LCA Collaboration Server

LCA Collaboration Server

User Manual and Introduction

Version: 1.0.4

Authors: Jonas Bunsen, Sebastian Greve, Andreas Ciroth

Date: 23rd July, 2018

GreenDelta GmbH
Müllerstrasse 135
D-13357 Berlin, Germany
www.greendelta.com

Tel. +49 30 48 496 – 030
Fax +49 30 48 496 – 991
gd@greendelta.com

GreenDelta
openLCA
@openLCA
ask.openLCA

Contents

List of Figures

List of Tables

Glossary

Checkout (p.18) - A *checkout* is the recovery of a previous commit from the LCA Collaboration Server repository. During a *checkout*, the local database will be replaced with a selection version of the database on the LCA Collaboration Server.

Commit (p.10) - A *commit* refers to a package of data sets that is contributed to the LCA Collaboration Server repository by a user from the user's local machine.

Fetch (p.12) - To *fetch* refers to updating the user's local database (openLCA) with changes from the remote repository of the LCA Collaboration Server.

Group (p.8) - A *group* contains one ore more repositories. *Users* and *teams* can be added to a *group* to gain access to the repositories that are assigned to a specific *group*.

Repository (p.9) - A *repository* it the equivalent to a database in openLCA and consists of grouped data sets (projects, product systems, processes, flows, indicators and parameters and background data). A *repository* mirrors the local databases of the working group's users.

Team (p.27) - A *team* is an aggregation of users. *Teams* can be added to *groups* to gain access to one or more specific repositories.

Abbreviations

GLAD - *Global LCA Data Access network*
LCA - *Life Cycle Assessment*
LCA CS - *LCA Collaboration Server*
LCT - *Life Cycle Thinking*
PEF - *Product Environmental Footprint*
USDA - *United States Department of Agriculture*

Has your team ever faced limitations in conducting an LCA study because only one user can work on an LCA model at a time?

The LCA Collaboration Server facilitates next-level collaborative Life Cycle Assessment modelling!

Introduction

openLCA >1.7 accommodates full integration of the LCA Collaboration Server and facilitates the development of LCA studies in distributed teams. The LCA Collaboration Server is not only a milestone for openLCA but offers innovative features and opportunities for LCA practitioners that are unparalleled by any other LCA software.[1] openLCA users are now able to work simultaneously on an LCA study while tracking each other's changes along synchronised databases.

1.1 What exactly is the LCA Collaboration Server?

The LCA Collaboration Server is a server application that complements openLCA (the LCA desktop application). It facilitates exchange and synchronisation of LCA data (e.g. flows, processes, product systems or entire LCA models) between users who work from different computers, enabling distributed, collaborative LCA modelling. The Collaboration Server introduces industry-established concepts from software development to the LCA world, with e.g. on-demand tracking of changes as well as comparison of databases and optional merging of data. It is so far unique. The Collaboration Server is available for free. Support is available on demand.

1.2 How can the LCA Collaboration Server be for free?

The LCA Collaboration Server was developed by GreenDelta with support from the United States Department of Agriculture (USDA), creator of the LCA Digital Commons[2]. Without support from the USDA, GreenDelta would not be able to offer the LCA Collaboration Server for free and GreenDelta regards this is a great chance to return a favour to the LCA community and support Life Cycle Thinking (LCT). Check-out conclusion and outlook on p.28 to see what we have in mind.

1.3 How to use the LCA Collaboration Server

With the general advancement of LCA and the spreading of LCT, collaborative work on LCA studies becomes increasingly common e.g. within a company or team at the same location, within a company or team at different locations within multilateral projects (e.g. in research) on an international level and under participation of various entities such as companies, universities or consultancies as well as co-development of an LCA among executing contractors and clients. Moreover, it is becoming increasingly commonplace to use reference data for background processes, elementary flows, impact assessment methods, and other elements of existing LCA models. Often sharing of reference data for updating databases and distribution to users who are (also) distributed comes along with technical issues. In addition, quality assurance and review of LCA models through

[1] Software solutions that offer a functionality close to that of the LCA Collaboration Server may be *SimaPro's* multi-user feature from *PRé Sustainability* and *Soda4LCA* from the *Karlsruhe Institute of Technology*

[2] https://www.lcacommons.gov/

an external reviewer is of interest for anybody who wants to create a consistent LCA database. The LCA Collaboration Server is exactly developed with such use cases in mind:

- Working in distributed teams on LCA modelling, without continuous access to the exact same database
- Distributing LCA reference data e.g. process data sets, elementary flows, units, contacts, entire LCA models, considering updates and changes in reference data
- Creating a consistent and reviewed database

The functionality of the LCA Collaboration Server is largely inspired by modern code sharing and repository software and introduces industry-proven state-of-the-art file and model management from IT- to LCA development:

- Simultaneous work on LCA studies with nearly real-time (on-demand) integration of changes from users working from different computers
- Tracking of changes to assure quality

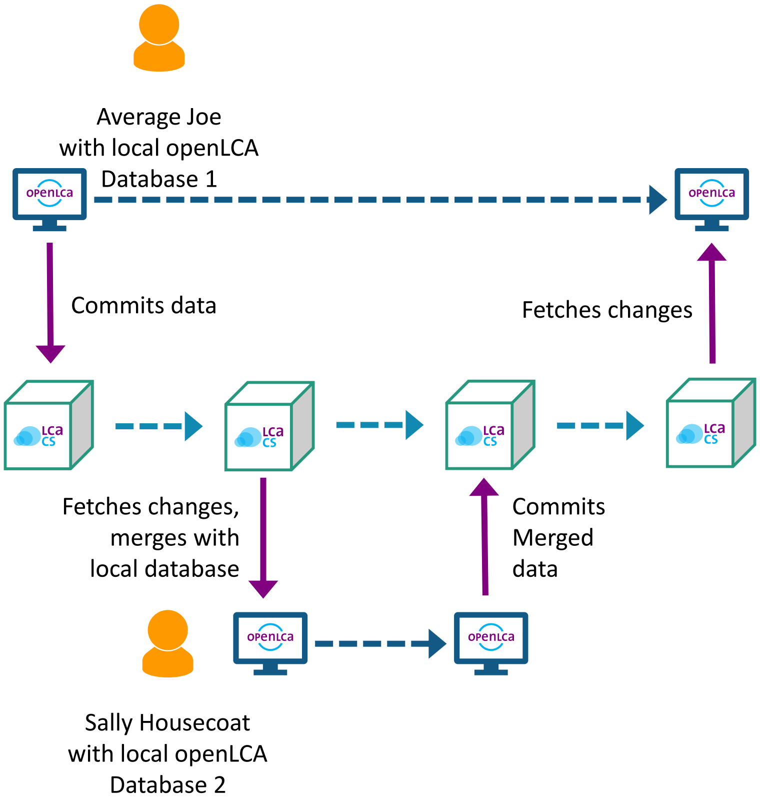


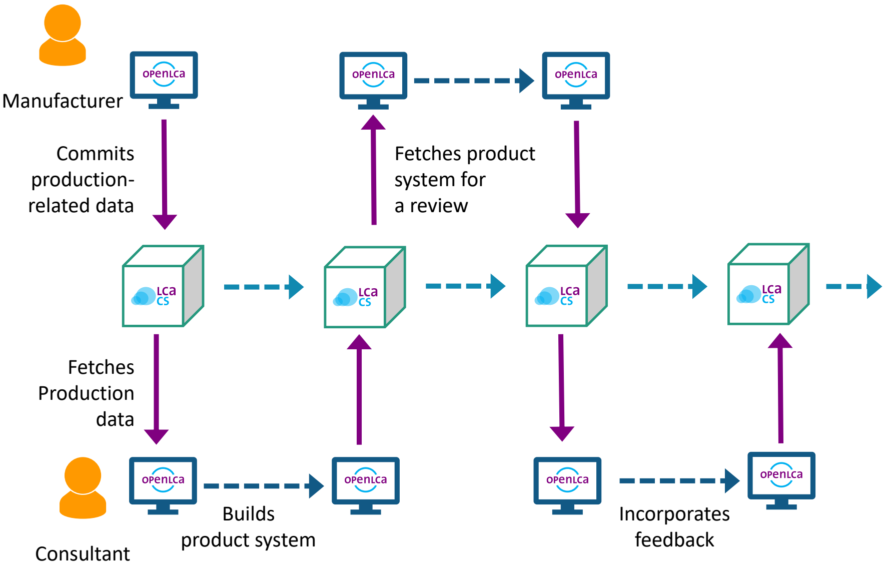
Figure 1.1: The LCA Collaboration Server facilitates LCA modelling in distributed teams.

1.3.1 Use case: Working on LCAs in a distributed working team

A manufacturer of recyclable bottles contracts a consulting office in another city to co-develop an LCA study of their flagship product. The manufacturer holds all the knowledge on how energy inputs, raw materials and other flows relate to specific production processes but lacks experience in LCA modelling. The consulting office has plenty of experience with LCA but depends on the manufacturer's information to assure the correctness and conclusiveness of the LCA study.

Use case: Working on LCAs in a distributed working groups

To ease daily exchange of LCA-related information and to limit costly and time-consuming personal meetings, both parties agree to set up an LCA Collaboration Server attached to the IT infrastructure of the consulting office. The manufacturer commits all information about specific flows and processes during the production of their product directly via the LCA Collaboration Server to a joint repository. After consultation with the manufacturer, the consulting office utilises the provided information to build an LCA model and saves it on the LCA Collaboration Server. Upon completion, the LCA model is fetched by the client from the LCA Collaboration Server for a review to validate whether the LCA model matches the actual production process. During the manufacturer's review, it turns out that specific flow properties proved to be inaccurate. The manufacturer issues comments directly via the LCA Collaboration Server dashboard to the consulting office which are then incorporated by the latter. With this approach, the LCA Collaboration Server facilitates iterative co-development of an LCA model in a joint repository through on-demand updates.

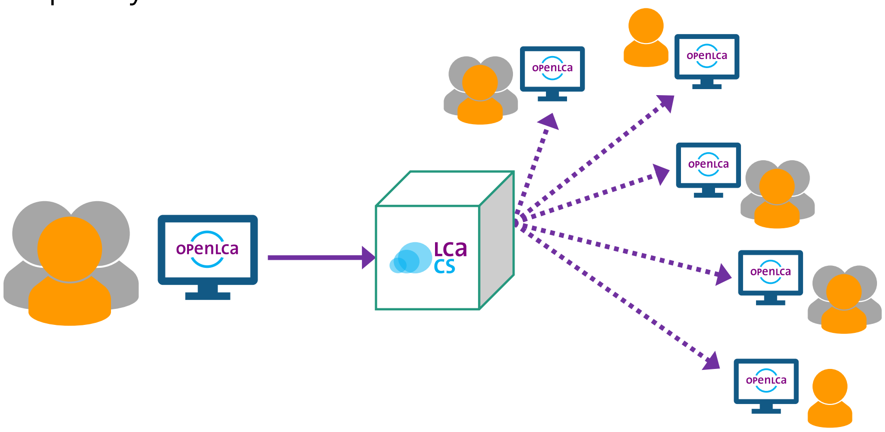


1.3.2 Use case: Straightforward sharing of data sets, LCA models and entire databases

Often it is challenging to exchange complete LCA models without losing a database's integrity. Moreover, it is especially difficult to track minor changes in a database. The LCA Collaboration Server facilitates straightforward sharing of LCA models, flows, processes and entire databases. Instead of having to first export LCA databases and to then transfer them via e-mail, file hosting applications or data storage devices, LCA models and databases can now easily be shared using the LCA Collaboration Server.

Use case: Straightforward sharing of LCA models and databases

A consortium member of a research project has collected LCA data sets to be utilised by other partners of the consortium. The consortium member sets up an LCA Collaboration Server and creates a repository from which the other consortium members can track changes. All members can fetch changes on demand and track the history of changes in the models created. Moreover, all members can comment on data in the repository.

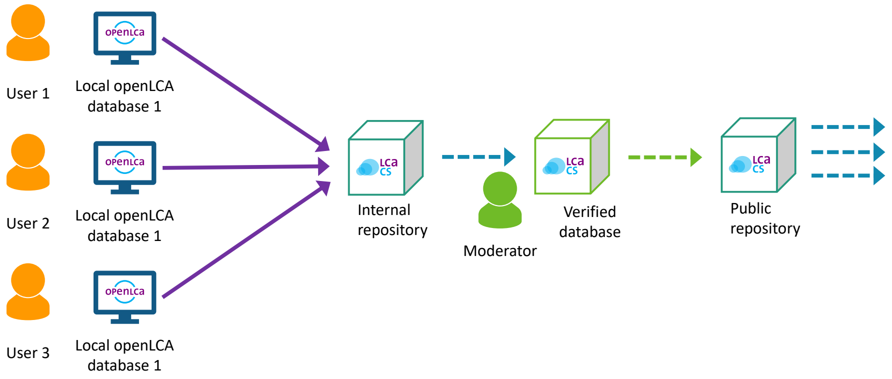


1.3.3 Use case: Building and managing verified public LCA repositories

The tracking feature of the LCA Collaboration Server introduces *versioning* and allows to review contributions to an LCA Collaboration Server repository. In case contributions do not meet preassigned requirements, *checking out previous versions* (p.18) facilitates the creation of verified databases.

Use case: Building and managing verified public LCA repositories

A research project aims at establishing a comprehensive LCA repository that contains data sets on polymers and plastics. To collect all the required information, the project team sets up an LCA Collaboration Server. Enterprises from the plastics industry contribute information on energy inputs, raw material inputs, ancillary inputs and other physical input flows into specific production processes as well as co-products, wastes, and emissions of those processes into the air, water and soil. However, most the participating enterprises have little experience with LCA and some of the data sets that are being contributed do not fulfil predefined requirements. Therefore, the enterprises first upload their data directly through openLCA and via the LCA Collaboration Server into a project-internal repository. A moderator reviews the project-internal repository and decides individually which data sets are accepted or declined. Accepted data sets are published to a publicly available repository via the LCA Collaboration Server.

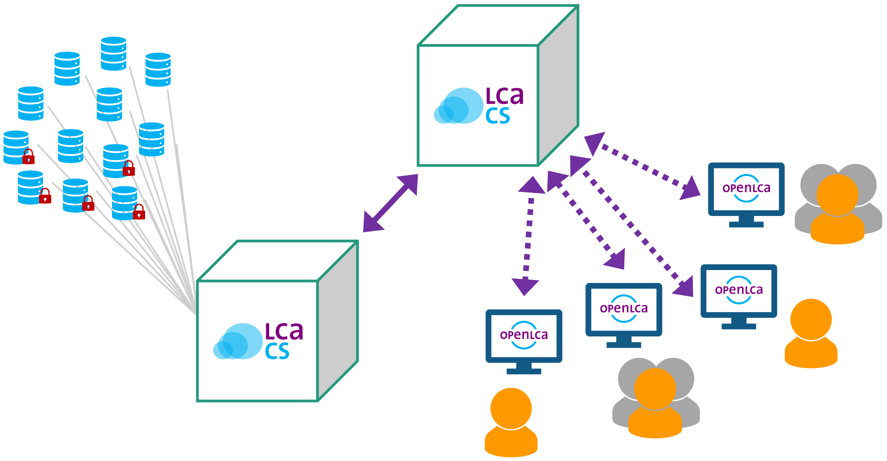


1.3.4 Use case: Managing and distributing reference data

Many projects aim at establishing reference data for LCA studies that conform with a specific industrial or environmental standard. Often, publishing reference data requires a compromise between accessibility and editability. The LCA Collaboration Server allows to share reference data within editable and publicly available repositories.

