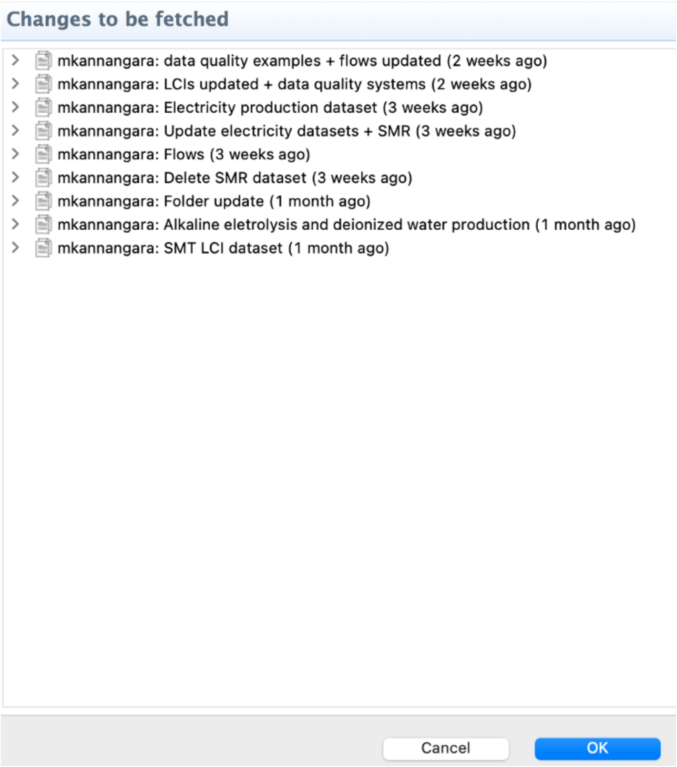


Figure 11: Fetching data into a local database

3. Two windows will pop up (Figure 12), the first one lists all changes everyone committed since the last fetch. Select OK on this window.



Changes to be fetched

- mkannangara: data quality examples + flows updated (2 weeks ago)
- mkannangara: LCIs updated + data quality systems (2 weeks ago)
- mkannangara: Electricity production dataset (3 weeks ago)
- mkannangara: Update electricity datasets + SMR (3 weeks ago)
- mkannangara: Flows (3 weeks ago)
- mkannangara: Delete SMR dataset (3 weeks ago)
- mkannangara: Folder update (1 month ago)
- mkannangara: Alkaline eletrolysis and deionized water production (1 month ago)
- mkannangara: SMT LCI dataset (1 month ago)

Figure 12: List of changes to be fetched since last commit

4. The other window lists all the commits since the last fetch in a merged and structured list. This is grouped into its respective data type. A user can choose to Discard local changes, overwrite remote changes, or accept the changes by selecting OK.

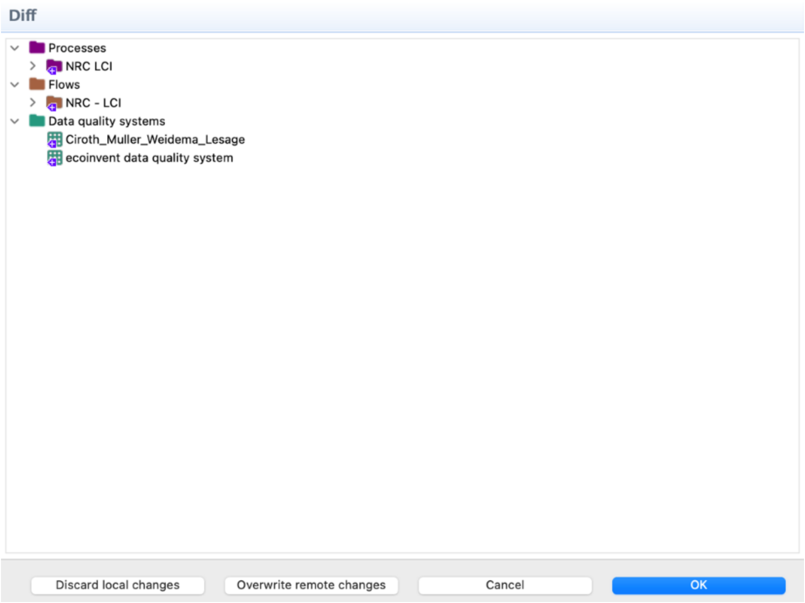


Figure 13: Commits to be fetched in each datasets. User can discard, overwrite or accept changes.

It is recommended to fetch regularly to be up to date, maintain clarity and transparency with teams working on the same datasets.

4.2 Committing data

When changes are made to existing datasets or new datasets are required to be added to the repository, a commit must be performed. To commit data sets to a repository, take the steps below.

