soda4LCA release 2.0.0 Installation and Configuration Guide

Table of Contents

List of Tables

Chapter 1. General information

1.1. Intended audiences

This guide gives detailed step-by-step instructions on how to set up a soda4LCA database node. It is intended for network and website managers.

1.2. System prerequisites

This software is conceived as a web application and runs inside a servlet container such as Apache Tomcat. The following components must already be in place in order to install soda4LCA:

- Java 1.6 or newer
- MySQL 5.0 database or newer
- J2EE servlet container (recommended: Tomcat 6.0)

The instructions and configuration examples are for Tomcat 6.0. If you use another servlet container or Tomcat version, the necessary instructions may differ, in that case consult the container's documentation.

The guide refers to the base directory of the Tomcat installation as `$CATALINA_HOME`.

Chapter 2. Upgrading

2.1. Upgrading from a previous release to soda4LCA 2.x

For an upgrade of a pre-2.x release to a 2.x release, the database schema must be completely emtpy. Make sure you export your data before wiping your database schema. Any user accounts need to be recreated once the upgrade is complete.

To upgrade your soda4LCA installation from a previous (1.x) release to 2.x, follow these steps:

1. Export your data (see section **"Export"** in the Administration Guide for detailed instructions) and save the resulting ZIP file.
2. Stop the servlet container (Tomcat).
3. Your application's database schema needs to be completely empty before you start up the 2.x release. The easiest way is to drop and re-create your database schema. For instance, if the name of your database schema is `soda4LCAdb`, executing the following statements will ensure your database schema is empty:

    ```
    DROP SCHEMA soda4LCAdb;

    CREATE SCHEMA soda4LCAdb DEFAULT CHARACTER SET utf8;
    ```

4. Locate the application's folder inside the `$CATALINA_HOME/webapps` folder. The name of this folder matches your application's context path. For instance, if your application' context path is `/Node`, there will be a folder `Node`. Delete that (and only that) folder.
5. Change the name of the soda4LCA 2.x WAR archive to match your context path. For example, if your context path is `/Node`, name it `Node.war`. Then copy it to `$CATALINA_HOME/webapps`, overwriting the existing WAR file.
6. Now start the servlet container (Tomcat).
7. Once the application is up and running, log in as `admin` and change the default password.
8. Now you can import your data using the ZIP file created in step 1. Refer to the section **"Import"** in the Administration Guide for detailed instructions.

Chapter 3. Preparing the environment

3.1. Database system variables

For being able to import large datasets into the database application, the `max_allowed_packet` system variable needs to be adjusted in your MySQL instance.

- `max_allowed_packet=25M`

Depending on your operating system and MySQL installation type, there are different ways to set these variables. You can either add or replace the above line to your `my.cnf` or alternatively pass them as command line parameters in MySQL's startup script like this `--max_allowed_packet=25M`.

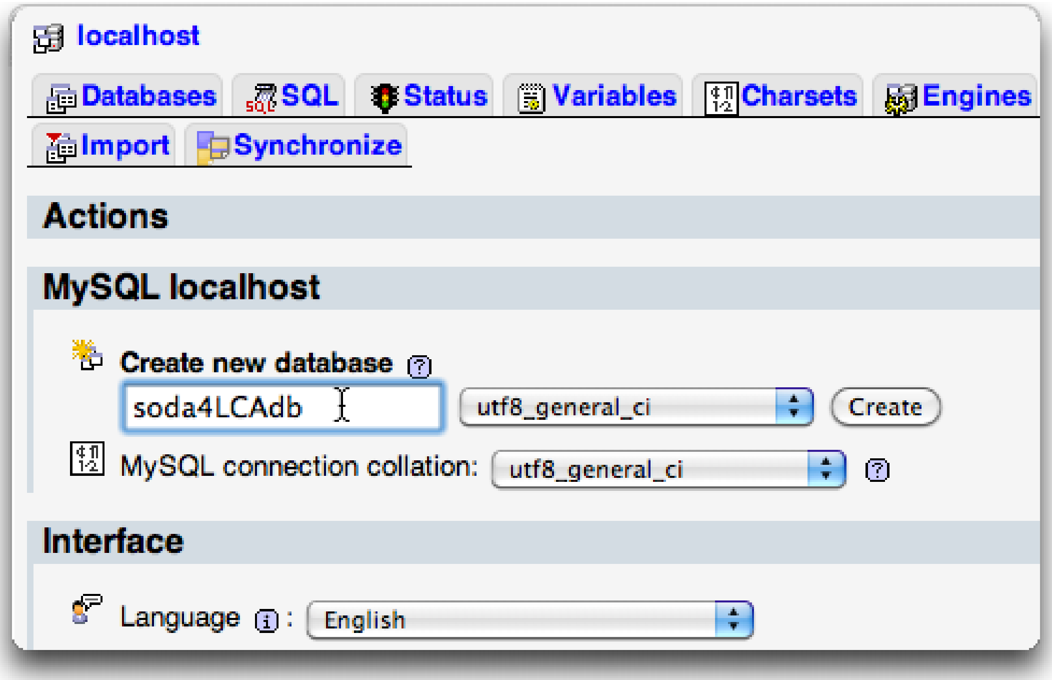
3.2. Recommended: Increasing heap size for Tomcat's Java Virtual Machine

