LCA Collaboration Server v.1.3

User Manual and Introduction

Version: 1.0.1

Authors: Julia Cilleruelo Palomero, Sebastian Greve, Andreas Ciroth

Date: 08 February 2022

GreenDelta GmbH

Kaiserdamm 13
14057 Berlin, Germany
www.greendelta.com

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

GreenDelta  openLCA
Contents

Contents .................................................................................................................................................. i
List of Figures ....................................................................................................................................... iv
Glossary ................................................................................................................................................ vi
Abbreviations ...................................................................................................................................... vi
1. Introduction ...................................................................................................................................... 1
  1.1 How can the LCA Collaboration Server be for free? ..................................................................... 1
  1.2 What are use cases of the LCA Collaboration Server? ................................................................... 1
2. First steps .......................................................................................................................................... 2
  2.1 Install the LCA Collaboration Server on your server .................................................................... 2
  2.2 Hosting and further support ............................................................................................................ 2
  2.3 Getting familiar with the LCA Collaboration Server dashboard .................................................. 3
    2.3.1 Expanding on the repository dashboard .................................................................................. 5
    2.3.2 Expanding on the admin area dashboard ............................................................................... 8
    2.3.3 Groups and user roles ............................................................................................................ 9
3. How-To: Basic Workflow .................................................................................................................. 10
  3.1 Repositories - Create a new repository for your working group .................................................. 10
  3.2 How to connect and disconnect the LCA CS to your openLCA and database ............................. 10
    Connecting openLCA to the LCA CS ............................................................................................ 10
    Connecting a database in openLCA to the LCA CS ..................................................................... 11
  3.3 Uploading the database to the LCA collaboration Server for the first time ................................ 12
  3.4 Committing data sets to the connected repository ...................................................................... 13
  3.5 Fetching data sets from the repository ........................................................................................ 14
  3.6 The workflow is always linear ..................................................................................................... 17
  3.7 Conflicts ....................................................................................................................................... 17
  3.8 Pitfalls .......................................................................................................................................... 17
    3.8.1 Validating databases ............................................................................................................. 18
    3.8.2 Example 1 - Missing interlinked flow ................................................................................... 18
    3.8.3 Example 2 - Modified flow property .................................................................................... 18
4 Features ............................................................................................................................................. 19
  4.1 Basic features .................................................................................................................................. 19
    4.1.1 Commit history ................................................................................................................................. 19
    4.1.2 Checkout commits ................................................................................................................................. 19
    4.1.3 Search function and filter (LCA Collaboration Server dashboard & openLCA) ......................... 20
    4.1.4 Download of whole repositories or selected datasets ...................................................................... 22
  4.2 Advanced features ............................................................................................................................ 22
    4.2.1 Managing repositories .................................................................................................................... 22
    4.2.2 Restoring previous versions of a dataset ......................................................................................... 23
    4.2.3 Notifications .................................................................................................................................... 23
    4.2.4 Messaging ....................................................................................................................................... 24
    4.2.5 Comments and review mode ........................................................................................................... 24
    4.2.6 Tasks .............................................................................................................................................. 26
    4.2.7 Diff utility ....................................................................................................................................... 27
  4.3 Expert features ................................................................................................................................ 27
    4.3.1 Additional roles ............................................................................................................................... 27
5 Conclusion and Outlook .......................................................................................................................... 29
  5.1 Building a community of best practice - A public repository for LCA models ................................. 29
  5.2 The LCA Collaboration Server as a nexus for LCA data sets ......................................................... 29
6 Support and Contact .............................................................................................................................. 30
  6.1 Prioritised support via the GreenDelta HelpDesk ............................................................................. 30
  6.2 Public support via ask.openLCA.org ............................................................................................... 30
  6.3 Demonstration of the LCA Collaboration Server ............................................................................ 30
  6.4 Contact GreenDelta GmbH ............................................................................................................. 30
Privacy .................................................................................................................................................. 31
Appendix ................................................................................................................................................ 32
  A.1 Known issues ................................................................................................................................... 32
    A.1.1 Redirected requests .......................................................................................................................... 32
    A.1.2 Fetching deleted data sets ............................................................................................................. 32
    A.1.3 Error reporting ............................................................................................................................... 32
List of Figures