1. Right click on the dataset name
2. Click on *Repository* and select *Commit*

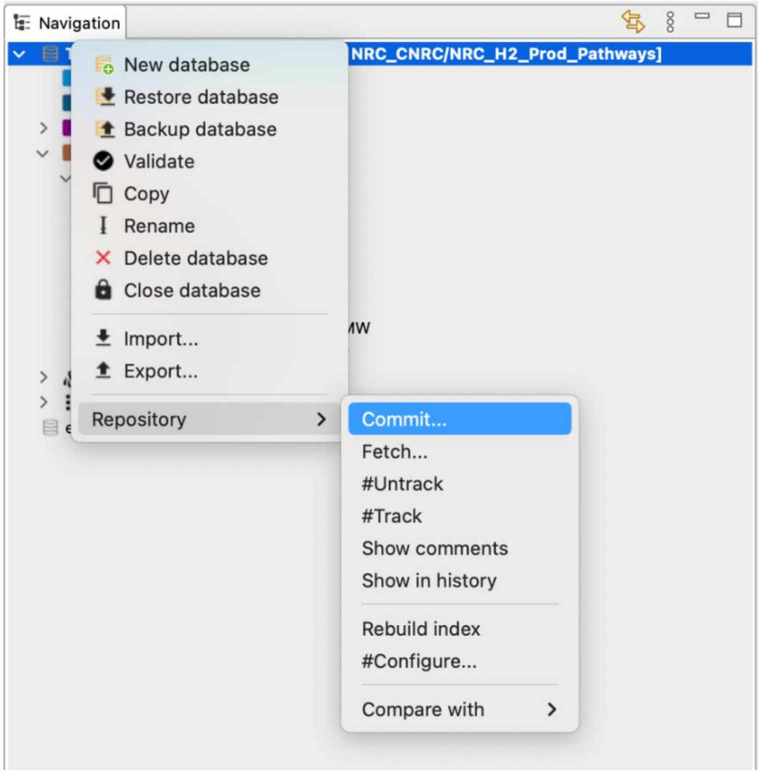
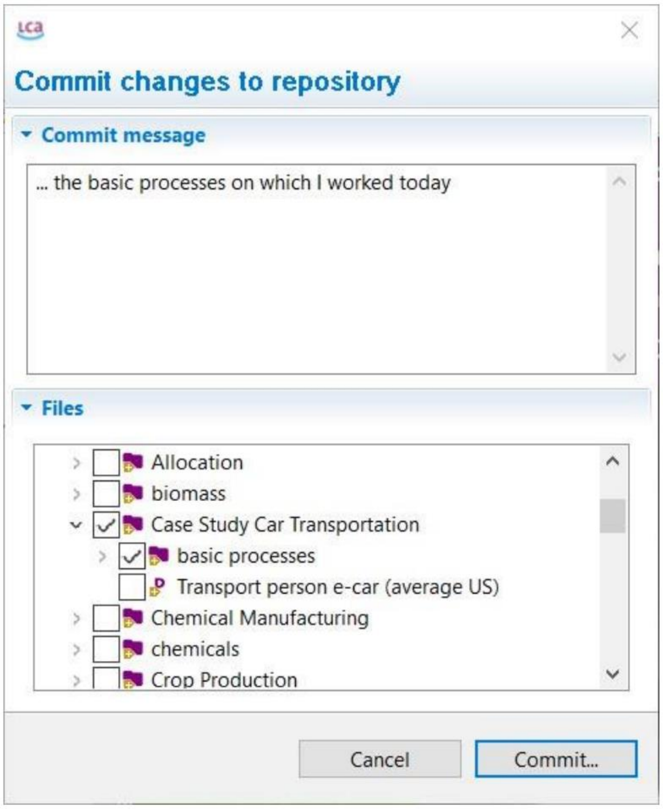


Figure 14: Committing data to an online repository from a local database

3. The user is then prompted to select which datasets are to be committed
4. Add a commit message to let your team know of the changes, this is mandatory in each commit (Figure 15).



... the basic processes on which I worked today

- ☐ Allocation
- ☐ biomass
- ☑ Case Study Car Transportation
  - ☑ basic processes
  - ☐ Transport person e-car (average US)
- ☐ Chemical Manufacturing
- ☐ chemicals
- ☐ Crop Production

Figure 15: Committing datasets to an online repository. Always include a commit message.

When committing to an empty repository for the first time or committing large datasets, this may take a while, approximately 15 minutes and openLCA may ‘not respond’ anymore. This is only temporary and may last for several minutes, the commit is still ongoing. It is also possible to select only specific datasets for a commit by right-click + CTRL. To maintain clarity and transparency, it is recommended to commit small datasets as this helps team members trace and understand changes.

4.3 Avoiding Conflicts when Fetching and Committing Data

When a team is working on the same repository, there can be different changes all at once. To avoid conflicts, first it is important to understand that the workflow of commits and fetches is always linear. A user cannot commit datasets to the server indiscriminately. In fact, before each commit, the user must fetch the current state of the repository from the openLCA CS. This is shown in Figure 16.