Use case: Managing and distributing reference data

The concept of Product Environmental Footprints (PEFs) has been initiated by the European Commission to measure the environmental performance of a product along its lifecycle. Currently, reference data for PEFs is under development and could be added to a public LCA Collaboration Server repository that is editable by users worldwide.



How to: Set Up

2.1 Get the LCA Collaboration Server

1. Download the LCA Collaboration Server from https://www.openlca.org/
2. Install a local instance of the LCA Collaboration Server on your server
   - Hardware requirements for the LCA Collaboration Server (p.42)
   - Installation - short description (p.32)
   - Installation - long description (p.33)
3. Open the LCA Collaboration Server dashboard in a web browser (p.8)

2.1.1 LCA Collaboration Server dashboard

The icons in the top right corner give access to the features outline in Table 2.1.1 (from left to right; Figure 2.1 p.8).

Table 2.1: Features accessible via the LCA Collaboration Server dashboard.

Icon	Feature
Wrench	Admin area (creating users, teams, etc.)
Paper tray	Tasks (create and manage tasks)
Speech bubble	Messaging (start conversations with your team members)
Exit symbol	Log out
Pencil	Review mode (*on/off*)

Clicking on the user name in the bottom left corner of the LCA Collaboration Server dashboard gives access to account properties, messaging and notification settings. The dashboard may look slightly different for members if members have different user rights.

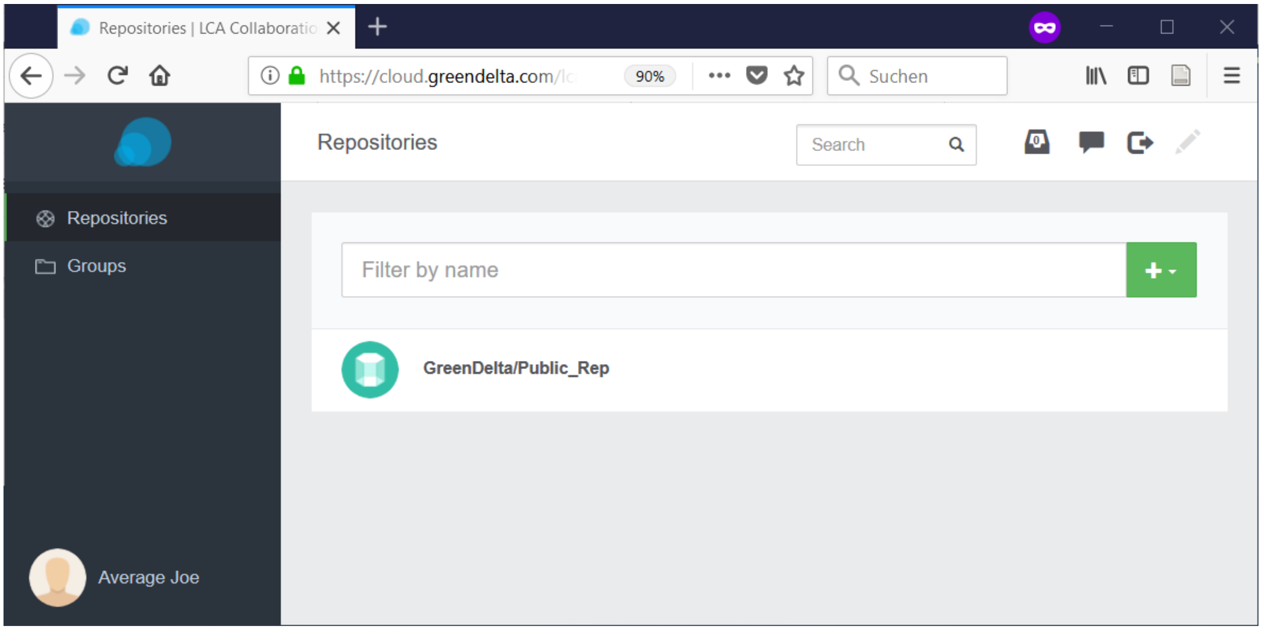


Figure 2.1: The LCA Collaboration Server dashboard.

2.1.2 Groups and user roles

The group feature allows to create *working groups* to which a repository or several repositories (see below) can be assigned. A working group can be managed by assigning different user roles to its members (Table 2.1.2)[1].

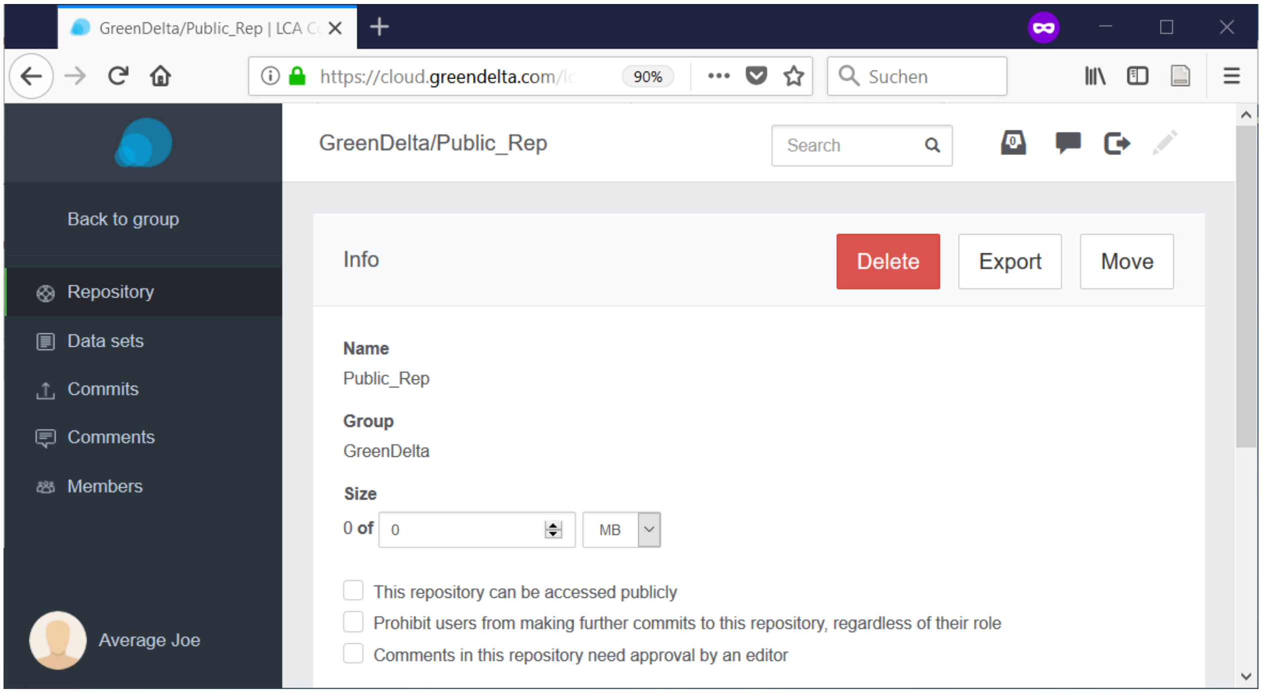
Table 2.2: User rights

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

[1] Additional roles are those of the data manager (administrate libraries) and user manager (create users and teams); p. 26.

2.1.3 Repositories - Create a new repository for your working group

A repository is the equivalent to a database in openLCA and consists of grouped data sets (projects, product systems, processes, flows, indicators and parameters and background data). Basically, a repository mirrors the local openLCA databases of the working group's users. When creating a repository, it is possible to limit the repository's maximum size on the server (*0 = unlimited*). The left sidebar of the repository dashboard contains menu entries for the repository, data sets, commits[2], comments and members of the repository (Figure 2.2).



Name
Public_Rep

Group
GreenDelta

Size
0 of 0 MB

□ This repository can be accessed publicly
□ Prohibit users from making further commits to this repository, regardless of their role
□ Comments in this repository need approval by an editor

Figure 2.2: The LCA Collaboration Server repository dashboard.

2.2 Set up: openLCA (client)

Download the most recent version of openLCA from http://www.openlca.org and install it on your computer (refer to p.42 for hardware requirements).

2.2.1 Enable the LCA Collaboration Server feature in openLCA.

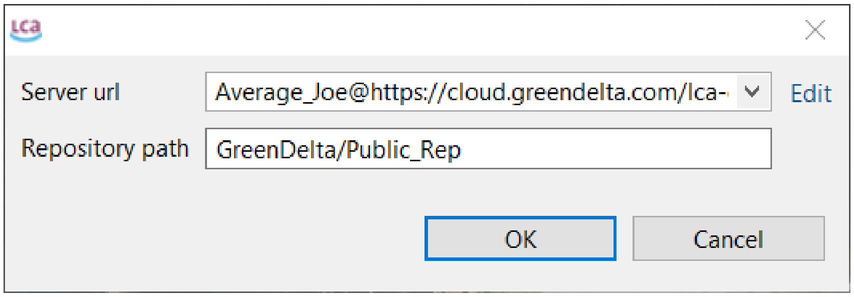
Click on *file* > *preferences* > *collaboration* and check mark *enable collaboration*. Add the server URL, user and password and click *apply*.

[2] In simple terms, *a commit* is a package of data sets that a user contributes to the repository from his/her local database and can be understood as a change to the working group's shared database (please refer to p.10 for more information).

How-To: Basic Workflow

3.1 Connecting to a repository

Start openLCA and open a database. Right-click on the database with which you would like to work with your distributed team and select *repository* > *connect to repository*. The user will be prompted to select a Server URL as well as the repository path which follows the scheme of *GroupName* *RepositoryName* (Figure 3.1).

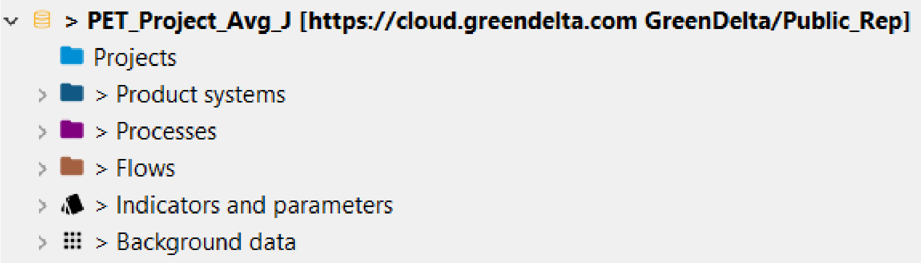


Server url: Average_Joe@https://cloud.greendelta.com/lca-

Repository path: GreenDelta/Public_Rep

Figure 3.1: Connecting a database in openLCA to a repository of the LCA Collaboration Server.

Upon clicking on *OK*, openLCA connects to the repository and rebuilds the index of the selected database. Upon successful connection, the repository URL is displayed in squared brackets behind the name of the local database (Figure 3.2).



> PET_Project_Avg_J [https://cloud.greendelta.com GreenDelta/Public_Rep]
Projects
> Product systems
> Processes
> Flows
> Indicators and parameters
> Background data

Figure 3.2: A successful connection of a database with a repository is indicated through the repository URL in squared brackets behind the name of the local database.

To disconnect from a repository right-click on the respective database and select *repository* > *disconnect from repository*.

3.2 Committing data sets to the connected repository

Upon successfully connecting a database to a repository, a right-click on any item in the list underneath the database's name displays the new menu entry 'Repository' (Figure 3.3).

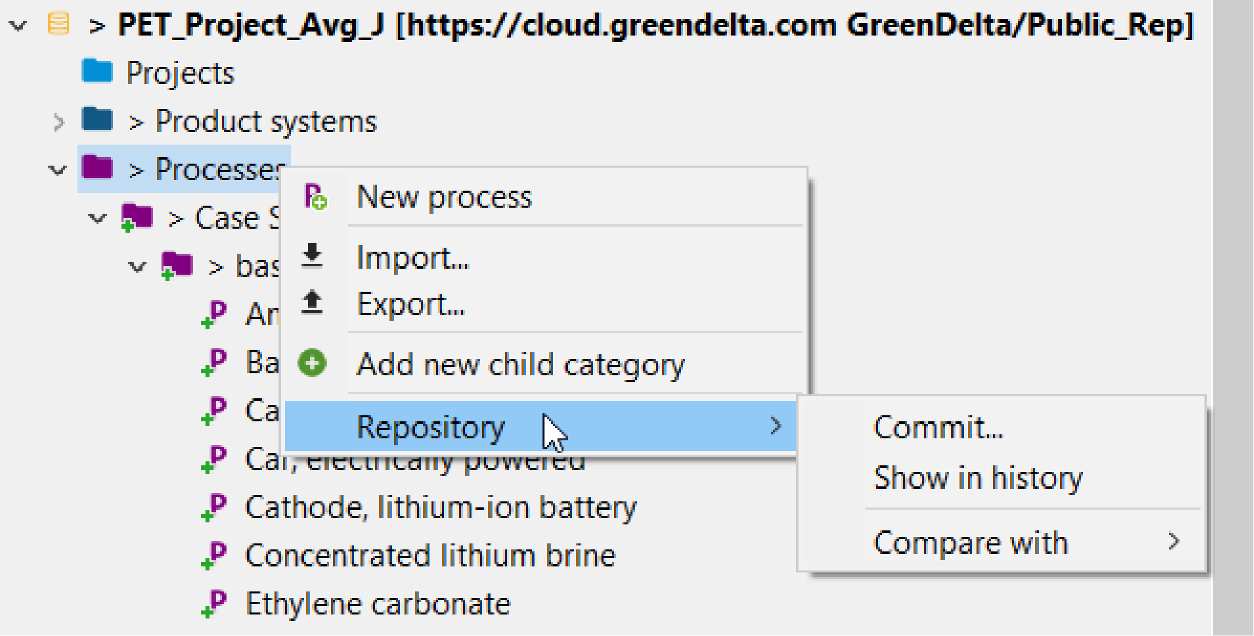
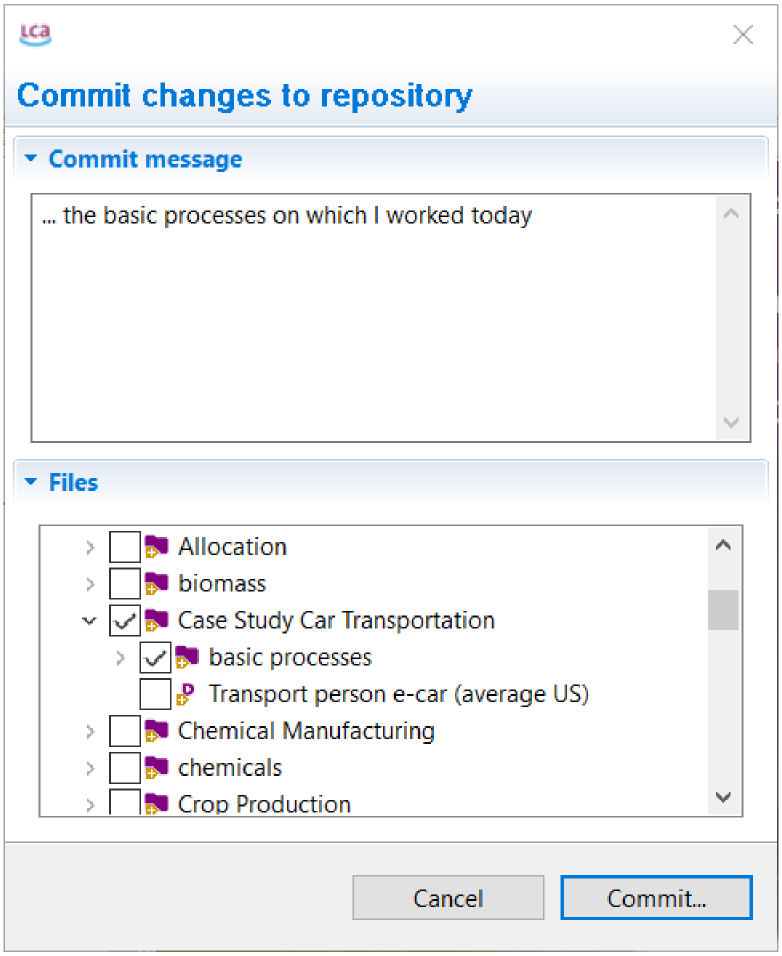


Figure 3.3: Right-click on any item in the database to initiate a commit.