Figure 2-1 Hosting options offered for the LCA Collaboration Server ................................................................. 3
Figure 2-2 The LCA Collaboration Server dashboard .................................................................................................. 4
Figure 2-3 User’s dashboard ........................................................................................................................................... 5
Figure 2-4 Repository dashboard, of which Average Joe has an “owner” role .............................................................. 6
Figure 2-5 Download and export of data sets .................................................................................................................. 8
Figure 2-6 Admin area in the LCA Collaboration Server ................................................................................................. 8
Figure 3-1 Preferences window in openLCA ................................................................................................................... 11
Figure 3-2 add repository from openLCA to the LCA CS .................................................................................................. 12
Figure 3-3 Connected database ..................................................................................................................................... 12
Figure 3-4 Right-click on any item in the database to initiate a commit .......................................................................... 13
Figure 3-5 Committing changes to a repository requires a ’commit message’ ................................................................. 14
Figure 3-6 Fetching a repository from the LCA Collaboration Server in openLCA ......................................................... 15
Figure 3-7 Simple list of changes that need to be fetched ............................................................................................... 16
Figure 3-8 Structured list of all changes that need to be fetched .................................................................................... 16
Figure 3-9 The workflow of a commit is always linear ................................................................................................... 17
Figure 3-10 ’Validate’ allows to check whether all linkages within a database are working ...................................... 18
Figure 3-11 Success message when validating the database .......................................................................................... 18
Figure 4-1 History of commits in the LCA Collaboration Server dashboard ................................................................. 19
Figure 4-2 History of commits in openLCA .................................................................................................................... 19
Figure 4-3 A checkout replaces the local openLCA database with a specific commit ................................................. 20
Figure 4-4 Grouped search results ............................................................................................................................... 21
Figure 4-5 Ungrouped search results ............................................................................................................................ 21
Figure 4-6 Download and export of data sets ................................................................................................................ 22
Figure 4-7 Notifications settings ....................................................................................................................................... 23
Figure 4-8 The messaging feature can be accessed via the speech bubble in the top right corner .......................... 24
Figure 4-9 Upon activating the review mode, comment bubbles can be seen in different locations. Regardless of whether the review mode is activated or not, commented items are indicated through a green speech bubble .................................................................................................................................. 25
Figure 4-10 Comment overview in the LCA Collaboration Server dashboard ............................................................ 25
Figure 4-11 Comment overview in openLCA ................................................................. 26

Figure 4-12 If an item has been commented, a 'Comments' tab shows up in the bottom right corner of the item's tab ......................................................................................................................... 26

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, ..........................................................................................................................27

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 ........................................................................................................... 28
Glossary

Commit - A **commit** refers to contributing a package of data sets from a user’s local database in openLCA to a repository on the LCA Collaboration Server.

Fetch - To **fetch** refers to updating a user’s local database in openLCA with changes from a repository on the LCA Collaboration Server.

Repository - 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.

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

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

Checkout - A **checkout** is the recovery of a previous commit from a repository. During a **checkout**, the local database in openLCA is replaced with a selected version of the repository from the LCA Collaboration Server.

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!
1. Introduction

With the general advancement of Life Cycle Assessment (LCA) and the spreading of Life Cycle Thinking (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 an external reviewer is of interest for anybody who wants to create a consistent LCA database.

The LCA Collaboration Server, now in version 1.3, is a server application that complements openLCA (the LCA desktop application). It is available for free, and support is available on demand.

Since version 1.7, openLCA 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. OpenLCA users are now able to work simultaneously on an LCA study while tracking each other’s changes along synchronised databases.

1.1 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. 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 section 5 to see what we have in mind.