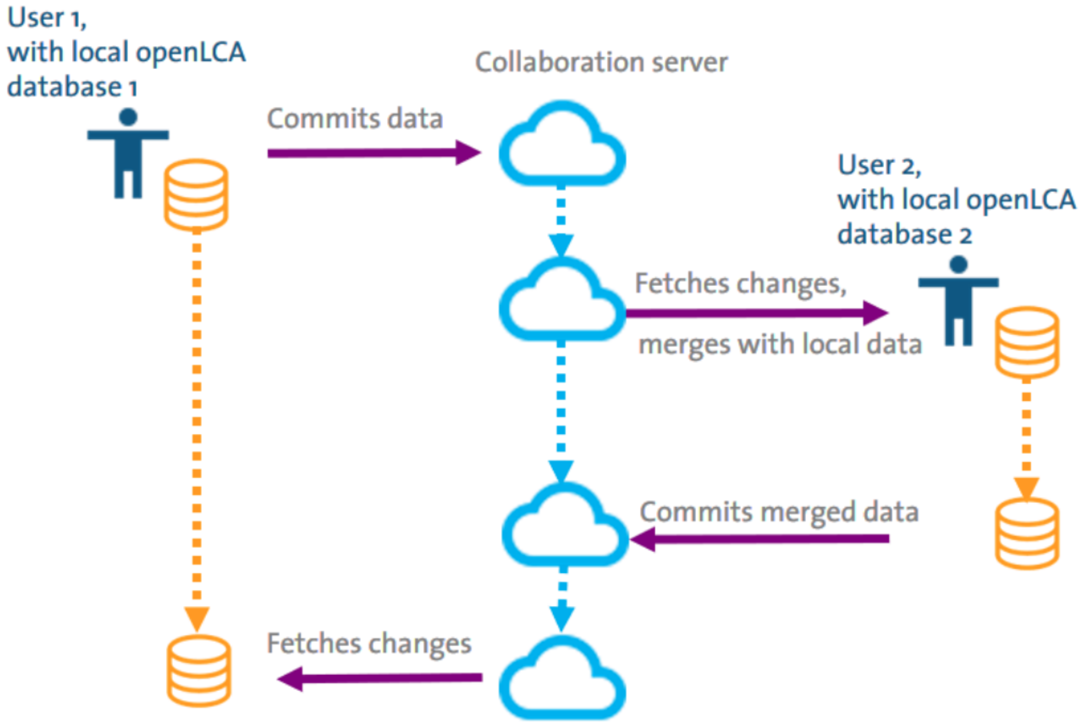


Figure 16: Linear workflow of the openLCA Collaboration Server

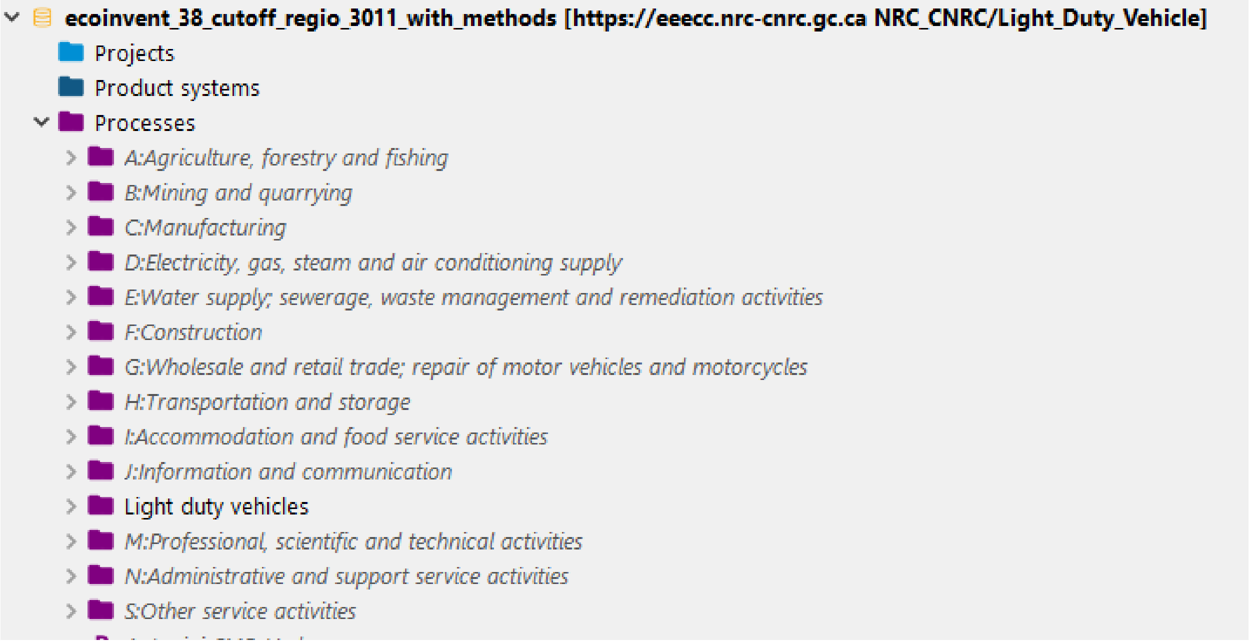
Conflicts however may occur if two or more users work on the same database item or interlinked database item at the same exact time. Therefore, to avoid this, the best solution is directly coordinating work within your team as well as to regularly commit and fetch data.

5. Data management

5.1 Working large background inventory databases

Background databases does not necessarily have to be a part of an online repository. When foreground and background systems are defined in LCA study, the same boundaries can be applied in managing the LCA data. Separating background datasets are especially useful when dealing with large and commercial databases such as EcoInvent as it improves commit/fetch times and respects the licence agreements.