Click on commit to initiate committing changes to the repository. The user will be prompted to select which data sets are to be committed. A commit message is mandatory and must be added to each commit (Figure 3.4). The speed of the commit depends on your network connection and the size of your database.



Commit changes to repository

Commit message

... the basic processes on which I worked today

Files

Figure 3.4: Committing changes to a repository requires a ’commit message’.

Hint I: To maintain transparency and clarity, it is recommended to regularly commit small data sets rather than to occasionally commit large data sets. This helps team members to trace and understand changes, guarantees clean versioning of a repository and helps to avoid pitfalls (p.16).
Hint II: If commits are large (e.g. entire databases) openLCA may ‘*not respond*’ anymore. This is only temporary and may last for several minutes. The commit is still ongoing (see network activity). It is also possible to select only specific data sets for a commit via right-click + CTRL.

3.3 Fetching data sets from the repository

To fetch data sets from a repository, right-click on a local database (make sure you’re connected to the repository) and select *repository* > *fetch* (Figure 3.5).

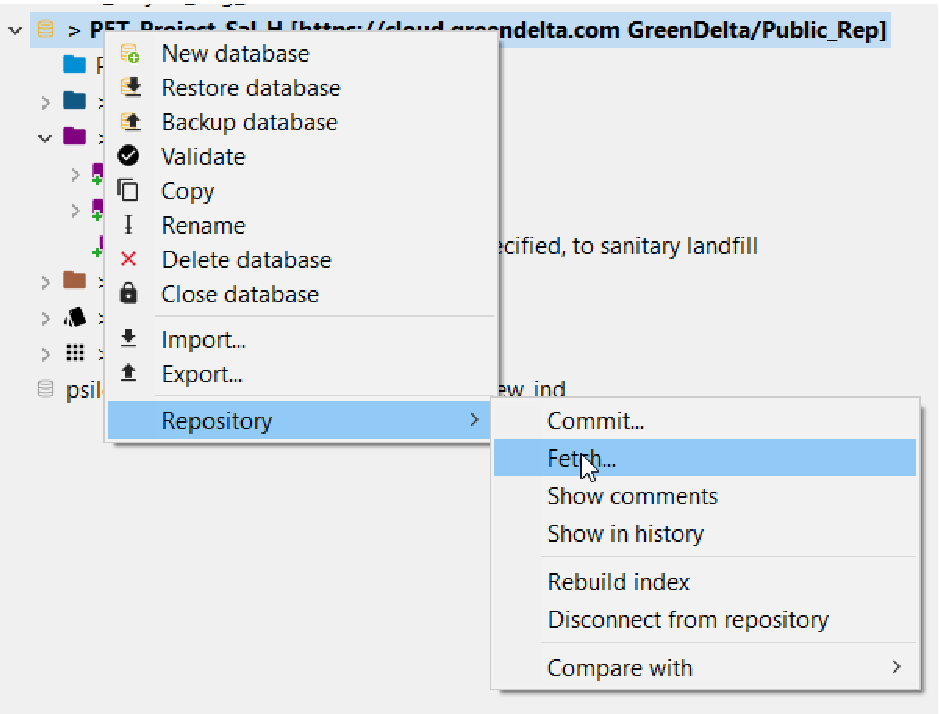
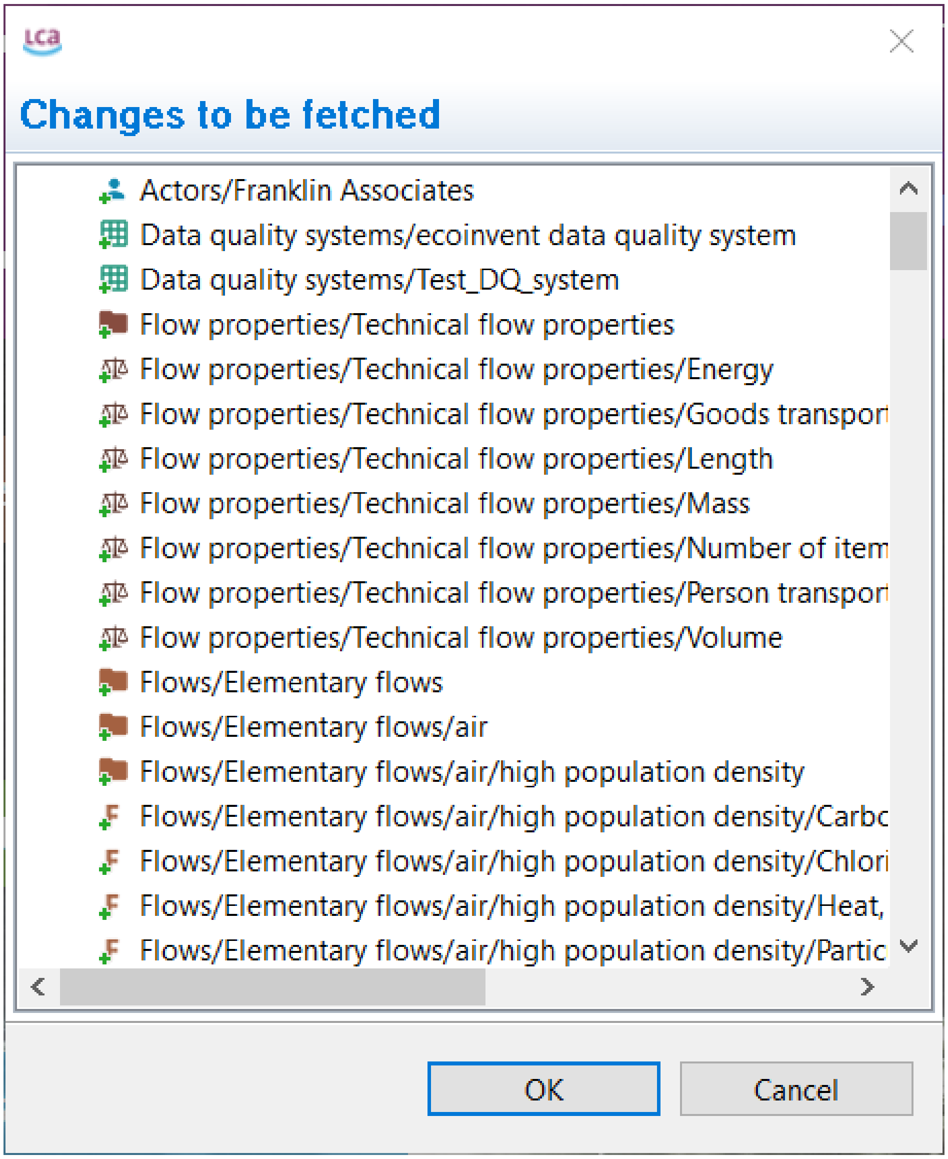


Figure 3.5: Fetching a repository from the LCA Collaboration Server in openLCA.

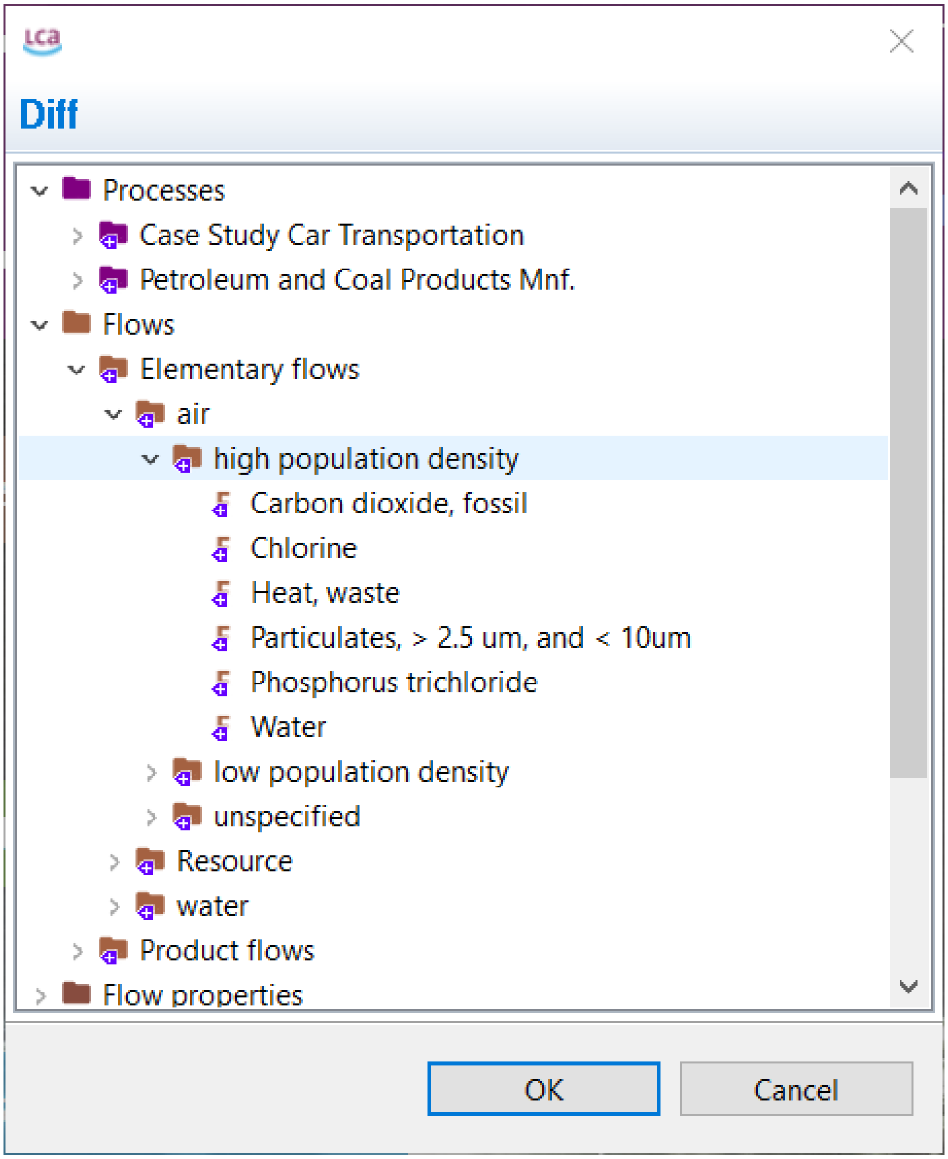
Two windows will pop up: The first one lists all changes of each individual commit since the last fetch (Figure 3.6). The second one lists all commits since the last fetch in a merged and structured list (Figure 3.7).



Changes to be fetched

- Actors/Franklin Associates
- Data quality systems/ecoinvent data quality system
- Data quality systems/Test_DQ_system
- Flow properties/Technical flow properties
- Flow properties/Technical flow properties/Energy
- Flow properties/Technical flow properties/Goods transpor
- Flow properties/Technical flow properties/Length
- Flow properties/Technical flow properties/Mass
- Flow properties/Technical flow properties/Number of item
- Flow properties/Technical flow properties/Person transpor
- Flow properties/Technical flow properties/Volume
- Flows/Elementary flows
- Flows/Elementary flows/air
- Flows/Elementary flows/air/high population density
- Flows/Elementary flows/air/high population density/Carbc
- Flows/Elementary flows/air/high population density/Chlori
- Flows/Elementary flows/air/high population density/Heat,
- Flows/Elementary flows/air/high population density/Partic

Figure 3.6: Simple list of changes that need to be fetched.



Diff

- Processes
  - Case Study Car Transportation
  - Petroleum and Coal Products Mnf.
- Flows
  - Elementary flows
    - air
      - high population density
        - Carbon dioxide, fossil
        - Chlorine
        - Heat, waste
        - Particulates, > 2.5 um, and < 10um
        - Phosphorus trichloride
        - Water
      - low population density
      - unspecified
    - Resource
    - water
  - Product flows
- Flow properties

Figure 3.7: Structured list of all changes that need to be fetched.

Hint: To maintain transparency and clarity, it is recommended to fetch regularly. This enables the user to trace and understand changes easily and helps to avoid conflicts (p.16).

3.4 The workflow is always linear

To understand how exactly the LCA Collaboration Server works, it is important to note that the workflow of *commits* and *fetches* is always linear. In practice, this means for instance that users cannot commit data sets to the server indiscriminately. In fact, before each commit, the user has to fetch the current state of the repository from the LCA Collaboration Server. (Figure 3.8).

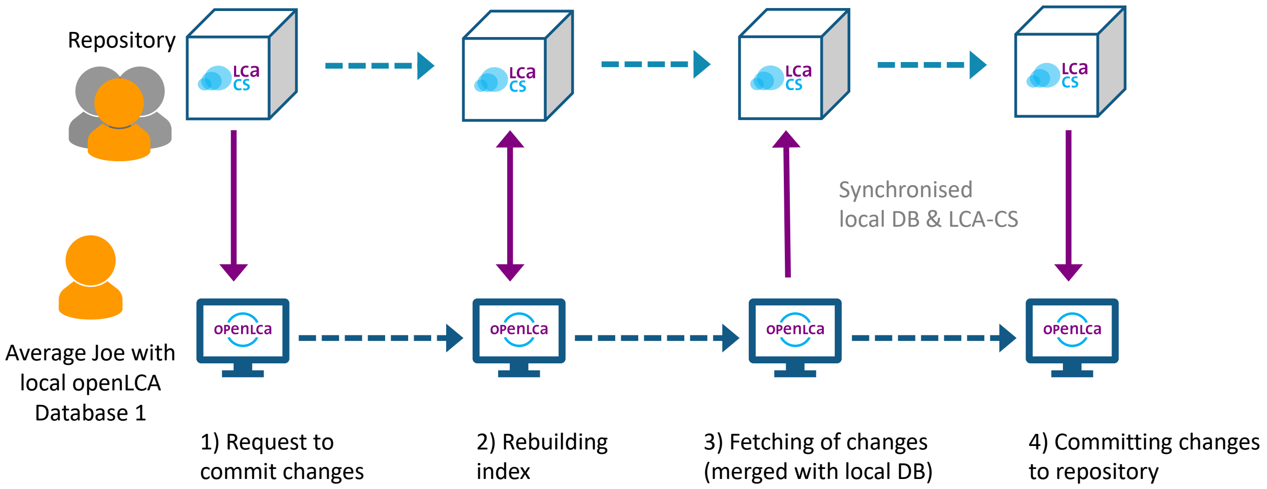

Repository
Average Joe with local openLCA Database 1
Synchronised local DB & LCA-CS
LCA CS
openLCA
1) Request to commit changes
2) Rebuilding index
3) Fetching of changes (merged with local DB)
4) Committing changes to repository

Figure 3.8: The workflow of a commit is always linear.

3.5 Conflicts

Conflicts may occur if two or more users work on the same database item or interlinked database items at the exact same time. This is best avoided by directly coordinating work within your team as well as to perform regular fetches and to commit regularly. The diff utility (p.25) assists in detecting commits by other users and to identify potential conflicts (p.16).

3.6 Pitfalls

Eventually, users are responsible for the integrity of a repository and must understand that a repository is a complex interlinked system. Interlinkages must be respected and a commit must always contain all interlinkages to any of the changes in the data set.