1.2 What are use cases of the LCA Collaboration Server?

The following are some of the use cases covered by the LCA Collaboration Server:

1) A team working on the same database/LCA model, and wanting to synchronise their data.

2) A single LCA practitioner working on a database/LCA model from different devices, and wanting to synchronise their data.

---

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/
3) A personal documentation backup system with change control, where one can comment blocks of changes to document your work.

4) Straightforward 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.

5) Publication of data, where others can view or download publicly available data on the LCA Collaboration Server and import them directly into openLCA without registration.

6) Creation of consistent and verified databases with the option to check previous version.

2. First steps

The LCA Collaboration Server (LCA CS) is an open-source tool available for free. For that reason, it can be downloaded, installed, and maintained by the user alone, see section 2.1. However, not all users have the IT infrastructure to host their own LCA Collaboration Server or do not want to bother with its installation and operation. For that case, GreenDelta also offers hosting solutions and support, see section 2.2.

This section also explains how to make the link between the openLCA in your machine and the LCA CS, the dashboard of the LCA CS, and the types of users and roles available for this tool.

2.1 Install the LCA Collaboration Server on your server

With this option the user can use the LCA CS for free, and has to install and maintain it themselves. Here are the steps:

1. Download the LCA Collaboration Server from https://www.openlca.org/download/

2. Install a local instance of the LCA Collaboration Server on your server
   - Hardware requirements for the LCA Collaboration Server (p.36 in the Appendix)
   - Installation - follow: https://www.openlca.org/lca-collaboration-server-1-3-0-installation-guide

2.2 Hosting and further support

If the user wants to start using the tool directly without having to instal or maintain it, GreenDelta can host the user’s server. The hosting plans are described in Figure 2-1, and it can be purchased in openLCA Nexus: https://nexus.openlca.org/service/LCA%20Collaboration%20Server%20Hosting.

For users and administrators of the LCA Collaboration Server who wish to receive guaranteed and prioritised professional support in handling their LCA Collaboration Server, GreenDelta offers paid support via the GreenDelta HelpDesk: https://nexus.openlca.org/service/openLCA%20Support%20(help%20desk). Public user-to-user support is available via https://ask.openlca.org/.
### 2.3 Getting familiar with the LCA Collaboration Server dashboard

Figure 2-2 shows the dashboard of the LCA Collaboration Server, this is the first page after login. The icons in the top right corner give are explained in Table 2-1.

---

To the left of the dashboard we have a menu:

**Activities tab:** displays the latest activities relating to the repositories the user is involved with. It is a log of the commits, tasks and comments.
Repositories tab: is the central place for the repositories\(^4\) the user is involved with. Repositories can be managed from here, including creating a new repository, or importing one.

Groups tab: holds the groups that the user is part of. A group is made up of a team of users that works on a number of repositories. For example, if a new member joins an LCA team working with 5 repositories in the LCA Collaboration Server, this new user can join the “group” containing those 5 repositories and have access to all of them at once instead of having to be added to each of the repositories individually. Once a group is entered, the user can add tags, or add and edit the members, depending on their user role (section 2.3.3).

Tags: this tab brings the user to the tag cloud (experimental feature).

Clicking on the user’s name in the bottom left corner of the LCA Collaboration Server dashboard opens the user dashboard, see Figure 2-3. This gives access to account properties, messaging and notification settings. The dashboard may look slightly different for members with different user rights.

![User's dashboard](image)

**Figure 2-3 User's dashboard**

### 2.3.1 Expanding on the repository dashboard

After clicking on a repository in the repository library (Figure 2-2 The LCA Collaboration Server dashboard), the following dashboard is displayed:

\(^4\) 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.
The first tab, **Repository**, shows the information of the repository. Further down, there is the “Metadata” section, a great opportunity to document the database. The repository **ID** is the “**groupname/repositoryname**”. The repository name can be changed following:

**Move -> Name**

**Caution**, this will also change the ID of the repository. To change the name without changing the ID, add a “Label”, below “Group” in Figure 2-4, to override the repository name on the repository library, without changing the ID.