In order to limit background datasets to the local database, follow the work flow up to importing the background datasets into database (described in section 3.2.). Then, right click on the database and select "Repository/Untrack”. This step stops OpenLCA CS from tracking the background datasets for any changes. Once, the untracking is complete, foreground datasets from repository can be fetched or new foreground datasets can be added to local database and be committed (described in section 4). The tracked and untracked sections of the database can be seen in OpenLCA desktop as shown in Figure 17.



ecoinvent_38_cutoff_regio_3011_with_methods [https://eeecc.nrc-cnrc.gc.ca NRC_CNRC/Light_Duty_Vehicle]
- Projects
- Product systems
- Processes
  - *A:Agriculture, forestry and fishing*
  - *B:Mining and quarrying*
  - *C:Manufacturing*
  - *D:Electricity, gas, steam and air conditioning supply*
  - *E:Water supply; sewerage, waste management and remediation activities*
  - *F:Construction*
  - *G:Wholesale and retail trade; repair of motor vehicles and motorcycles*
  - *H:Transportation and storage*
  - *I:Accommodation and food service activities*
  - *J:Information and communication*
  - Light duty vehicles
  - *M:Professional, scientific and technical activities*
  - *N:Administrative and support service activities*
  - *S:Other service activities*

Figure 17: Tracked and untracked (greyed out) sections of the database. In this repository, “Light duty vehicle” category is tracked as it is foreground data and rest of the EcoInvent database is not tracked.

5.2 Working with multiple repositories on the openLCA Collaboration Sever

An openLCA collaboration server may contains thousand repositories, each of them belonging to different working groups. Each repository can be made publicly available or kept private within their users. When developing a model, users would typically begin by importing dataset backup or fetching background datasets from collaboration servers. If using a dataset backup, it is important for the team working on the same model to use the same dataset backup version. Hence, it is preferred to install the dataset backup on the collaboration server, then each team member fetches the background dataset from the collaboration server.

When a model uses multiple repositories, users may decide to track or untrack particular items of their model in regard to the repository set as active. Since each repository have their own access configuration, public or restricted to specific users, it is advised that contributors track the changes (their own contribution and the contribution of their team member) towards only one repository. Background datasets fetched as reader from multiple repositories can be tracked or untracked (see figure below). It is advised to set as untracked for regular work, but periodically track and fetch the latest data when users are made aware of changes in the background datasets through the built-in messaging system of the collaboration server (email notifications).

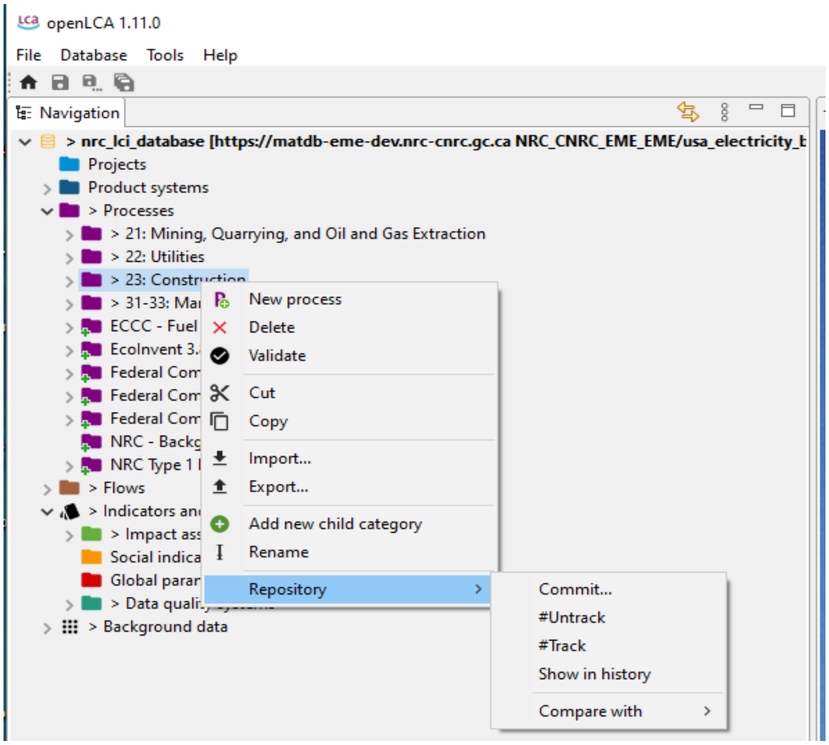


Figure 18: Tracking and Untracking particular items in the local database

5.3 Validating databases

Databases should be validated regularly to assure that interlinkages within a database are set correctly and identify whether all the linkages within the database are working. To validate databases right-click on the database and select *validate*.

If a validation fails, the missing interlinkages should be fixed before committing. Users should not commit databases when the validation has failed, they should instead discard the changes, fetch the repository again and redo the modelling steps. Validation may fail because of i) missing interlinked flows and ii) modified flow properties are not committed. These are explained further below.