To ensure optimum performance of the soda4LCA application, the heap size of the Java Virtual Machine should be increased to at least 400MB. This can be achieved by setting the `CATALINA_OPTS` environment variable to `-Xmx400M`.

Chapter 4. Installing the application

4.1. Database setup

1. Create a new, empty database schema using an UTF-8 character set and the default collation. For example, when using phpMyAdmin, use the "Create new database" section on the main page:

   Create new database: soda4LCAdb utf8_general_ci

   MySQL connection collation: utf8_general_ci

   Language: English

   The database schema name used in all the examples is `soda4LCAdb`. If you are using a different schema name, be sure to change it accordingly in any sample code you may be copying from this manual.

2. Obtain and install MySQL database driver

   a. Download the MySQL database driver here: http://dev.mysql.com/downloads/connector/j/.

   b. Unpack it and place the `mysql-connector-java-5.x.xx-bin.jar` file into the `$CATALINA_HOME/lib` folder.

4.2. Basic application setup

These instructions are for setting up a single instance of the application. If you require multiple instances running inside the same container, follow the instructions in the section "Advanced application setup".

1. Add the following declaration to the `<GlobalNamingResources>` section of your `$CATALINA_HOME/conf/server.xml` and adjust username and password and, if necessary, the database URL accordingly:

```
<Resource auth="Container"
     driverClassName="com.mysql.jdbc.Driver"
     maxActive="8"
     maxIdle="4"
     validationQuery="SELECT 1"
     testOnBorrow="true"
     removeAbandoned="true"
     removeAbandonedTimeout="300"
     logAbandoned="true"
     name="jdbc/soda4LCAdbconnection"
     type="javax.sql.DataSource"
     url="jdbc:mysql://localhost/soda4LCAdb?
useUnicode=yes&characterEncoding=UTF-8"
     username="myusername"
     password="mypassword"
/>
```

2. To configure the application, a properties file providing information about node name etc. is required. Using `soda4LCA.properties.template` (in the Installation_Guide folder of the documentation) as a template, create the application's configuration file and place it into `$CATALINA_HOME/conf/soda4LCA.properties`. For detailed step-by-step instructions, refer to the section Configuring and customizing a node.

Important

Ensure to configure the `files.location.datafiles` properties in the "data and temp directories" section of your `soda4LCA.properties` to point somewhere outside the directory containing web application, otherwise you will lose your data (external files attached to source datasets) on re- or undeploy! See the section Data and temporary directories section in the "Configuring and customizing a node" chapter for details.

4.3. Advanced application setup

If you require multiple instance of the application running in the same container, individual database connections can be declared per-webapp by editing the webapp's context configuration file in Tomcat's `$CATALINA_HOME/conf/Catalina/localhost` folder. You can specify the path to the configuration file for each instance using the soda4LCAProperties environment via JNDI.

```
<Context antiJARLocking="true" swallowOutput="true">
    <Resource auth="Container"
        driverClassName="com.mysql.jdbc.Driver"
        maxActive="8" maxIdle="4"
        validationQuery="SELECT 1"
        testOnBorrow="true"
        removeAbandoned="true"
        removeAbandonedTimeout="300"
        logAbandoned="true"
        name="jdbc/soda4LCAdbconnection"
        type="javax.sql.DataSource"
        url="jdbc:mysql://localhost/soda4LCAdb?
useUnicode=yes&characterEncoding=UTF-8"
        username="myusername"
        password="mypassword"/>
    <Environment name="soda4LCAProperties" value="/path/to/
soda4LCA.properties" type="java.lang.String"/>
</Context>
```

Make context configuration immutable

It is a good idea to make the context configuration file immutable (using `chattr +i`) so it won't be overwritten by Tomcat when the context is redeployed.