3.6.1 Validating databases

To assure that all interlinkages within a database are set correctly, databases should be validated regularly (right-click on a database and select *validate*). A validation identifies whether all linkages within a database are working.

If a validation fails, missing interlinkages should be fixed before performing a commit. If a user is unable to validate a database it is recommended to discard changes, fetch the repository again and to redo modelling steps. A user may fail to validate a database if corrupt data sets were fetched from the repository that have been committed by other users.

3.6.2 Example 1 - Missing interlinked flow

A user adds a flow to a local database and uses the flow as an input for a process. The user commits the new 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.

3.6.3 Example 2 - Modified flow property

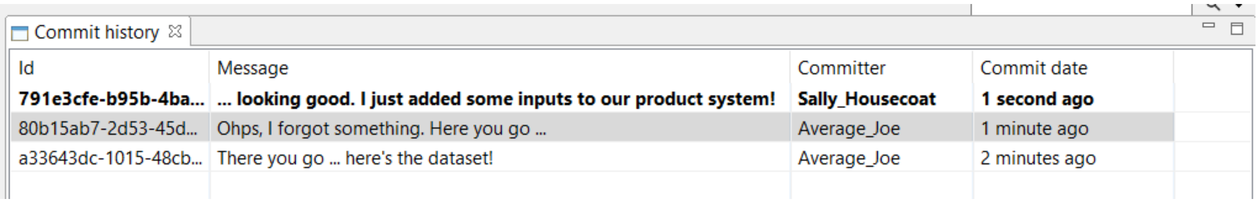
A user changes the flow property of an existing flow in a local openLCA database and uses the flow with the updated property for a process. The user commits the process to the repository but not the flow with the modified flow property. The modification of the flow is missing in the repository and corrupts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

Features

4.1 Basic features

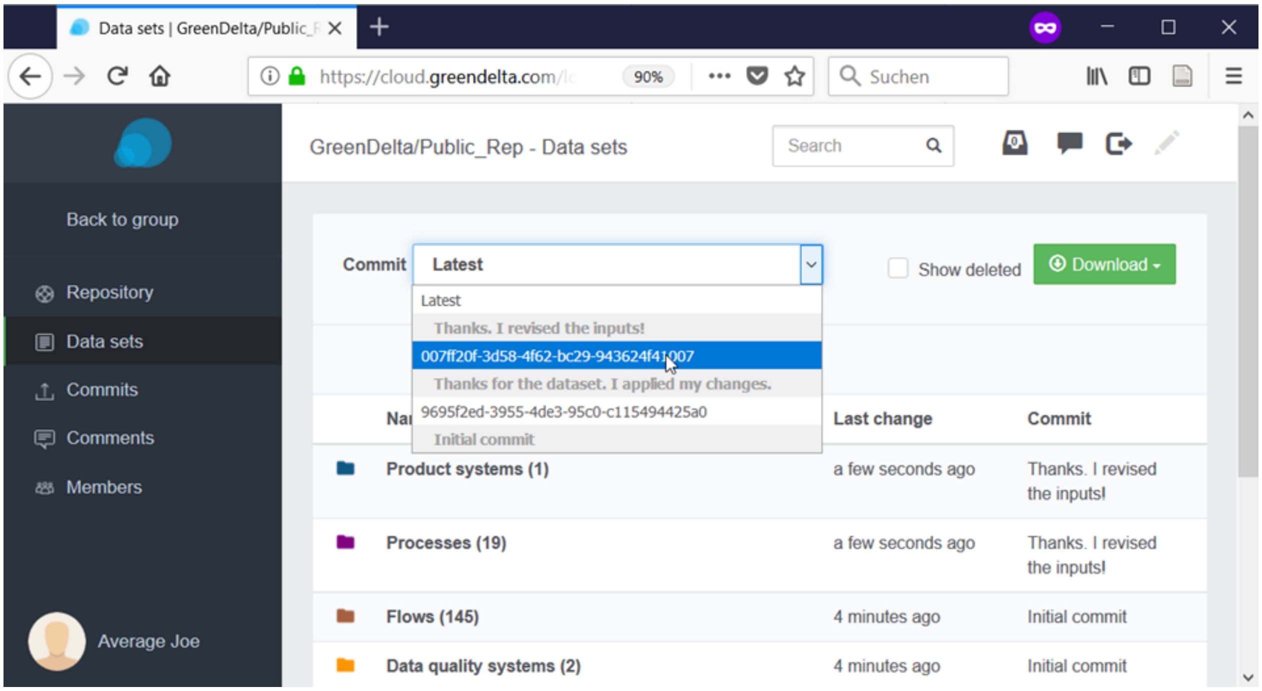
4.1.1 Commit history

A history of all commits is available in openLCA via right-click on a database Repository > Show in history (Figure 4.1) and via the LCA Collaboration Server dashboard (Figure 4.2).



Id	Message	Committer	Commit date
791e3cfe-b95b-4ba...	**... looking good. I just added some inputs to our product system!**	**Sally_Housecoat**	**1 second ago**
80b15ab7-2d53-45d...	Ohps, I forgot something. Here you go ...	Average_Joe	1 minute ago
a33643dc-1015-48cb...	There you go ... here's the dataset!	Average_Joe	2 minutes ago

Figure 4.1: History of commits in openLCA.



Commit: Latest

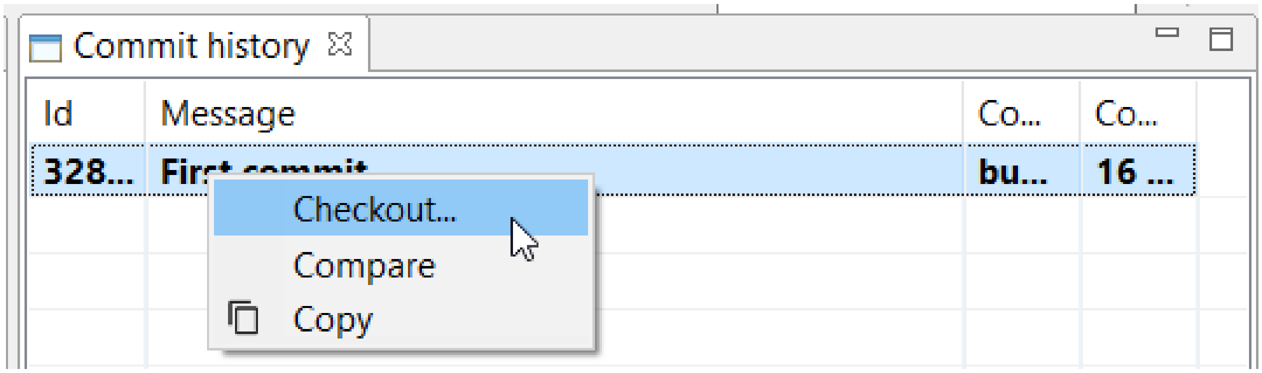
- Latest
- Thanks. I revised the inputs!
- 007ff20f-3d58-4f62-bc29-943624f41007
- Thanks for the dataset. I applied my changes.
- 9695f2ed-3955-4de3-95c0-c115494425a0
- Initial commit

Name	Last change	Commit
Product systems (1)	a few seconds ago	Thanks. I revised the inputs!
Processes (19)	a few seconds ago	Thanks. I revised the inputs!
Flows (145)	4 minutes ago	Initial commit
Data quality systems (2)	4 minutes ago	Initial commit

Figure 4.2: History of commits in the LCA Collaboration Server dashboard.

4.1.2 Checkout commits

A checkout replaces the local openLCA database with a selected commit (version of the repository) from the LCA Collaboration Server. To checkout a commit, open the commit history via right-click repository > show in history. Right-click on a commit in the commit history in openLCA and select checkout (Figure 4.3). 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.

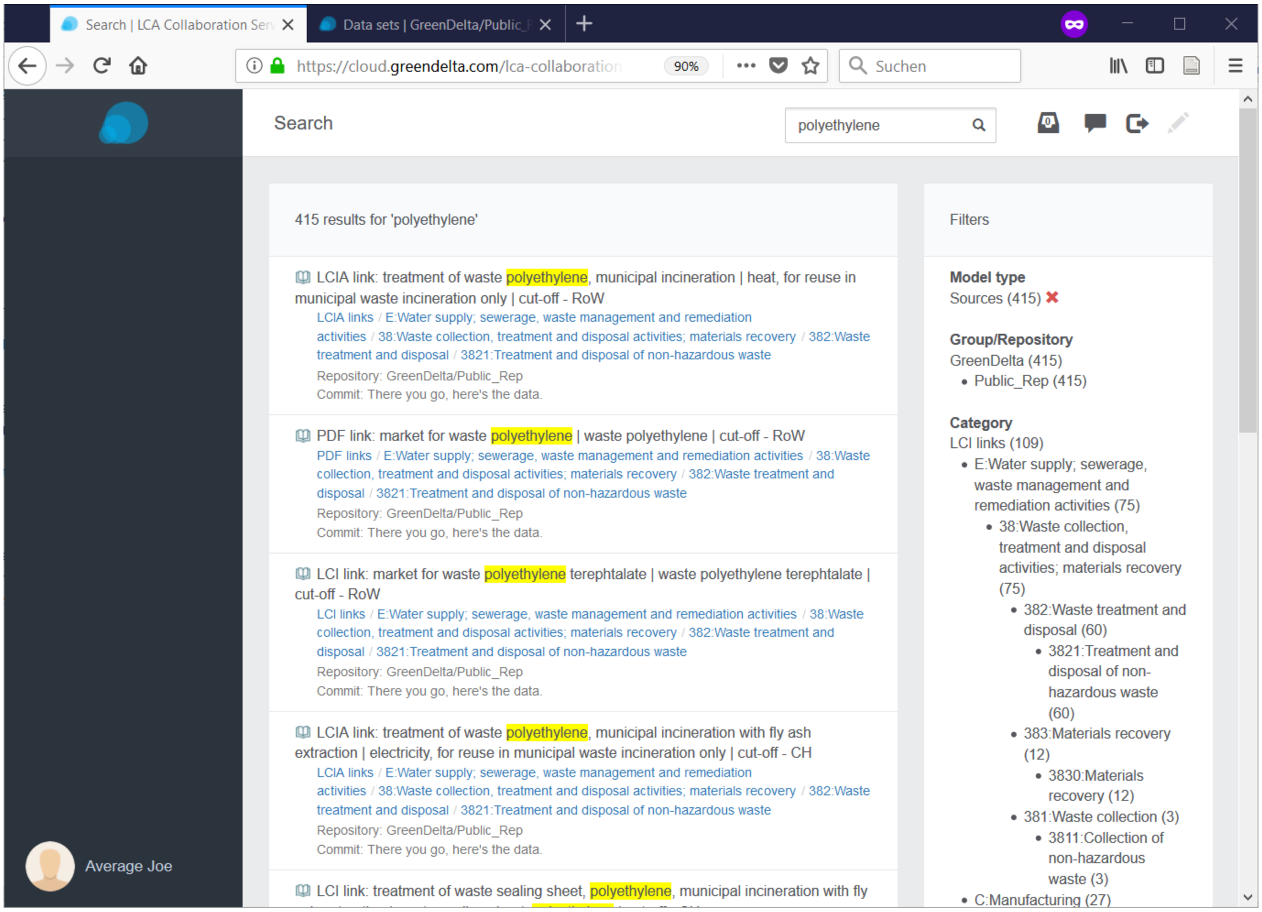


Id	Message	Co...	Co...
328...	First commit	bu...	16 ...

Figure 4.3: A checkout replaces the local openLCA database with a specific commit.

4.1.3 Search function and filter (LCA Collaboration Server dashboard)

A repository-overarching search function is available via the LCA Collaboration Server dashboard. Simply use the search form in the top bar. Filters allow enhanced navigation through the search results (Figure 4.4 and Figure 4.5). All search items are referenced and link to the respective items. The search function also allows unregistered users to search through a repository if the latter is a public repository.


Search | LCA Collaboration Serv
Data sets | GreenDelta/Public_
https://cloud.greendelta.com/lca-collaboration
90%
Suchen
Search
polyethylene
Average Joe

415 results for 'polyethylene'

LCIA link: treatment of waste polyethylene, municipal incineration | heat, for reuse in municipal waste incineration only | cut-off - RoW
LCIA links / E:Water supply; sewerage, waste management and remediation activities / 38:Waste collection, treatment and disposal activities; materials recovery / 382:Waste treatment and disposal / 3821:Treatment and disposal of non-hazardous waste
Repository: GreenDelta/Public_Rep
Commit: There you go, here's the data.

PDF link: market for waste polyethylene | waste polyethylene | cut-off - RoW
PDF links / E:Water supply; sewerage, waste management and remediation activities / 38:Waste collection, treatment and disposal activities; materials recovery / 382:Waste treatment and disposal / 3821:Treatment and disposal of non-hazardous waste
Repository: GreenDelta/Public_Rep
Commit: There you go, here's the data.

LCI link: market for waste polyethylene terephtalate | waste polyethylene terephtalate | cut-off - RoW
LCI links / E:Water supply; sewerage, waste management and remediation activities / 38:Waste collection, treatment and disposal activities; materials recovery / 382:Waste treatment and disposal / 3821:Treatment and disposal of non-hazardous waste
Repository: GreenDelta/Public_Rep
Commit: There you go, here's the data.

LCIA link: treatment of waste polyethylene, municipal incineration with fly ash extraction | electricity, for reuse in municipal waste incineration only | cut-off - CH
LCIA links / E:Water supply; sewerage, waste management and remediation activities / 38:Waste collection, treatment and disposal activities; materials recovery / 382:Waste treatment and disposal / 3821:Treatment and disposal of non-hazardous waste
Repository: GreenDelta/Public_Rep
Commit: There you go, here's the data.

LCI link: treatment of waste sealing sheet, polyethylene, municipal incineration with fly

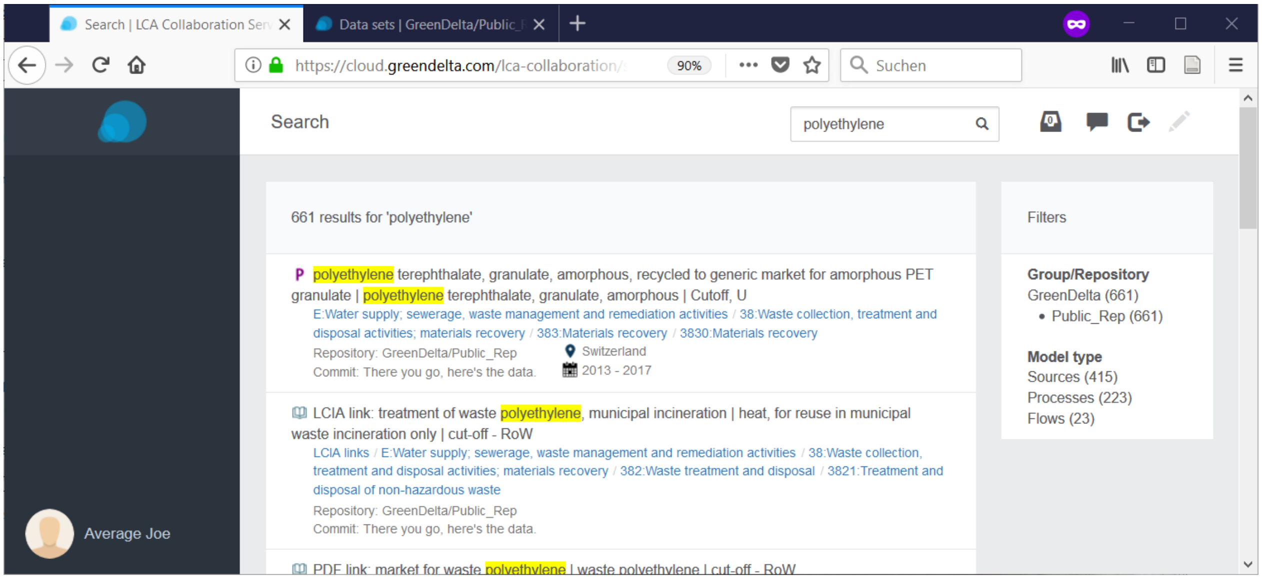
Filters

Model type
Sources (415) ✖

Group/Repository
GreenDelta (415)
- Public_Rep (415)

Category
LCI links (109)
- E:Water supply; sewerage, waste management and remediation activities (75)
  - 38:Waste collection, treatment and disposal activities; materials recovery (75)
    - 382:Waste treatment and disposal (60)
      - 3821:Treatment and disposal of non-hazardous waste (60)
    - 383:Materials recovery (12)
      - 3830:Materials recovery (12)
    - 381:Waste collection (3)
      - 3811:Collection of non-hazardous waste (3)
- C:Manufacturing (27)

Figure 4.4: Grouped search results.