Missing interlinked flow

When a user adds a flow to a local database and uses the flow as an input for a process to the repository but does not commit the underlying flow. The linked flow is missing in the repository and corrupts the database for other users.

Modified flow property

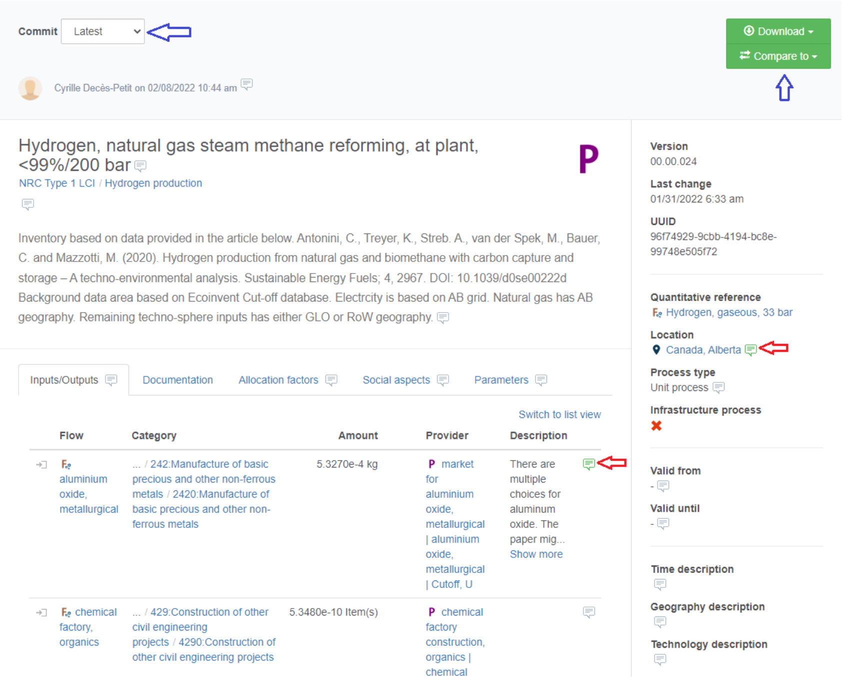
When a user changes the flow property of an existing flow in a local openLCA database and uses the flow with the updates property in a process. The user commits the process to the repository but not the flow with the modified flow property. The modification of the flow is missing in the repository and corrupts the data set in the repository for other users.

5.4 Version Control

Quality control

The collaboration server provides the features required to maintain a high level of data quality. As shown on Figure 17, a dataset is displayed along with essential tracking information, version number, date of last change, and the universal unique identifier of the dataset. Since the dataset can have multiple changes, quality control teams can switch amongst different versions committed to the server. They can also compare the displayed version with another version to track the differences.

Further, the strength of the collaboration server is to let the quality control teams comment on each field of the dataset as identified with a little comment icon on the right side of each field. As indicated with a red arrow, fields that have been commented have a green comment icon instead of a gray one.



Cyrille Decès-Petit on 02/08/2022 10:44 am

Hydrogen, natural gas steam methane reforming, at plant, <99%/200 bar

NRC Type 1 LCI / Hydrogen production

Inventory based on data provided in the article below. Antonini, C., Treyer, K., Streb. A., van der Spek, M., Bauer, C. and Mazzotti, M. (2020). Hydrogen production from natural gas and biomethane with carbon capture and storage – A techno-environmental analysis. Sustainable Energy Fuels; 4, 2967. DOI: 10.1039/d0se00222d Background data area based on Ecoinvent Cut-off database. Electrcity is based on AB grid. Natural gas has AB geography. Remaining techno-sphere inputs has either GLO or RoW geography.

Flow	Category	Amount	Provider	Description
aluminium oxide, metallurgical	... / 242:Manufacture of basic precious and other non-ferrous metals / 2420:Manufacture of basic precious and other non-ferrous metals	5.3270e-4 kg	market for aluminium oxide, metallurgical | aluminium oxide, metallurgical | Cutoff, U	There are multiple choices for aluminum oxide. The paper mig... Show more
chemical factory, organics	... / 429:Construction of other civil engineering projects / 4290:Construction of other civil engineering projects	5.3480e-10 Item(s)	chemical factory construction, organics | chemical	