The user can add **tags** which further classify repositories through this tab. They can look for repositories with specific tags through the search function.

**Activities** logs the activities of the repository (i.e. commits or comments to the repository), which are also displayed in the “Activities” of the main menu.
The **Data sets** tab allows to see all of the data in the repository, and also to download and export repositories and datasets through the download feature shown in Figure 2-5. It allows to download individual versions (commits) of a repository as JSON-LD or ILCD.

The **Commits** tab logs all of the commits to the repository, with the option to download the commit history too. The data committed can always be commented too.

The **Comments** tab holds all of the comment history of comments to committed processes.

Adding *tags* to repositories can be done in **Repository tab**. Individual datasets can also be tagged, as well as full repositories.

The **Members** tab holds a record of both the members of the repository, and the members of the repository through a group. Members can be added and deleted here, and also the user roles described in Section 2.3.3 can be changed.

Furthermore, the “owner” of the repository will be able to see the following managing tools for their repository:

More on user roles will be explained in section 2.3.3, and these buttons will also be explained in section 4.2.1.
2.3.2 Expanding on the admin area dashboard

One can access the admin area by clicking on the icon. The icon can only be seen by data managers, user managers, or server administrator. The regular user cannot see it. From the admin area, one can create, delete or manage repositories, users, groups of repositories and teams of users. In Settings the admin user can configure the Collaboration Server and enable or disable features.
2.3.3 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 2-2). A summary of the differences is resumed in Table 2-2.

<table>
<thead>
<tr>
<th>Rights</th>
<th>Reader</th>
<th>Contributor</th>
<th>Reviewer</th>
<th>Editor</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read repositories and fetch contents</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Commit data to repositories</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Comment specific fields of data sets</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Review comments</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Manage comments</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Manage reviews</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Create repositories</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Edit repository members</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Adjust settings</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Move repositories</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Delete repositories</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Generate change log</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Data manager (administrate libraries) and user manager (create users and teams) are other additional roles; p.27
3 How-To: Basic Workflow

3.1 Repositories - Create a new repository for your working group

Go to Repository in the main menu. Click to create or import a new repository. There are several ways to upload the database the user wants to work with to the repository in the Collaboration Server:

1. Create new repository and commit the database from openLCA (section 3.3)
2. Import JSON-LD file from a previously exported JSON-LD file from the desired database in openLCA.
3. Import repository – this option imports a previously exported repository zip file (e.g. from another CS, or maybe a backup file)

It is very important that all of the users that will be using the repository have the same version of the database that is uploaded to the repository. Users can also create a new database in their openLCA and ‘fetch’ the database in the repository.

When creating a repository, it is possible to limit the repository’s maximum size on the server (0 = unlimited). For more about the repository board, see section 2.3.1.

3.2 How to connect and disconnect the LCA CS to your openLCA and database

Connecting openLCA to the LCA CS

The user must have openLCA\(^6\) installed in their computer, and a server holding the LCA Collaboration Server assigned.

On your openLCA, to open the window shown Figure 3-1, click

file > settings > collaboration,

Check enable collaboration. Right click on the table in the same window and click create new. Please enter your server url, your user name, password, and click is default to proceed. Click Apply and Close.

Tip: if it doesn’t work, double check you have the correct url, and you haven’t accidentally included any extra spaces or similar. The right url should be displayed after hovering over the logo in your LCA CS in the web.

Checked against libraries – checks if the user can commit specific data to that library. The server admin, or another high user role, can decide to choose certain data in the repository that should not be committed further.

---

\(^6\)To download openLCA visit: [https://www.openlca.org/download/](https://www.openlca.org/download/)
Check referenced changes – suggests new commits to the library that might otherwise cause an error. For example, the user has added a new flow to an existing process, where the flow is not yet in the LCA CS repository. When the process is committed, the flow should also be committed. This is reminded by the software, if this option is checked.