Search | LCA Collaboration Serv
Data sets | GreenDelta/Public_
https://cloud.greendelta.com/lca-collaboration/
90%
Suchen
Search
polyethylene
Average Joe

661 results for 'polyethylene'

P polyethylene terephthalate, granulate, amorphous, recycled to generic market for amorphous PET granulate | polyethylene terephthalate, granulate, amorphous | Cutoff, U
E:Water supply; sewerage, waste management and remediation activities / 38:Waste collection, treatment and disposal activities; materials recovery / 383:Materials recovery / 3830:Materials recovery
Repository: GreenDelta/Public_Rep
Commit: There you go, here's the data.
Switzerland
2013 - 2017

LCIA link: treatment of waste polyethylene, municipal incineration | heat, for reuse in municipal waste incineration only | cut-off - RoW
LCIA links / E:Water supply; sewerage, waste management and remediation activities / 38:Waste collection, treatment and disposal activities; materials recovery / 382:Waste treatment and disposal / 3821:Treatment and disposal of non-hazardous waste
Repository: GreenDelta/Public_Rep
Commit: There you go, here's the data.

PDF link: market for waste polyethylene | waste polyethylene | cut-off - RoW

Filters

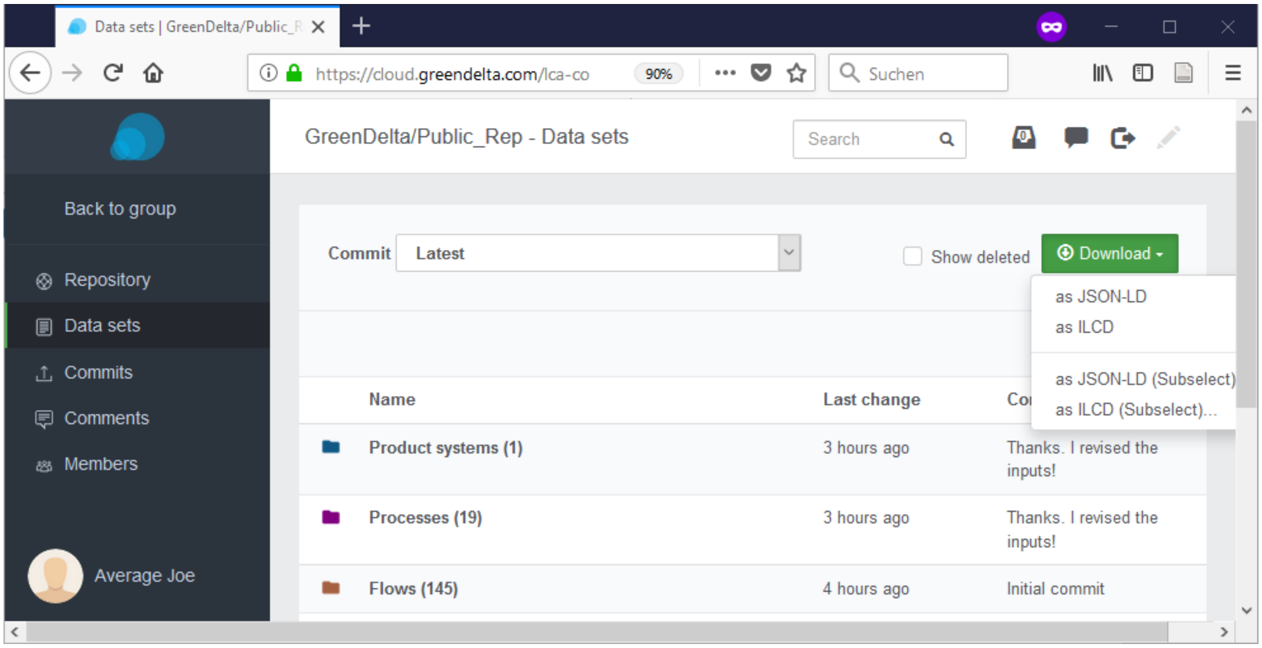
Group/Repository
GreenDelta (661)
- Public_Rep (661)

Model type
Sources (415)
Processes (223)
Flows (23)

Figure 4.5: Ungrouped search results.

4.1.4 Download and export of repositories

The download feature allows to download individual versions (commits) of a repository as *JSON-LD* or *ILCD* (Figure 4.6) file.



Name	Last change	Co
Product systems (1)	3 hours ago	Thanks. I revised the inputs!
Processes (19)	3 hours ago	Thanks. I revised the inputs!
Flows (145)	4 hours ago	Initial commit

Figure 4.6: Download and export of data sets.

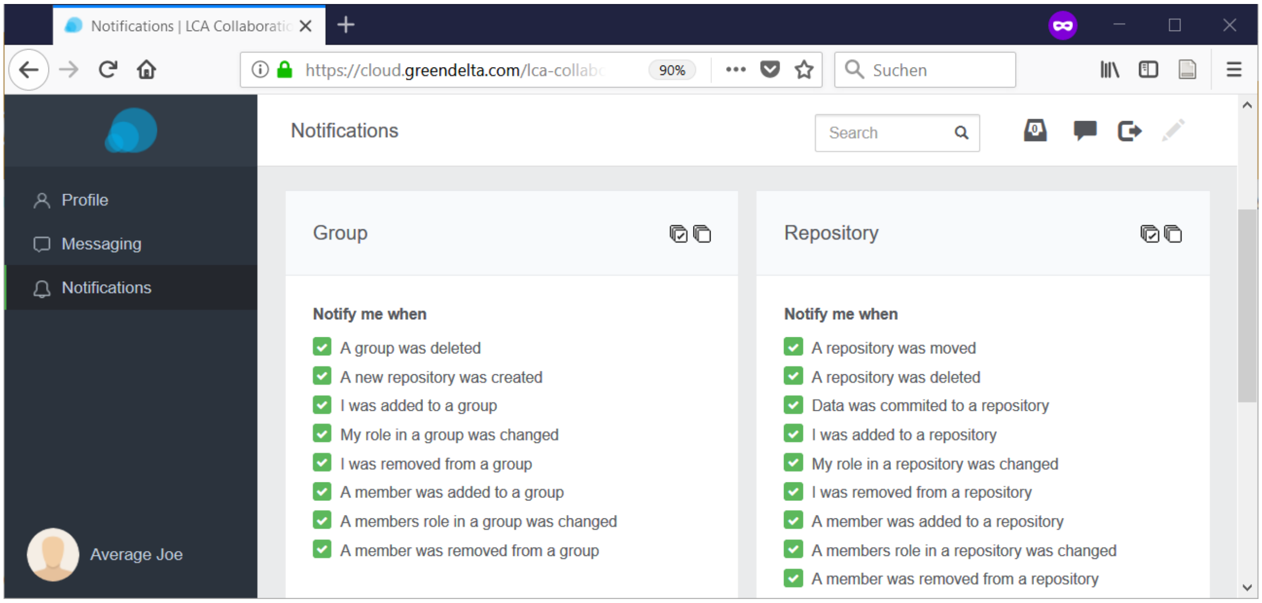
4.2 Advanced features

4.2.1 Restoring previous versions of a repository

There is no direct option to restore previous versions of a repository. However, a workaround does exist. Right-click on a database in openLCA and select *Repository* > *compare with* > *HEAD revision*. Afterwards, right-click and select overwrite with local changes and commit.

4.2.2 Notifications

Notification settings can be adjusted upon clicking on the user name in the bottom left corner of the LCA Collaboration Server dashboard. Afterwards, select *Notifications* from the entries in the menu on the left side (Figure 4.7).



Group

Notify me when

☑ A group was deleted
☑ A new repository was created
☑ I was added to a group
☑ My role in a group was changed
☑ I was removed from a group
☑ A member was added to a group
☑ A members role in a group was changed
☑ A member was removed from a group

Repository

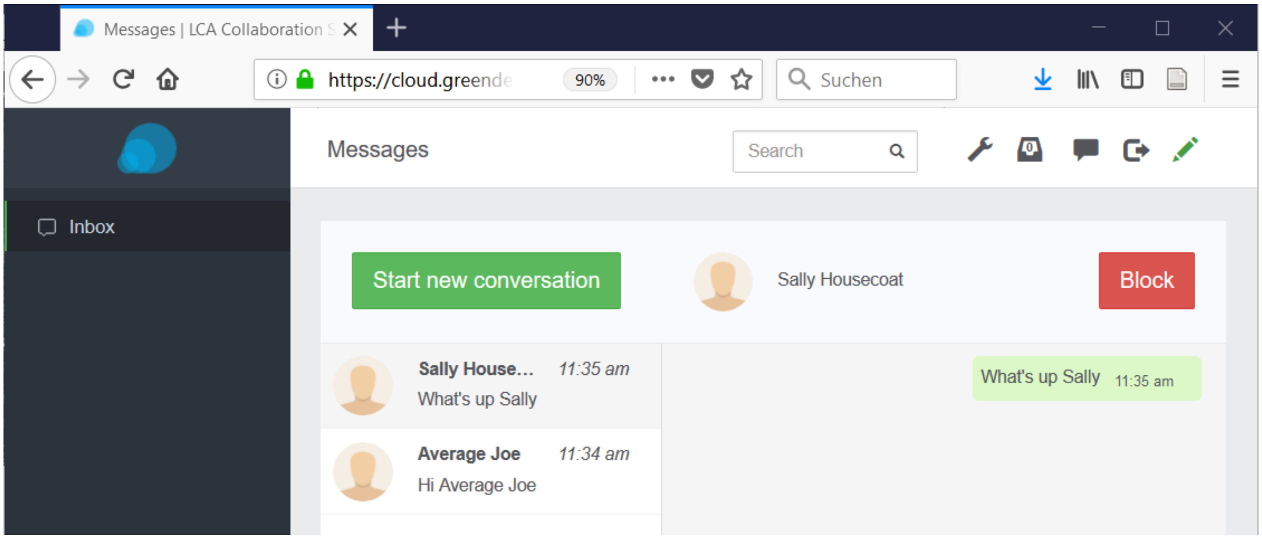
Notify me when

☑ A repository was moved
☑ A repository was deleted
☑ Data was commited to a repository
☑ I was added to a repository
☑ My role in a repository was changed
☑ I was removed from a repository
☑ A member was added to a repository
☑ A members role in a repository was changed
☑ A member was removed from a repository

Figure 4.7: Notifications settings.

4.2.3 Messaging

A messaging feature for conversations among team members is available via the speech bubble in the top right-corner of the LCA Collaboration Server user dashboard (Figure 4.8). To enable the messaging feature click on your profile settings in the bottom-left corner of the LCA Collaboration Server dashboard and select Messaging in the left sidebar.



Sally House... 11:35 am
What's up Sally

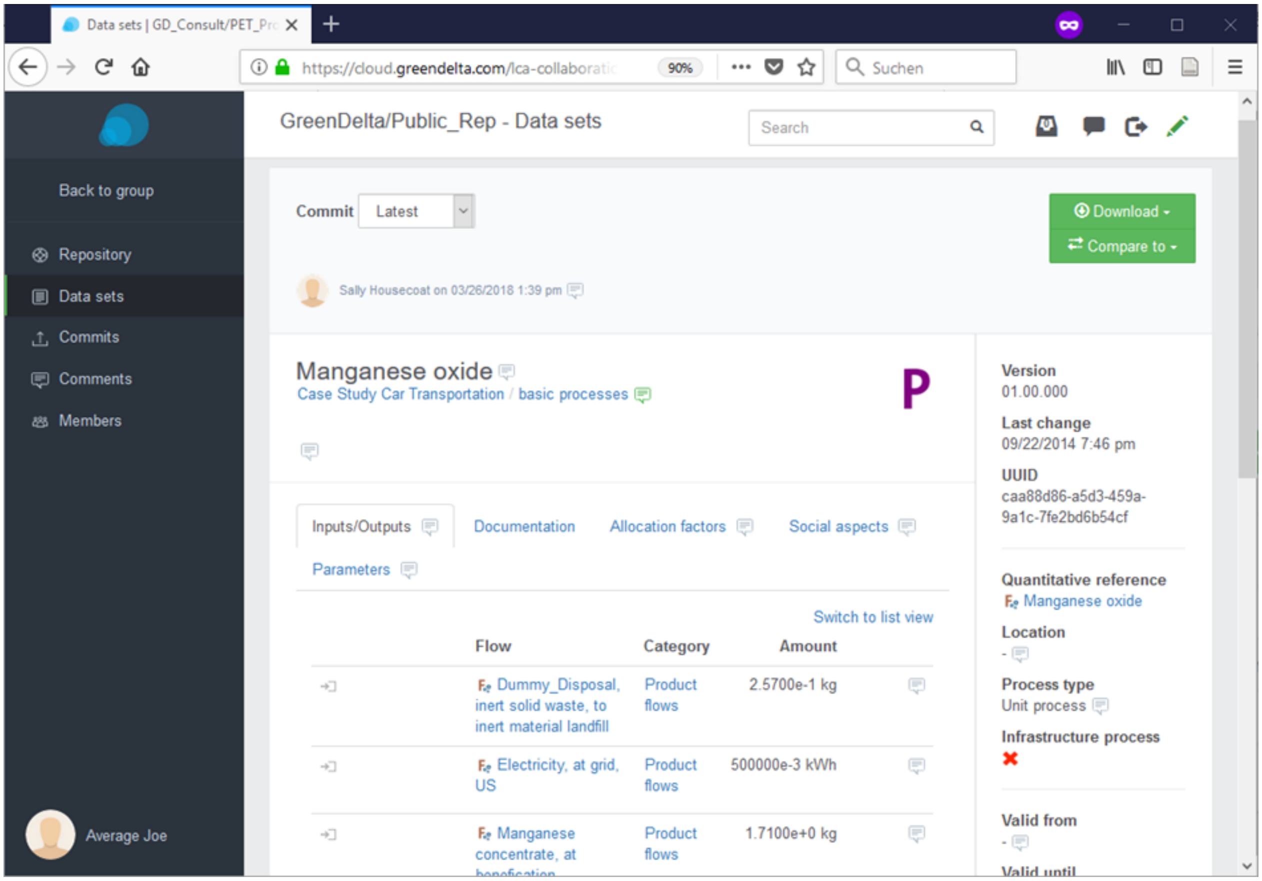
Average Joe 11:34 am
Hi Average Joe

What's up Sally 11:35 am

Figure 4.8: The messaging feature can be accessed via the speech bubble in the top right corner.