Version
00.00.024

Last change
01/31/2022 6:33 am

UUID
96f74929-9cbb-4194-bc8e-99748e505f72

Quantitative reference
Hydrogen, gaseous, 33 bar

Location
Canada, Alberta

Process type
Unit process

Infrastructure process
✖

Valid from
-

Valid until
-

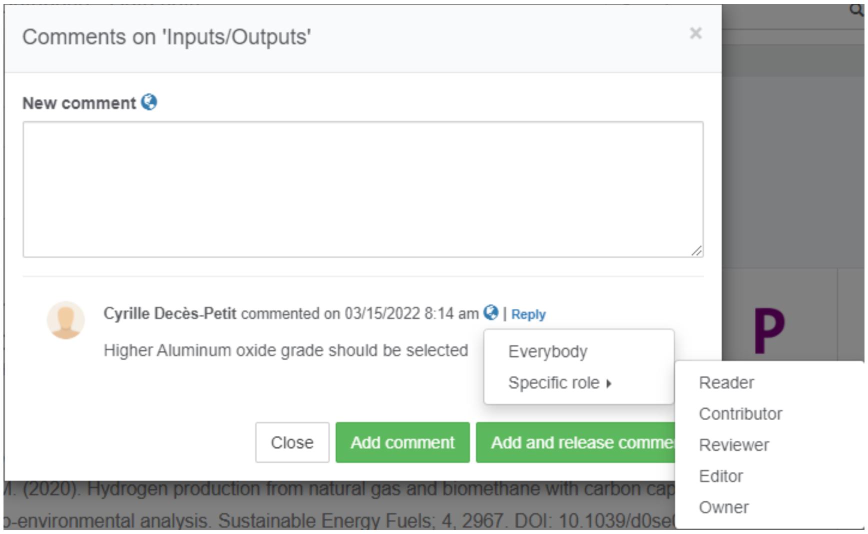
Time description

Geography description

Technology description

Figure 19: Review of a process dataset

When opening a comment (see Figure 20), users can add their own comment, and reply to existing comment. They can also specify if the comment can be seen by everyone who have access to the repository, or only to users with a specific role on the repository.



Comments on 'Inputs/Outputs'

Cyrille Decès-Petit commented on 03/15/2022 8:14 am

Higher Aluminum oxide grade should be selected

l. (2020). Hydrogen production from natural gas and biomethane with carbon cap
o-environmental analysis. Sustainable Energy Fuels; 4, 2967. DOI: 10.1039/d0se

Figure 20: Comments on a dataset field

Checkout commits

A checkout is used to replace the local openLCA database with a selected commit (version of the repository) from the openLCA CS. To checkout a commit, open the commit history via right-click *repository* then select *show in history.* Right click on a commit in the commit history in openLCA and select checkout (Figure 21). After the checkout, the database in openLCA is different from the repository. Before the next commit, the local database in openLCA would be replaced with the repository and thus the checkout reversed. However, users may want to check a commit out, disconnect from the repository and continue to work on the checked-out version of the database.

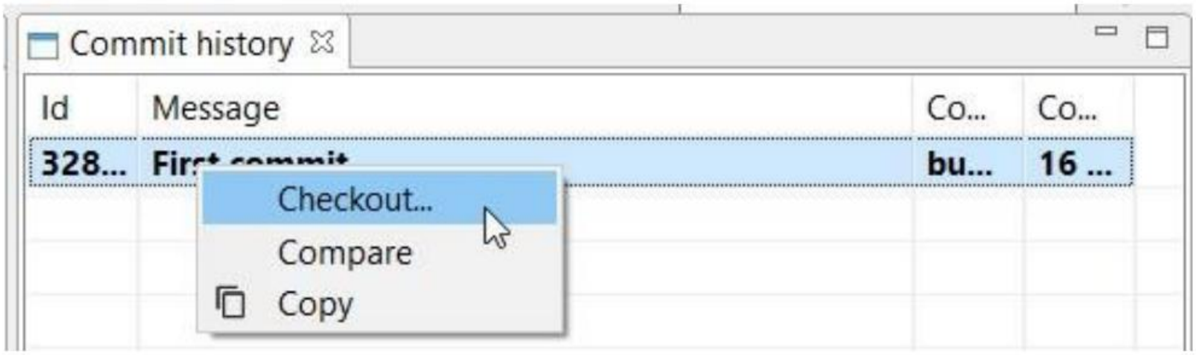


Figure 21: A checkout replaces the local openLCA database with a specific comment

Lifecycle datasets can be reviewed and validated within the collaboration team. Other collaborators can then retrieve and uses these data. They have then the ability to provide feedback on specific data of the dataset. These comments recorded on the collaboration server enhances the review and validation processes. Once validated, the data can be stored in publicly available repositories within the same collaboration server.

Restoring previous versions of a dataset

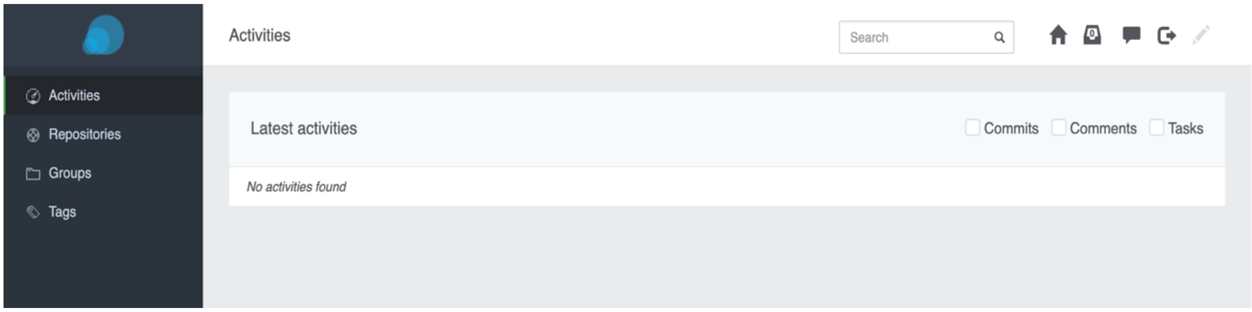
Although there is no direct option to restore previous versions of a repository, a workaround exists. This can be done by:

1. Right click on database in openLCA and select *Repository.*
2. Select *compare* with then *HEAD revision*.
3. Right-click and select overwrite with local changes and commit.