**Connecting a database in openLCA to the LCA CS**

Once openLCA is connected to the LCA CS, the user can connect the database to a specific repository by right clicking on the database in openLCA > repository > #configure...

Right click on the empty table, and click *add* to add the repository (Figure 3-2). Then, click *OK* and set the database as *Active*. After clicking OK one last time, openLCA connects to the repository and rebuilds the index of the selected database. A connected database looks like the one shown in Figure 3-3, where the URL is shown.

![Figure 3-1 Preferences window in openLCA](image-url)
To disconnect from a repository right-click on the respective database and select repository > disconnect from repository.

### 3.3 Uploading the database to the LCA collaboration Server for the first time

In the case that the user has created a new (empty) repository in the Collaboration Server, the database is added to the LCA CS by right clicking on the database name in openLCA > Repository > Commit. **The user should always include a commit message to be able to click commit.**
Uploading the whole database (first commit) to the LCA CS may take around **15 minutes**. 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.

### 3.4 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 3-4).

![Figure 3-4 Right-click on any item in the database to initiate a commit](image)

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. The speed of the commit depends on network connection and the size of the database.
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 (see section 3.8).

**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.

### 3.5 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 3-6).
Two windows will pop up: The first one lists all changes of each individual commit since the last fetch (Figure 3-7). The second one lists all commits since the last fetch in a merged and structured list (Figure 3-8).
Figure 3.7 Simple list of changes that need to be fetched

Figure 3.8 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 (see section 3.8).

### 3.6 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 3-9).

![Figure 3-9 The workflow of a commit is always linear](image)

### 3.7 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 (see section 4.2.7) assists in detecting commits by other users and to identify potential conflicts (see section 3.8).

### 3.8 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.
3.8.1 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, Figure 3-10, Figure 3-11). 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.

![Image of Validate feature]

*Figure 3-10 'Validate' allows to check whether all linkages within a database are working*

![Image of Success message]

*Figure 3-11 Success message when validating the database*

3.8.2 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.

3.8.3 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 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.
4 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 4-2) and via the LCA Collaboration Server dashboard (Figure 4-1).

![Figure 4-1 History of commits in the LCA Collaboration Server dashboard](image1.png)

Figure 4-1 History of commits in the LCA Collaboration Server dashboard

![Figure 4-2 History of commits in openLCA](image2.png)

Figure 4-2 History of commits in openLCA

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.

![Commit history](image)

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 & openLCA)

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.

From openLCA v1.11 onwards, the user will be allowed to search and download datasets from collaboration servers without needing to connect the database to the LCA CS.
Figure 4.4 Grouped search results

Figure 4.5 Ungrouped search results
4.1.4 Download of whole repositories or selected datasets

The download feature allows to download individual versions (commits) of a repository as JSON-LD or ILCD (Figure 4-6) file. It is also possible to download individual datasets from the Collaboration Server.

Figure 4-6 Download and export of data sets

4.2 Advanced features

4.2.1 Managing repositories

The “owner” of the repository will be able to see a set of buttons that can edit the entire repository.

- **Delete** To delete the repository.
- **Export** To export the repository as a zip file. The data format is specific to the LCA CS. This file can be uploaded again to the LCA CS.
- **Clone** To make a copy of the repository in the LCA CS. It clones the whole content including metadata. One can select where you want to clone it to. You can clone over a certain point. This becomes handy when updating the database whilst maintaining the older version.
To change the name of a repository. Caution should be taken here, as this will also change the reference ID or repository path which will cause further problems with the connections to this repository. Every user should update the repository path.

4.2.2 Restoring previous versions of a dataset

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.

4.2.3 Notifications

Notification settings can be adjusted upon clicking on the user’s 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).

![Figure 4-7 Notifications settings](image-url)
4.2.4 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.