4.2.4 Comments and review mode

Specific fields of a data set as well as entire data sets as such can be commented. Comments can be added upon activating the review mode through a click on the pencil in the top-right corner of the LCA Collaboration Server dashboard. Once the review mode has been activated, speech bubbles show up for each item that is commentable (Figure 4.9). Regardless of whether the review mode is active or not, an existing comment is indicated through a green speech bubble (Figure 4.9).
Comments can be released immediately, added as a draft or added for a review by a moderator. An overview of all comments in the LCA Collaboration Server dashboard is available via the left sidebar (Figure 4.10). Comments can also be viewed in openLCA. Right-click on the respective database and navigate to *repository* -> *show comments*. If an item has been commented, a 'Comments' tab shows up in the bottom right corner of the item's tab (Figure 4.12).



Commit: Latest

Sally Housecoat on 03/26/2018 1:39 pm

Manganese oxide
Case Study Car Transportation / basic processes

Flow	Category	Amount
Dummy_Disposal, inert solid waste, to inert material landfill	Product flows	2.5700e-1 kg
Electricity, at grid, US	Product flows	500000e-3 kWh
Manganese concentrate, at beneficiation	Product flows	1.7100e+0 kg

Version
01.00.000

Last change
09/22/2014 7:46 pm

UUID
caa88d86-a5d3-459a-9a1c-7fe2bd6b54cf

Quantitative reference
Manganese oxide

Location
-

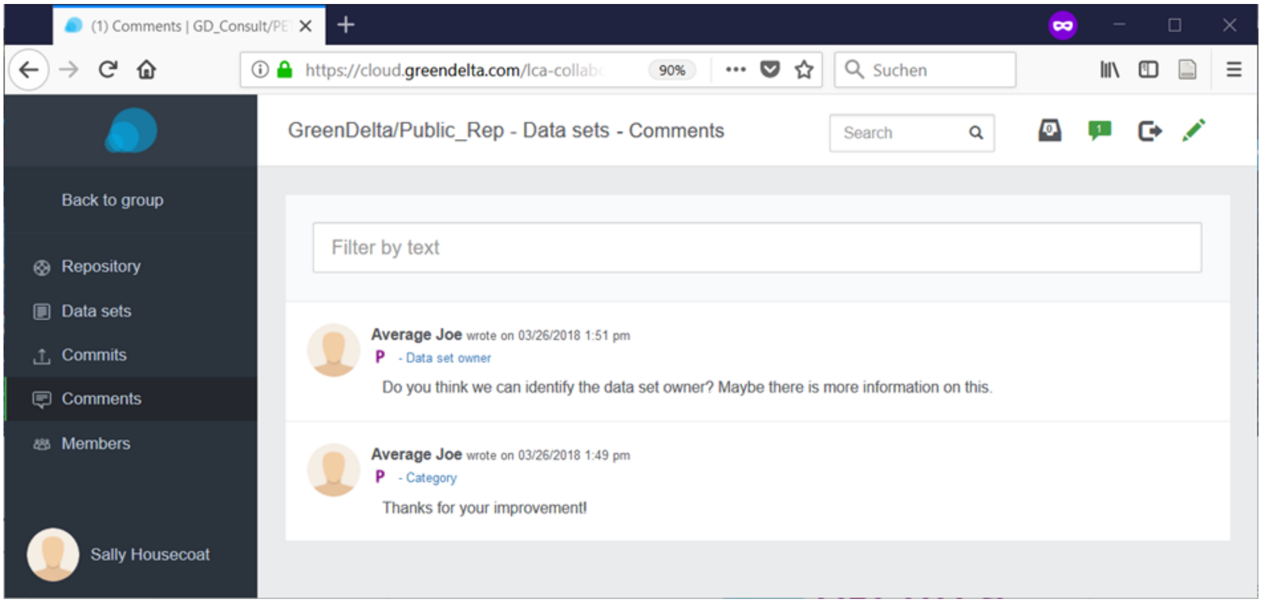
Process type
Unit process

Infrastructure process
✗

Valid from
-

Valid until

Figure 4.9: Upon activating the review mode, speech bubble indicate commentable items. Regardless of whether the review mode is activated or not, commented items are indicated through a green speech bubble.



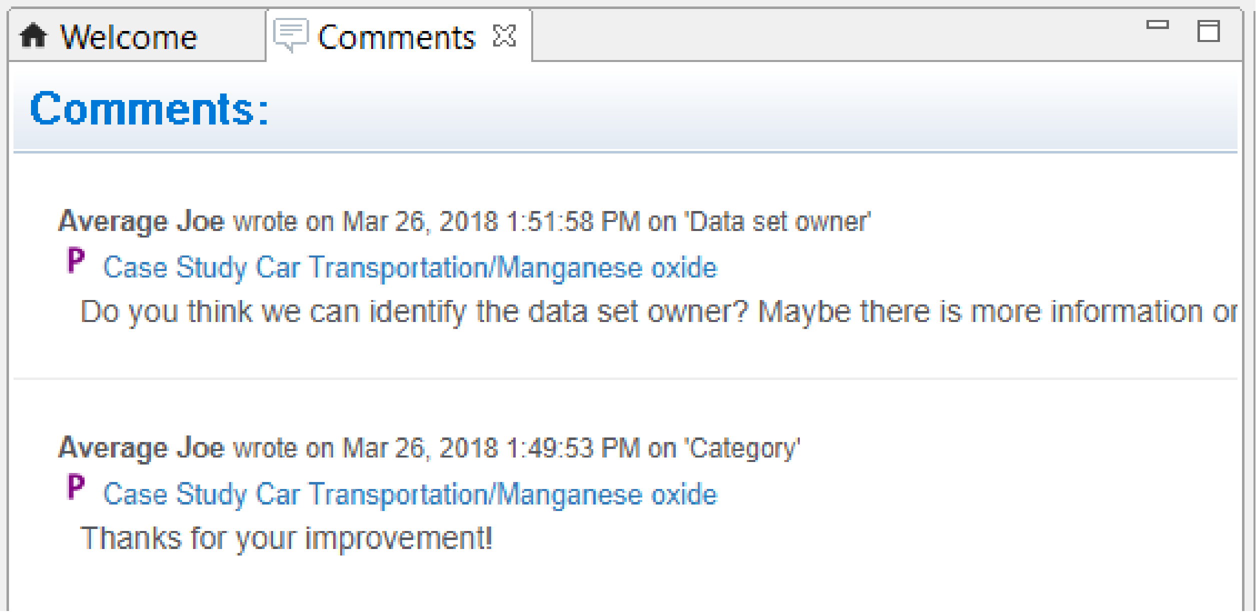
GreenDelta/Public_Rep - Data sets - Comments

Filter by text

Average Joe wrote on 03/26/2018 1:51 pm
P - Data set owner
Do you think we can identify the data set owner? Maybe there is more information on this.

Average Joe wrote on 03/26/2018 1:49 pm
P - Category
Thanks for your improvement!

Figure 4.10: Comment overview in the LCA Collaboration Server dashboard.

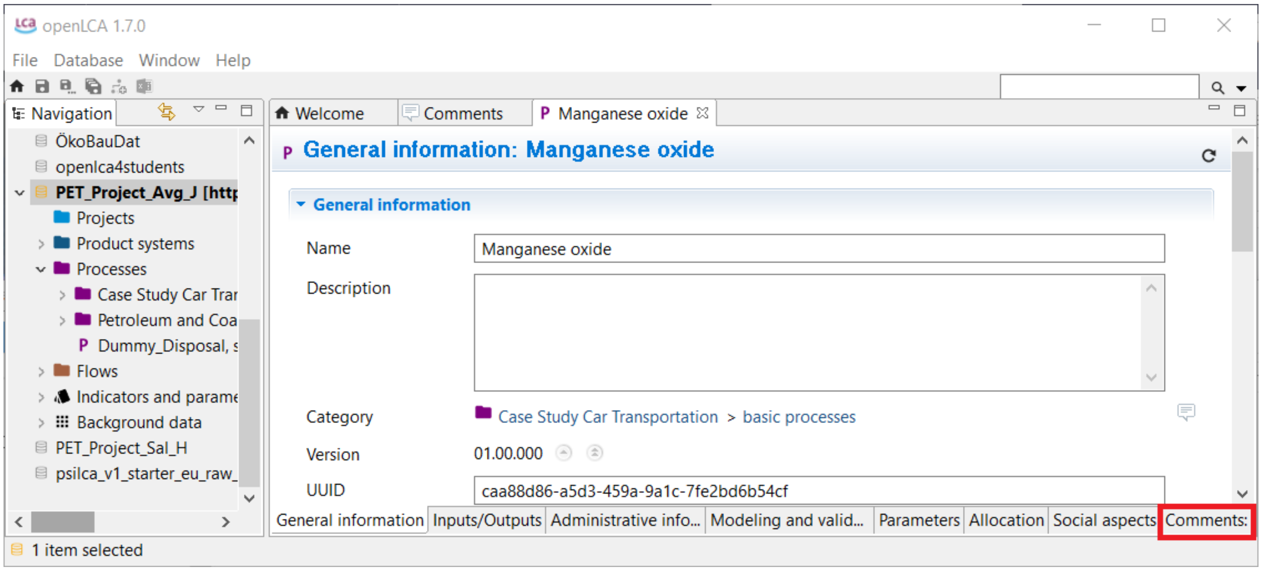


Comments:

Average Joe wrote on Mar 26, 2018 1:51:58 PM on 'Data set owner'
P Case Study Car Transportation/Manganese oxide
Do you think we can identify the data set owner? Maybe there is more information or

Average Joe wrote on Mar 26, 2018 1:49:53 PM on 'Category'
P Case Study Car Transportation/Manganese oxide
Thanks for your improvement!

Figure 4.11: Comment overview in openLCA.



General information: Manganese oxide

General information

Name: Manganese oxide

Description:

Category: Case Study Car Transportation > basic processes

Version: 01.00.000

UUID: caa88d86-a5d3-459a-9a1c-7fe2bd6b54cf

Figure 4.12: If an item has been commented, a 'Comments' tab shows up in the bottom right corner of the item's tab.

4.2.5 Tasks

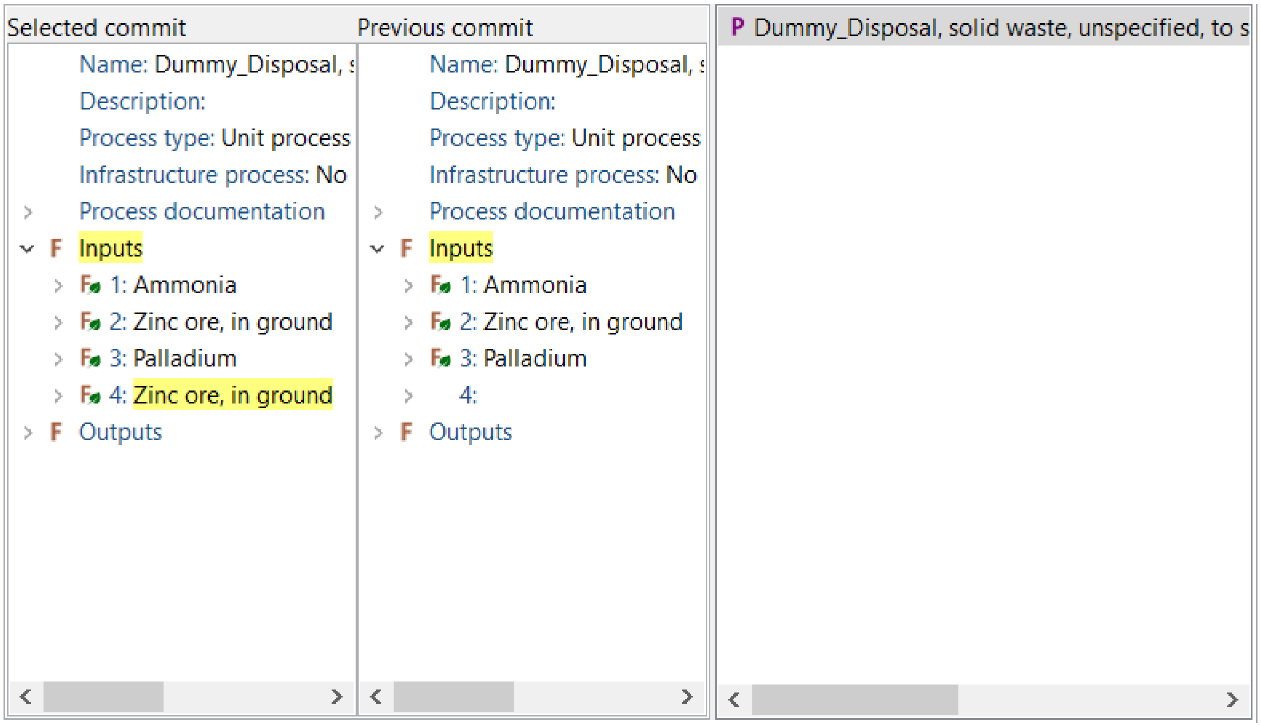
The task system can be accessed via the paper tray symbol in the top-right corner of the LCA Collaboration Server dashboard and allows creating tasks which can be assigned to *Reviewers* (p.8). Proper use of the task system requires at least one user in the team to have *Reviewer* user rights (Table 4.2.5).

Table 4.1: Tasks and their status.

Status	Description
In progress	Awaiting release or moderation (depending on whether a user is *Editor* or *Reviewer*
Completed	Released and/or reviewed
Cancelled	No further action is required

4.2.6 Diff utility

The diff utility in openLCA is a data comparison tool to display differences between commits. It compares data sets in the local database with those in the repository of the LCA Collaboration Server. To open the diff utility right-click on a database and select *Repository > Show in history*. The diff utility supports solving conflicts by identifying differences between the local database and a version of the repository (Figure 4.13).



Selected commit

Name: Dummy_Disposal, s
Description:
Process type: Unit process
Infrastructure process: No
Process documentation
Inputs
1: Ammonia
2: Zinc ore, in ground
3: Palladium
4: Zinc ore, in ground
Outputs

Previous commit

Name: Dummy_Disposal, s
Description:
Process type: Unit process
Infrastructure process: No
Process documentation
Inputs
1: Ammonia
2: Zinc ore, in ground
3: Palladium
4:
Outputs

P Dummy_Disposal, solid waste, unspecified, to s

Figure 4.13: openLCA's Diff utility displays differences between the local version of a database and a version of the repository. In the Figure, the flow *Zinc ore, in ground* has been added to the process *Dummy disposal, ...*

4.3 Expert features

4.3.1 Additional roles

Additional roles are that of the Data Manager and User Manager. Both can be assigned as such by the **Server Administrator**.

- **Data Manager –** A data manger can manage libraries and push data to GLAD[1]
- **User Manager –** A user manager can manage users and teams

Libraries

Libraries are protected data sets that require an additional confirmation to commit changes.