6. OpenLCA Collaboration Server Website

6.1 Dashboard /Home Page

In this section we look at the openLCA CS website and explain its components, how to use it and stay up to date with collaborations in a team. The figure below shows the first page after login. It may look different depending on the type of access a user has.

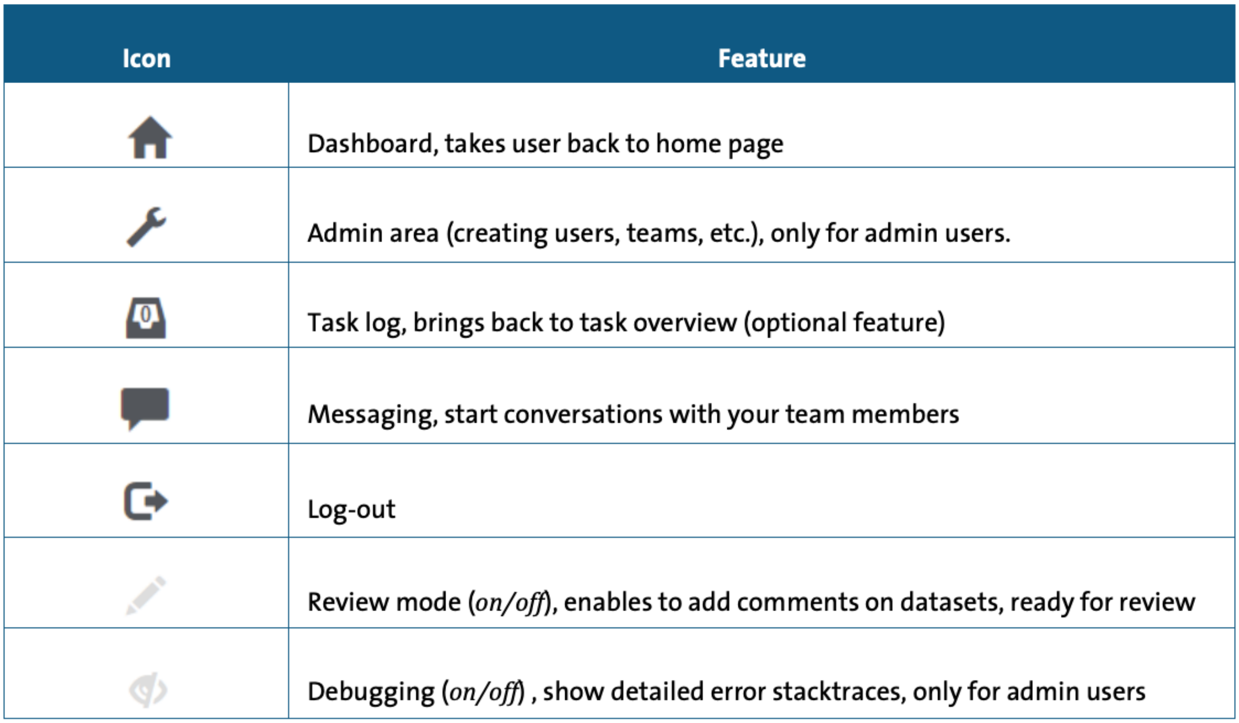


Latest activities ☐ Commits ☐ Comments ☐ Tasks

No activities found

Figure 22: Collaboration server dashboard

The icons at the top right corner of Figure 22 are explained in the table below.



Icon	Feature
🏠	Dashboard, takes user back to home page
🔧	Admin area (creating users, teams, etc.), only for admin users.
0	Task log, brings back to task overview (optional feature)
💬	Messaging, start conversations with your team members
↪	Log-out
✏	Review mode (*on/off*), enables to add comments on datasets, ready for review
👁	Debugging (*on/off*) , show detailed error stacktraces, only for admin users

Figure 23: Explanation of collaboration server dashboard tools

On the left of Figure 22 we then have tabs that are explained below:

Activities tab: This tab explains the latest activities that are associated with the repositories the user has access to. The user can choose to see the latest activities on commits, comments, or tasks by checking on their specific boxes.

Repositories tab: The repositories tab is the central place to access all repositories the user has been given access to. Repositories can also be managed from here, including creating a new repository or importing one. This although does depend on the type of role and access a user has.

Groups tab: The groups tab shows the groups a user is a part of. Each group has members that work on a number of repositories. To give access to multiple repositories that are in one group, a user can be added to that group.

Tags tab: This tab brings the user to the tag cloud.