![Figure 4-8 The messaging feature can be accessed via the speech bubble in the top right corner](image)

4.2.5 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 able to be commented (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 (Figure 4-11). If an item has been commented, a 'Comments' tab shows up in the bottom right corner of the item’s tab (Figure 4-12).
Upon activating the review mode, comment bubbles can be seen in different locations. Regardless of whether the review mode is activated or not, commented items are indicated through a green speech bubble.

Comment overview in the LCA Collaboration Server dashboard
If an item has been commented, a ‘Comments’ tab shows up in the bottom right corner of the item’s tab.

### 4.2.6 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 (Table 2-2). Proper use of the task system requires at least one user in the team to have Reviewer user rights (Table 2-2).
Table 4-1 Tasks and their status

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>In progress</td>
<td>Awaiting release or moderation (depending on whether a user is Editor or Reviewer)</td>
</tr>
<tr>
<td>Completed</td>
<td>Released and/or reviewed</td>
</tr>
<tr>
<td>Cancelled</td>
<td>No further action is required</td>
</tr>
</tbody>
</table>

4.2.7 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).

![Diff utility screenshot](image)

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 manager can manage libraries and push data to GLAD\(^7\)

• **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.

**Teams**

**Individual users can be combined to a team and teams can be added to groups (of repositories).** Teams consist of users, and groups contain repositories. Teams can be added to a repository. 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)\(^8\).

![Teams](image)

**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

---

\(^7\) Global LCA Data Access network ([http://unep-glad.71.ecedi.fr/](http://unep-glad.71.ecedi.fr/)). Feature developed for the USDA.

\(^8\) Please note that only admins can create teams
5 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 (section 1.1) 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 Life Cycle Thinking. What GreenDelta came up with is the vision of a public repository of freely available LCA models under the Creative Commons license.9

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.

If this sounds interesting, please reach out: https://www.openlca.org/contact/

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 browser10 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.

9 https://creativecommons.org/licenses/
10 e.g. http://nexus.openlca.org/
6 Support and Contact

6.1 Prioritised support via the GreenDelta HelpDesk

For more information about GreenDelta’s support services for openLCA and the LCA Collaboration Server, please refer to: https://www.openlca.org/helpdesk.

Support services include:

- Hosting of an LCA Collaboration Server (http://www.openlca.org/lca-collaboration-server-hosting-andservices/)
- Support in setting up an LCA Collaboration Server (contact us)
- Professional training on how to get the best out of the LCA Collaboration Server for your organisation (contact us)

6.2 Public support via ask.openLCA.org

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 LCA Collaboration Server. We encourage users to use ask.openlca.org for non-confidential support requests. However, with ask.openlca.org 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 us via email for confidential support requests or resort to one of GreenDelta’s service contracts for prioritised support via the GreenDelta HelpDesk.

6.3 Demonstration of the LCA Collaboration Server

Demonstration videos of the LCA Collaboration Server are available on the openLCA YouTube channel: https://www.youtube.com/c/openLCA. New content is coming soon, with the release of this latest version.

6.4 Contact GreenDelta GmbH

For further inquiries please contact GreenDelta GmbH.

Tel. +49 30 48 496 – 030 Fax +49 30 48 496 – 991
gd@greendelta.com GreenDelta GmbH
Kaiserdam 13
14057 Berlin
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 sent 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.

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

• 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.
A.2 Installation of the LCA Collaboration Server

We have prepared a detailed installation guide on openLCA.org - https://www.openlca.org/lca-collaboration-server-1.3.0-installation-guide

A.3 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 <basehref="/lca-collaboration"/> with your context path, e.g. <basehref="/anotherpath"/> 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):
Changing settings in the web application

Figure A.1: Admin area basic settings and enabled features
A.4.1 Basic settings

Server name: 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 elasticsearch 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, Notifications, Activities and Tags (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 / macOS / Linux
  - 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)

A.5.2 Requirements for the LCA Collaboration Server

Minimum:

- CPU: DualCore CPU or better
- RAM (Elasticsearch/LCA CS): 2GB/2GB
- Disk: HDD (the required 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/8GB
  - Disk: SSD

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.

---

11 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.