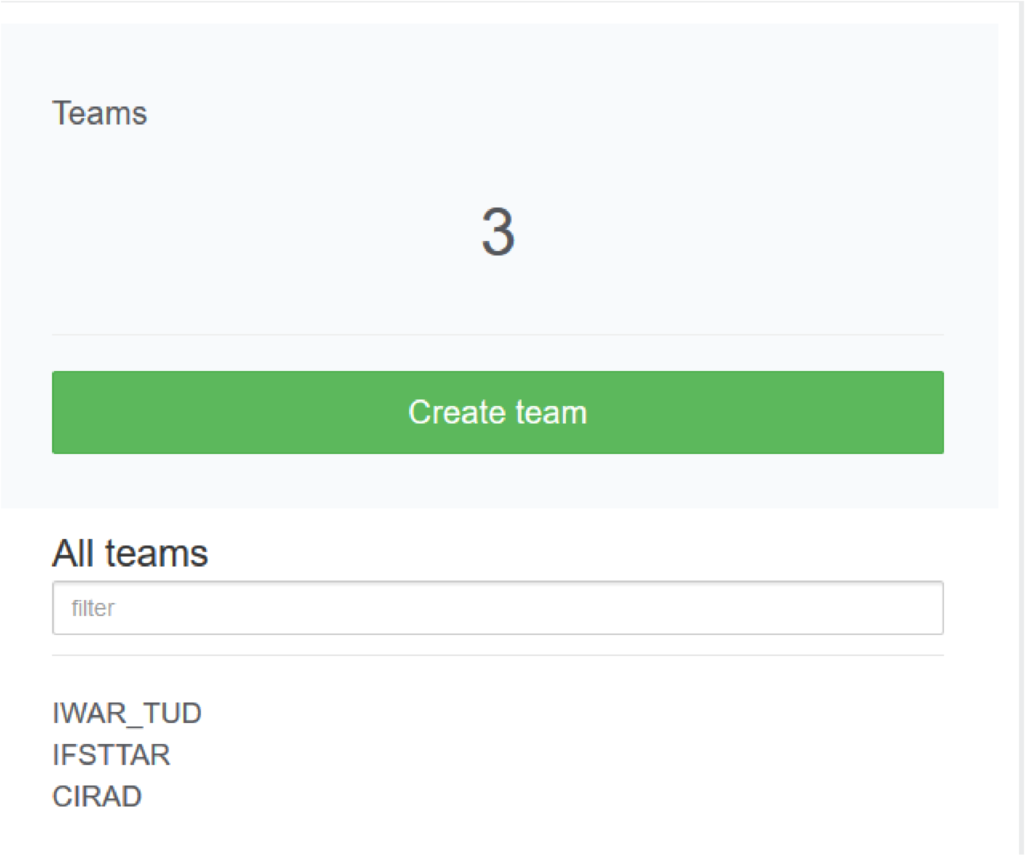
Create users

Users can be created by accessing the admin area via clicking on the wrench in the top-right corner of the LCA Collaboration Server dashboard.

[1] Global LCA Data Access network (http://unep-glad.71.ecedi.fr/)

Teams

Individual users can be combined to a team and teams can be added to groups. A team for users is the equivalent to what groups are for repositories and allow to combine users. Teams can be added to a repository or be recipient in the messenger. Teams are created by accessing the admin area via clicking on the wrench in the top-right corner of the LCA Collaboration Server dashboard (Figure 4.14)[2].



Teams

3

All teams

IWAR_TUD
IFSTTAR
CIRAD

Figure 4.14: Teams are created by accessing the admin area via clicking on the wrench in the top-right corner of the LCA Collaboration Server dashboard.

[2] Please note that only admins can create teams

Conclusion and Outlook

The LCA Collaboration Server is an unprecedented software tool and likewise, offers unprecedented opportunities for LCA practitioners and the LCA community. We have two things in our mind and kindly ask for your support!

5.1 Building a community of best practice - A public repository for LCA models

We appreciate that, with the development of the LCA Collaboration Server, the USDA (p.1) gave GreenDelta the opportunity not only to develop an unprecedented piece of Software but also to enhance the capabilities and reach of LCA practitioners worldwide. This is something we appreciate, and we hope so do the LCA Collaboration Server users. We were wondering how we can pick this up and return something to the LCA community to further promote the application of LCA and LC thinking. What GreenDelta came up with is the vision of a public repository of freely available LCA models under the Creative Commons license[1].
Often, it is challenging for LCA beginners to develop their first complex LCA model and to our knowledge, no global database for free LCA models exists. GreenDelta wants to address this. Therefore, we encourage you to share your LCA model with us for use in a public open access repository. Users will be able to access the repository directly from openLCA and import existing LCA models for educational purposes.

5.2 The LCA Collaboration Server as a nexus for LCA data sets

Currently, LCA practitioners must download LCA data sets for openLCA via the web browser[2] and import them into openLCA. Actually, this procedure is not too complicated but the LCA Collaboration Server offers an opportunity to render the import of LCA data sets into openLCA even more simple. In the long-term, GreenDelta would like to offer commercial and non-commercial LCA data sets directly via the LCA Collaboration Server.

[1] https://creativecommons.org/licenses/
[2] e.g. http://nexus.openlca.org/

Support and Contact

ask.openLCA

ask.openLCA (https://ask.openlca.org/) is a question-and-answer (Q&A) website on Life Cycle Assessment and the support platform for openLCA, openLCA nexus and the openLCA Collaboration Server. We encourage users to use ask.openLCA for non-confidential support requests. However, with ask.openLCA we are building a public knowledge platform for all LCA practitioners regardless of what software they use or LCA-related questions they may have. Don't stop contacting GreenDelta via email for confidential support requests.

6.1 GreenDelta's support services for the LCA Collaboration Server

- Assistance in setting up an LCA Collaboration Server
- Dedicated hosting of an LCA Collaboration Server
- Prioritised LCA Collaboration Server support via email
- Professional training on how to get the best out of the LCA Collaboration Server for your organisation

6.1.1 Demo access

Demo access to the LCA Collaboration Server may be granted upon request.

6.2 For further inquire please contact GreenDelta GmbH

Tel. +49 30 48 496 – 030
Fax +49 30 48 496 – 991
gd@greendelta.com
GreenDelta GmbH
Müllerstrasse 135
D-13357 Berlin, Germany
www.greendelta.com

Privacy

GreenDelta GmbH honours privacy and backs users in protecting confidential data. The LCA Collaboration Server is an independent application and decisions about where to host data as well as access to data are exclusively taken by its users. With the LCA Collaboration Server, users remain in full control of their data and commit to building their own sovereign IT-infrastructure for collaborative development of LCA studies in a distributed team.

Appendix

A.1 Known issues

A.1.1 Redirected requests

openLCA currently does not support handling redirected requests (e.g. reverse proxy)

A.1.2 Fetching deleted data sets

Fetching deleted data sets that were not used in the local database of the user who commits but that are used in the local database of the user who fetches will lead to broken references e.g.

1. *User A* deletes a flow that is not used in the user’s local database and commits the deletion to the LCA Collaboration Server
2. *User B* added a new process in the user’s local database and uses the flow as input
3. When *User B* then fetches the changes of User A. The flow is deleted locally, but the input in the process still refers to the (now) deleted flow

Workaround: If a user is aware that another user is going to fetch a deletion of a (locally) linked element. Modify any field in the element in question (e.g. add a letter in the name and remove it again and save). openLCA will track that the model has changed. When fetching the ”*deletion*” a conflict will come up and the user can manually choose to keep the local element

A.1.3 Error reporting

The Collaboration Server uses *log4j2* for logging. In addition to writing server-side log files, *log4j2* can be configured to send error emails to a specific email account. These emails can either be send from the local server directly or via an external email account. Admins will need to replace the file */WEB-INF/classes/log4j2.xml* in the built war-file. Below is an example configuration for an external email account. For more information see the *log4j2*.

- ***Note:*** If you want to directly report errors to us, please use error@greendelta.com as ’to’ attribute and append your company or institution name to the ’subject’ attribute.
- ***Important:*** The username and email address of the user encountering the error, will be added to the email. If you do not want this information to be send to us, please configure the logging to send the emails to your email address first and forward the email - cleared from personal information - to us.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="WARN">
```

```
        <Appenders>
            <Console name="Console" target="SYSTEM_OUT">
                <PatternLayout pattern="%d{yy-MM-dd HH:mm:ss:SSS} %-5p %c:%L - %m%n" />
            </Console>
            <SMTP name="Mail"
                from="some-account@your-company.com"
                to="error@your-company.com"
                subject="Error on LCA Collaboration Server"
                smtpPassword="the-password"
                smtpUsername="some-account@your-company.com"
                smtpProtocol="smtps"
                smtpPort="465"
                smtpHost="your-email-host-address" />
    </Appenders>
        <Loggers>
            <Root level="info">
                <AppenderRef ref="Console" level="info" />
                <AppenderRef ref="Mail" level="error"/>
            </Root>
        </Loggers>
    </Configuration>
```

A.2 Installation of the LCA Collaboration Server

Please refer to the short- (p.32) or long (p.33) installation instructions and make sure, all hardware and software requirements are fulfilled (p.41 and p.42). Both insructions refer to *Ubuntu 16.04*.

A.2.1 Installation instructions (short)

After installing and starting Elasticsearch you must create the Search Index, the LCA Collaboration Server database (Derby) and data directories. The easiest way to do this is to use the installer application which is a simple Java JAR. Run it e.g. with *java -jar lca-collaboration-installer.jar*.
The installer application will ask for the relevant configuration (these values can be reconfigured by the admin once the application is running). Directories (incl. parent directories) don't need to exist before running the installer which will create the directories itself. If directories do exist though, they must be empty. Otherwise, their creation is skipped.
Note: The OS user whose account is used for running tomcat needs sufficient user rights on the directories. If you are e.g. using a non-root user you will need to change the owner of the directories to the tomcat user (e.g. *chown tomcat8:tomcat8 /opt/collab -R* on Ubuntu with tomcat 8 installed via apt).
Prompted input:

- A directory to create the internal database in (e.g. /opt/collab/database)

- A directory where the data sets for each repository will be stored (e.g. /opt/collab/repositories)
- A directory for “refId” lists for protected libraries (e.g. /opt/collab/libraries)
- The elasticsearch cluster name (e.g. elasticsearch on a fresh installation)
- The elasticsearch host address (e.g. localhost if on the same workstation)
- The index name (e.g. lca-collaboration)
- An email address for the initially created admin user
- A password for the admin user

These initial settings or additional optional settings can be configured in the admin area. You can access it by clicking the wrench icon in the top left (if you have sufficient rights).

A.2.2 Installation instructions (long)

Update OS

```
apt update
apt dist - upgrade
```

Install Java 8

```
add-apt-repository ppa:webupd8team/java
apt update
apt install oracle-java8-installer
```

Install Elasticsearch

```
wget - qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | apt-key add -
echo "deb https://artifacts.elastic.co/packages/6.x/apt stable main" | tee - a
 ↪  /etc/apt / sources.list.d/elastic-6.x.list
apt update
apt install elasticsearch
```

Setup autostart for Elasticserch

```
/bin/systemctl daemon-reload
/bin/systemctl enable elasticsearch.service
```

Configure Elasticsearch JVM heap size

```
nano /etc/elasticsearch/jvm.options
# change -Xms1g to -Xms2g and -Xmx1g to -Xmx2g (for 2 GB of RAM)
service elasticsearch restart
```

Install and configure Tomcat to use port 80

```
apt install tomcat8
service tomcat8 stop
rm /var/lib/tomcat8/webapps/ROOT -r
nano /usr/share/tomcat8/bin/setenv.sh
^^Iexport CATALINA_OPTS="$CATALINA_OPTS -Xms2g"
^^Iexport CATALINA_OPTS="$CATALINA_OPTS -Xmx2g"
nano /etc/tomcat8/server.xml
<?xml version="1.0" encoding="UTF-8"?>
<Server port="8005" shutdown="SHUTDOWN">
<Listener className="org.apache.catalina.startup.VersionLoggerListener"/>
<Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on"/>
<Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener"/>
<Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener"/>
<Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener"/>
<GlobalNamingResources>
<Resource name="UserDatabase" auth="Container"
 ↪  type="org.apache.catalina.UserDatabase" description="User database that can be
 ↪  updated and saved" factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
 ↪  pathname="conf/tomcat-users.xml"/>
</GlobalNamingResources>
<Service name="Catalina">
<Connector port="80" protocol="HTTP/1.1" connectionTimeout="20000"
 ↪  redirectPort="443"/>
<Engine name="Catalina" defaultHost="localhost">
<Realm className="org.apache.catalina.realm.LockOutRealm">
<Realm className="org.apache.catalina.realm.UserDatabaseRealm"
 ↪  resourceName="UserDatabase"/>
</Realm>
<Host name="localhost"  appBase="webapps" unpackWARs="true" autoDeploy="true">
<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
 ↪  prefix="localhost_access_log" suffix=".txt" pattern="%h %l %u %t &quot;%r&quot;
 ↪  %s %b"/>
<Context path="/lca-collaboration">
<Parameter name="app.database" value="/opt/collab/database"/>
```

```
</Context>
</Host>
</Engine>
</Service>
</Server>
```

Bind ports 80

```
nano/etc/default / tomcat8
# change last line from# AUTHBIND = no to AUTHBIND = yes
touch/etc/authbind/byport/80
chown tomcat8/etc/authbind/byport/80
chmod 744 /etc/authbind/byport/80
```

Prepare collaboration server (you will need to upload the lca-collaboration-installer.jar to e.g. /opt and upload the lca-collaboration.war file to /var/lib/tomcat8/webapps)

```
java -jar /opt/lca-collaboration-installer.jar # Example inputs:
    /opt/collab/database
    /opt/collab/repositories
    /opt/collab/libraries
    elasticsearch
    localhost
    lca-collaboration
    your_admin_email@your_company.com
    your_admin_password
chown tomcat8:tomcat8 /opt/collab -R
service tomcat8 start
```

A.3 Install Secure Sockets Layer (SSL)

A.3.1 Update OS

```
apt update
apt dist-upgrade
```

A.3.2 Install Java 8

```
add-apt-repository ppa:webupd8team/java
apt update
apt install oracle-java8-installer
```

A.3.3 Install Elasticsearch

```
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | apt-key add -
echo "deb https://artifacts.elastic.co/packages/6.x/apt stable main" | tee -a
↪ /etc/apt/sources.list.d/elastic-6.x.list
apt update
apt install elasticsearch
```

A.3.4 Setup autostart for Elasticsearch

```
/bin/systemctl daemon-reload
/bin/systemctl enable elasticsearch.service
```

A.3.5 Configure Elasticsearch JVM heap size

```
nano /etc/elasticsearch/jvm.options
# change -Xms1g to -Xms2g and -Xmx1g to -Xmx2g (for 2 GB of RAM)
service elasticsearch restart
```

A.3.6 Install cerbot

```
add-apt-repository ppa:certbot/certbot
apt update
apt install certbot
```

A.3.7 Install and configure Tomact to use port 80 and 443

```
apt install tomcat8
service tomcat8 stop
rm /var/lib/tomcat8/webapps/ROOT -r
nano /usr/share/tomcat8/bin/setenv.sh
export CATALINA_OPTS="$CATALINA_OPTS -Xms2g"
export CATALINA_OPTS="$CATALINA_OPTS -Xmx2g"
nano /etc/tomcat8/server.xml
<?xml version="1.0" encoding="UTF-8"?>
<Server port="8005" shutdown="SHUTDOWN">
<Listener className="org.apache.catalina.startup.VersionLoggerListener"/>
<Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on"/>
<Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener"/>
<Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener"/>
<Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener"/>
<GlobalNamingResources>
<Resource name="UserDatabase" auth="Container"
 ↪  type="org.apache.catalina.UserDatabase" description="User database that can be
 ↪  updated and saved" factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
 ↪  pathname="conf/tomcat-users.xml" />
</GlobalNamingResources>
<Service name="Catalina">
<Connector port="80" protocol="HTTP/1.1" connectionTimeout="20000"
 ↪  redirectPort="443"/>
#
<Connector port="443" SSLEnabled="true" maxThreads="200" scheme="https"
 ↪  secure="true" keystoreFile="/var/lib/tomcat8/keystore.jks"
 ↪  keystorePass="gH3Dx8yA" clientAuth="false" sslProtocol="TLS" compression="on"
 ↪  compressionMinSize="2048" noCompressionUserAgents="gozilla, traviata"
 ↪  compressableMimeType="text/html,text/xml,text/plain,text/css,
 ↪  text/javascript,text/json,application/x-javascript,
 ↪  application/javascript,application/json"/>
<Engine name="Catalina" defaultHost="localhost">
<Realm className="org.apache.catalina.realm.LockOutRealm">
<Realm className="org.apache.catalina.realm.UserDatabaseRealm"
 ↪  resourceName="UserDatabase"/>
</Realm>
<Host name="localhost"  appBase="webapps" unpackWARs="true" autoDeploy="true">
<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
 ↪  prefix="localhost_access_log" suffix=".txt" pattern="%h %l %u %t &quot;%r&quot;
 ↪  %s %b"/>
<Context path="/lca-collaboration">
<Parameter name="app.database" value="/opt/collab/database"/>
</Context>
```