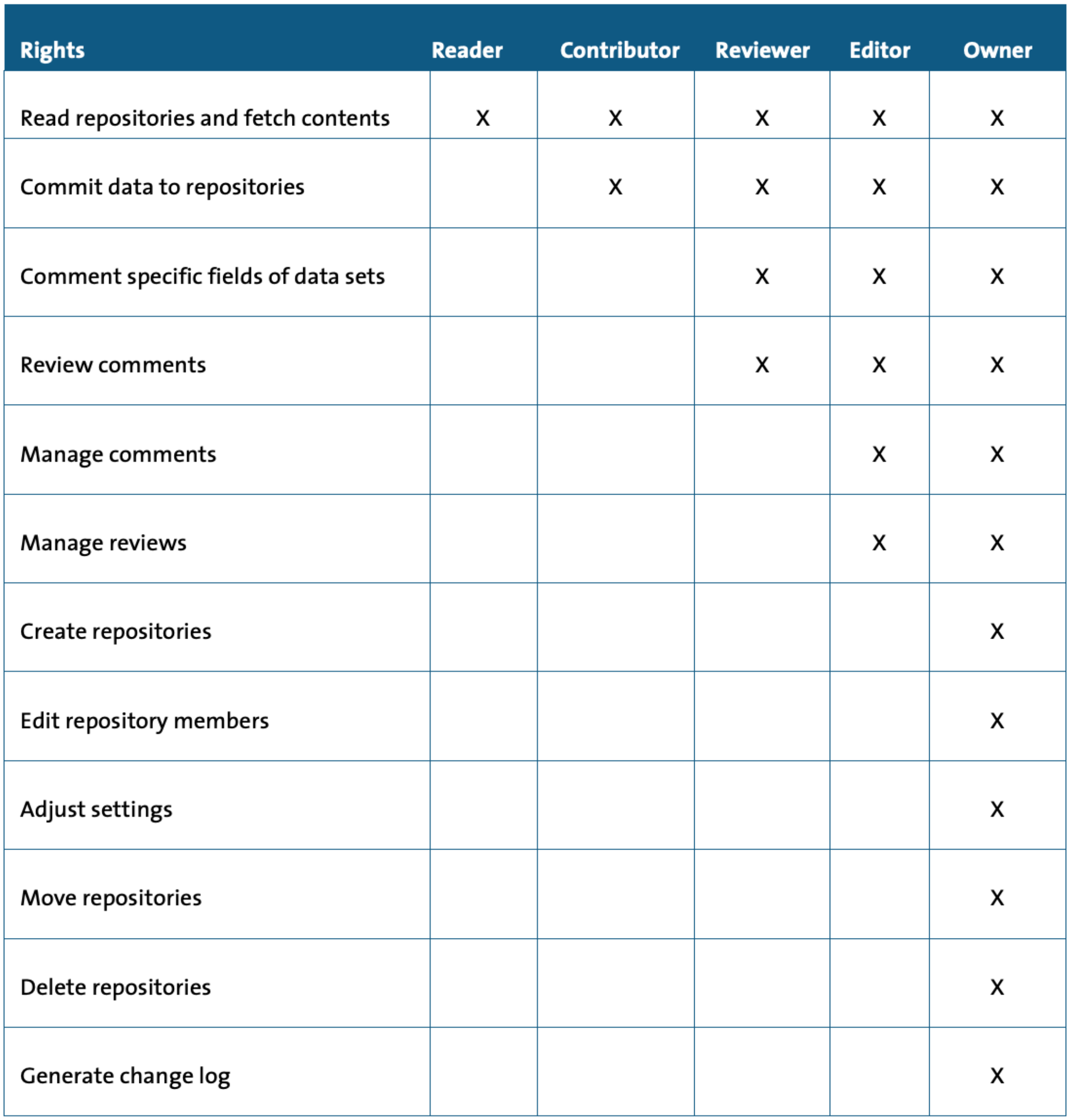
More information on the dashboard can be found in the openLCA Collaboration server user manual.

7. Users and access control

Users wanting to access an openLCA collaboration server should obtain an account with the level of privilege corresponding to their role with the data being accessed. The 2 mains roles are reader and contributor.

- The reader role allows users to read the data from the repository, download the full dataset, and keep their models up-to-date with the latest changed committed to the repositories.
- The contributor role allows users to read the data from the repository, and commit change to the dataset. Users with role can add and change to the dataset

The collaboration server provides several level of permissions to the users accessing the collaboration server (see Figure 24). A group developing a lifecycle dataset would typically be composed of a team of developers where some users would have the reader role allowing them to read the data from the repository, and others would have the contributor role allowing them to add new data and modify existing ones. The group would also have a quality control team composed of reviewers and editors. Reviewers would typically comments fields in a datasets as well as reviewing other's comments. Editors would manage comments and review and typically prepare change request to be implemented by the developer team. The repository owner role would administer the repositories and the users accessing the repositories.



Rights	Reader	Contributor	Reviewer	Editor	Owner
Read repositories and fetch contents	X	X	X	X	X
Commit data to repositories		X	X	X	X
Comment specific fields of data sets			X	X	X
Review comments			X	X	X
Manage comments				X	X
Manage reviews				X	X
Create repositories					X
Edit repository members					X
Adjust settings					X
Move repositories					X
Delete repositories					X
Generate change log					X

Figure 24: Rights and access of different users