```
</Host>
</Engine>
</Service>
</Server>
```

A.3.8 Bind ports 80 and 443

```
nano /etc/default/tomcat8
# change last line from #AUTHBIND=no to AUTHBIND=yes
touch /etc/authbind/byport/80
chown tomcat8 /etc/authbind/byport/80
chmod 744 /etc/authbind/byport/80
touch /etc/authbind/byport/443
chown tomcat8 /etc/authbind/byport/443
chmod 744 /etc/authbind/byport/443
```

Obtain certificate and create script for creating/updating keystore (replace occurrences of cloud.greendelta.com with your domain)

```
certbot certonly --standalone -d cloud.greendelta.com
echo '#!/bin/bash'>> "/etc/letsencrypt/live/cloud.greendelta.com/update-keystore.sh"
echo "" >> "/etc/letsencrypt/live/cloud.greendelta.com/update-keystore.sh"
echo "rm /etc/letsencrypt/live/cloud.greendelta.com/cert_and_key.p12
 ↪  /var/lib/tomcat8/keystore.jks">>
 ↪  "/etc/letsencrypt/live/cloud.greendelta.com/update-keystore.sh"
echo "eval "'"'"openssl pkcs12 -export -in
 ↪  /etc/letsencrypt/live/cloud.greendelta.com/cert.pem -inkey
 ↪  /etc/letsencrypt/live/cloud.greendelta.com/privkey.pem -out
 ↪  /etc/letsencrypt/live/cloud.greendelta.com/cert_and_key.p12 -name tomcat -CAfile
 ↪  /etc/letsencrypt/live/cloud.greendelta.com/chain.pem -password pass:gH3Dx8yA
 ↪  -caname root"'"'>>
 ↪  "/etc/letsencrypt/live/cloud.greendelta.com/update-keystore.sh"
echo "eval "'"'"keytool -importkeystore -deststorepass gH3Dx8yA -destkeypass
 ↪  gH3Dx8yA -destkeystore /var/lib/tomcat8/keystore.jks -srcstorepass gH3Dx8yA
 ↪  -srckeystore /etc/letsencrypt/live/cloud.greendelta.com/cert_and_key.p12
 ↪  -srcstoretype PKCS12 -alias tomcat"'"'>>
 ↪  "/etc/letsencrypt/live/cloud.greendelta.com/update-keystore.sh"
echo "eval "'"'"keytool -import -trustcacerts -alias root -file
 ↪  /etc/letsencrypt/live/cloud.greendelta.com/chain.pem -storepass gH3Dx8yA
 ↪  -keystore /var/lib/tomcat8/keystore.jks"'"'>>
 ↪  "/etc/letsencrypt/live/cloud.greendelta.com/update-keystore.sh"
```

```
echo "eval "'"'"keytool -importkeystore -srckeystore /var/lib/tomcat8/keystore.jks
 ↪  -srcstorepass gH3Dx8yA -destkeystore /var/lib/tomcat8/keystore.jks
 ↪  -deststorepass gH3Dx8yA -deststoretype pkcs12"'"'>>
 ↪  "/etc/letsencrypt/live/cloud.greendelta.com/update-keystore.sh"
echo "chown root:tomcat8 /var/lib/tomcat8/keystore.jks">>
 ↪  "/etc/letsencrypt/live/cloud.greendelta.com/update-keystore.sh"
echo "rm /var/lib/tomcat8/keystore.jks.old">>
 ↪  "/etc/letsencrypt/live/cloud.greendelta.com/update-keystore.sh"
chmod +x /etc/letsencrypt/live/cloud.greendelta.com/update-keystore.sh
/etc/letsencrypt/live/cloud.greendelta.com/update-keystore.sh
```

Prepare collaboration server (You will need to upload the lca-collaboration-installer.jar to e.g. /opt and upload the lca-collaboration.war file to /var/lib/tomcat8/webapps).

```
java -jar /opt/lca-collaboration-installer.jar # Example inputs:
    /opt/collab/database
    /opt/collab/repositories
    /opt/collab/libraries
    elasticsearch
    localhost
    lca-collaboration
    your_admin_email@your_company.com
    your_admin_password
chown tomcat8:tomcat8 /opt/collab -R
service tomcat8 start
```

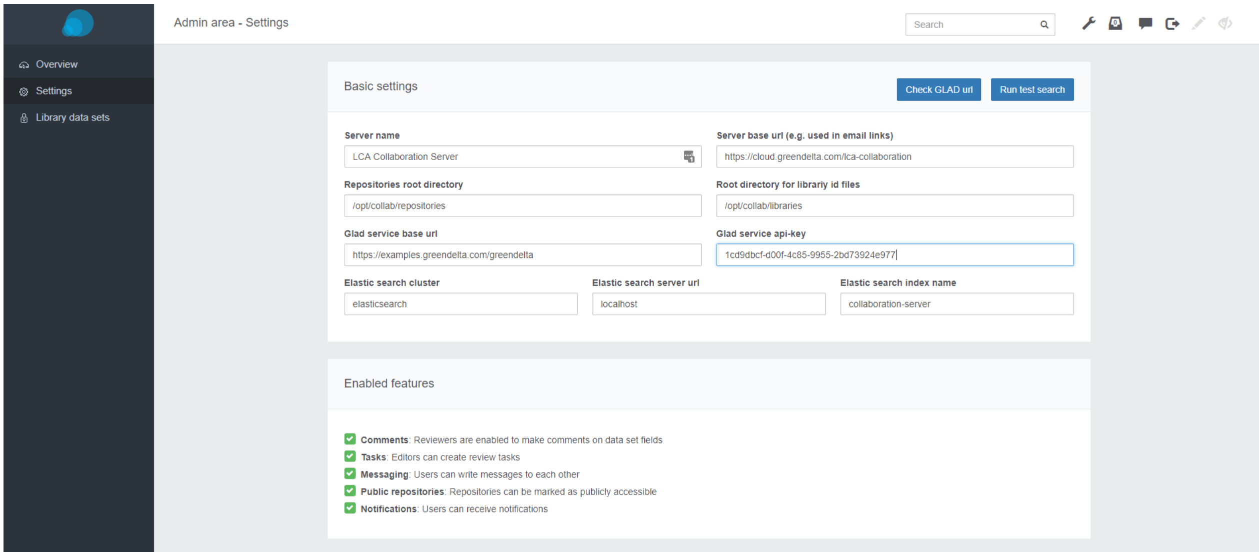
A.4 Configuration

If you want to run the application in the server root context or another context path then */lca-collaboration/*, you will need to adjust the *<base href="/lca-collaboration/"/>* with your context path, e.g. *<base href="/another-path/"/>* in the *index.html*, *login.html* and *imprint.html* files located in the root of the war file.

Before starting the application, you will need to configure the database directory, either as a system variable or as a context parameter, named app.database e.g. on Linux either set an environment variable *export app.database=/opt/collab/database* or in tomcat 8 server.xml (for each instance):

```
<Host name="localhost"  appBase="webapps"
       unpackWARs="true" autoDeploy="true">
  <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
         prefix="localhost_access_log" suffix=".txt"
         pattern="%h %l %u %t &quot;%r&quot; %s %b"/>
  <Context path="/lca-collaboration">
    <Parameter name="app.database" value="/opt/collab/database"/>
  </Context>
</Host>
```

Changing settings in the web application



Admin area - Settings

Basic settings

Server name: LCA Collaboration Server

Server base url (e.g. used in email links): https://cloud.greendelta.com/lca-collaboration

Repositories root directory: /opt/collab/repositories

Root directory for librariy id files: /opt/collab/libraries

Glad service base url: https://examples.greendelta.com/greendelta

Glad service api-key: 1cd9dbcf-d00f-4c85-9955-2bd73924e977

Elastic search cluster: elasticsearch

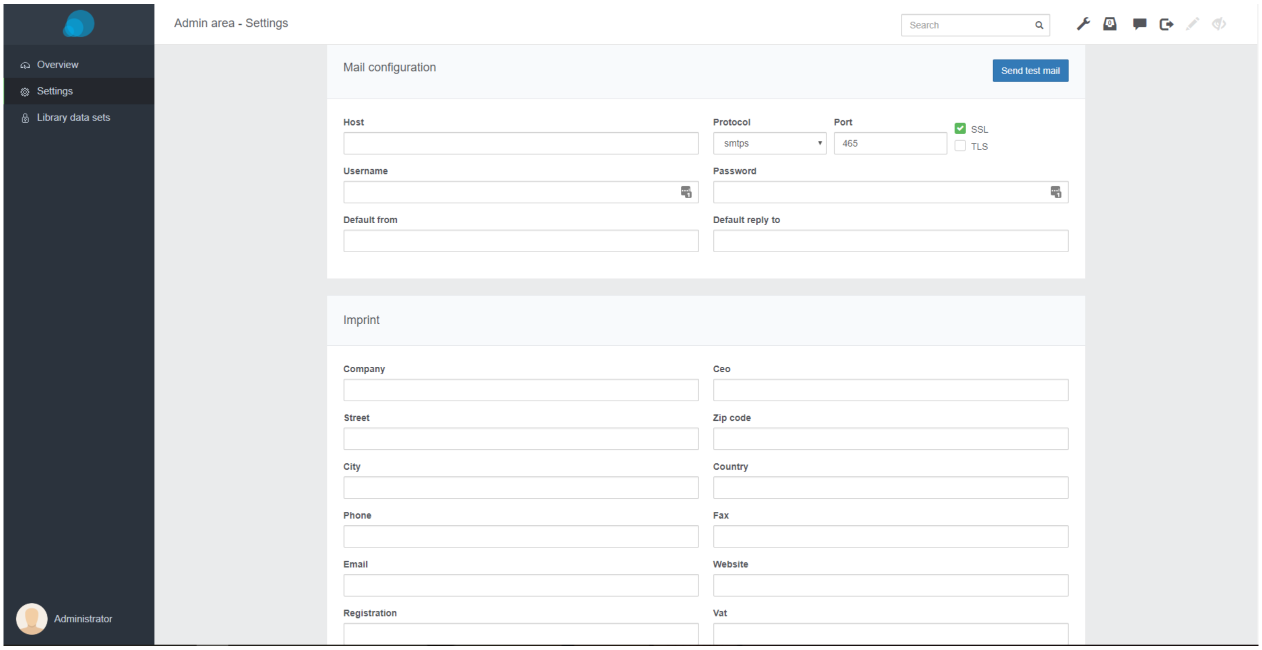
Elastic search server url: localhost

Elastic search index name: collaboration-server

Enabled features

- ☑ **Comments**: Reviewers are enabled to make comments on data set fields
- ☑ **Tasks**: Editors can create review tasks
- ☑ **Messaging**: Users can write messages to each other
- ☑ **Public repositories**: Repositories can be marked as publicly accessible
- ☑ **Notifications**: Users can receive notifications

Figure A.1: Admin area basic settings and enabled features



Admin area - Settings

Mail configuration

Host:

Protocol: smtps

Port: 465

☑ SSL

□ TLS

Username:

Password:

Default from:

Default reply to:

Imprint

Company:

Ceo:

Street:

Zip code:

City:

Country:

Phone:

Fax:

Email:

Website:

Registration:

Vat:

Figure A.2: Admin area mail configuration and imprint

A.4.1 Basic settings

Servername: The Server name, used for two-factor-authentication (optional)
Server base url: The base url used when linking pages in the notification emails
Repositories root directory: In this directory, the repository data sets will be stored, this will need extended disk space, depending on the amount of data sets committed.
Root directory for library id files: In this directory, configured library data set ref ids will be stored
Glad service base url: The base url to an external GLAD service
Glad service api key: To push data set descriptors to the GLAD service, an API key is required, which can be set here.
Elasticsearch cluster: The cluster name of your elasticsearch installation (default: elasticsearch)
Elasticsearch server url: The host address of the elasticsearch installation (default: localhost)
Elasticsearch index name: The name of the elasicsearch index used for this collaboration server instance (default: lca-collaboration)

A.4.2 Enabled features

Some of the collaboration server's advanced features can be disabled, this includes: Comments, Tasks, Messaging, Public repositories and Notifications (Figure A.2 p.40).

A.4.3 Mail configuration

To be able to use notifications, you need to configure an email account to send from. You can use an existing smpt email account (Figure A.1 p.40).

A.5 Hardware and Software requirements

A.5.1 Requirements for openLCA

- Windows[1]
  - CPU with 2 GHz or higher
  - 1 GB RAM (for analysing product systems with 2,500 processes, such as ecoinvent 2)
  - >3 GB RAM (for analysing product systems such as ecoinvent 3)
  - 6 GB RAM (for analysing product systems such as ecoinvent 3.4 or Psilca for social LCA)
  - 500 MB free hard disk space + space for databases (e.g. ecoinvent 3 requires 250MB)
- Mac OS[2]
  - CPU with 2 GHz or higher

[1] Software, required for the "Projects" feature: Windows 64 bit (for modern browser support): Microsoft Visual C++ 2010 Redistributable package (x64) http://www.microsoft.com/de-de/download/details.aspx?id=14632

[2] Software, required: Java in version 8; install Java before beginning with the openLCA installation (Java SE Development Kit for Mac OS) http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.htm

  - 1 GB RAM (for analysing product systems with 2,500 processes, such as ecoinvent 2)
  - >3 GB RAM (for analysing product systems such as ecoinvent 3)
  - 500 MB free hard disk space + space for databases (e.g. ecoinvent 3 requires 250MB)

- Linux[3]
  - CPU with 2 GHz or higher
  - 1 GB RAM (for analysing product systems with 2,500 processes, such as ecoinvent 2)
  - >3 GB RAM (for analysing product systems such as ecoinvent 3)
  - 500 MB free hard disk space + space

A.5.2 Requirements for the LCA Collaboration Server

Please note that this is a draft version of the LCA Collaboration Server manual. Hardware requirements will follow.

Minimum:

CPU: DualCore CPU or better

RAM (Elasticsearch/LCA CS): 2GB/2GB1[4]

Disk: HDD2 Test The disk space depends mostly on the amount of data you want to be able to host. The application itself only requires <100MB.

Recommended:

CPU: QuadCore CPU or better

RAM (Elasticsearch/LCA CS): 4GB/8GB9

Disk: SSD10[4]

Software requirements The LCA Collaboration Server is a web application based on Java Servlet. We recommend the use of Tomcat 8 in combination with Java 8. Most testing has been done under use of this configuration (for both Windows 10 and Ubuntu/Debian Linux).

The LCA Collaboration Server depends on Elasticsearch. Since version 6 it does not support JVM internal local nodes. Therefore, Elasticsearch must be installed and run separately.

[3] It is recommended to install libgfortran3 for high performance calculations

[4] Each Elasticsearch and the LCA CS will need separate RAM. The requirements depend mostly on the number of concurrent requests for fetching and committing data.

GreenDelta GmbH

Müllerstrasse 135
D-13357 Berlin, Germany
www.greendelta.com

Tel. +49 30 48 496 – 030
Fax +49 30 48 496 – 991
gd@greendelta.com

GreenDelta
openLCA
@openLCA
ask.openLCA

GreenDelta
sustainability consulting + software