4.4. WAR file installation

Now copy the `Node.war` file to `$CATALINA_HOME/webapps`. You may rename the WAR file to whatever you wish, the name of the WAR file will match your application's context path once deployed.

4.5. Startup the application

Once you have completed customizing the configuration file, start the servlet container by executing `$CATALINA_HOME/bin/startup.bat` (on Windows) or `$CATALINA_HOME/bin/startup.sh` (on Mac OS/unix) and closely observe the console output. If there are any errors, consult the following section "Common problems" for information on how to resolve them.

Once the application has started successfully, you can access the application by pointing your browser to the URL

```
http://localhost:8080/Node/
```

Please login in and immediately change the default administrator password as described below.

Immediately change default administrator password

The application ships with a default administrator password that has to be changed immediately upon successful installation in order to prevent unauthorized access.

Once the application is up and running, log in with the administrative account (username: admin, password: default) and change the password (select "My Profile" in the footer bar and enter a new password in the respective fields of the following screen).

4.6. Common problems

Error: application configuration properties not found

Make sure you have a soda4LCA.properties in place either at `$CATALINA_HOME/conf/soda4LCA.properties` or at some other location with a proper Environment entry in your Context configuration that points to the location of the file.

FATAL ERROR: could not determine base URL for service interface

If no hostname (and, optionally, port) have been declared in soda4LCA.properties, the application will try to detect the hostname, port and context path. If it cannot build a valid URL from the obtained information, this error will be raised.

Error: Cannot set nodeid/nodename from configuration file

This means the soda4LCA.properties configuration file does not contain said information. Refer to the soda4LCA.properties.template for an example configuration file.

FATAL ERROR: database schema is not properly initialized

This error occurs when the application finds a database schema that is corrupt or not compatible with the current application version. At first setup, make sure the database schema is empty. Also, be sure you have set the `lower_case_table_names` property correctly for your MySQL instance as described in section Database system variables.

FATAL ERROR: could not lookup datasource

This is likely caused by an incorrect servlet container configuration. Make sure you have added a correct Resource declaration in the GlobalNamingResources section of the server.xml as described above.

Chapter 5. Configuring and customizing a node

Follow these instructions to configure and customize your node. Optional steps are marked as such and can be omitted.

Make a copy of the included `doc/soda4LCA.properties.template` and place it in `$CATALINA_HOME/conf/soda4LCA.properties` (this is the default location where the application will be looking for it). Then edit this file, following these steps to complete configuring and customizing the application.

NOTE: The only information that always needs to be changed in this configuration file are

- all entries unter `service.node.*`
- all entries unter `service.admin.*`
- `files.location.datafiles`

All other options can be left untouched if no further customization is desired.

5.1. Hostname and port

If your host is not behind a proxy, the hostname will be autodetected and does not need to be configured, so it can simply be commented out. Example:

```
#service.url.hostname = localhost
```

Otherwise, use the hostname that is exposed to the public network. Example:

```
service.url.hostname = lcadata.acme.org
```

The port number needs to be set to the HTTP port of your application server. For a default Tomcat installation, this is port 8080. Example:

```
service.url.port = 8080
```

5.2. Node information

Choose a suitable node ID (this must not contain any spaces) and provide a node name, description and operator. The node ID is the name under which the node will be visible on the network. Example:

```
service.node.id = ACME_LCI
service.node.name = ACME public LCI data
service.node.description = provides public LCI data about products and services by ACME International, Inc.
service.node.operator = ACME International, Inc.
```

5.3. Administrative contact

Provide administrative contact information. Example:

```
service.admin.name = John Doe
service.admin.phone = +49 721 555-4242
service.admin.email = lci-node-admin@acme.com
service.admin.www = http://www.acme.com/
```

5.4. Template

Declare the template that will determine the appearance of the application. To use the default template that ships with the application, use "default". Example:

```
template = default
```

If you have made your own template (for instance, by deriving it from the default template), specify the folder name. Example:

```
template = mytemplate
```

The folder `mytemplate` has to reside inside the folder `web/templates` .

5.5. Theme

Specify the theme that will determine the look and feel of the application. Example:

```
theme = cupertino
```

Refer to the Primefaces documentation [http://primefaces.org/themes.html] for a list of possible themes that ship with the application by default.

5.6. Welcome page (optional)

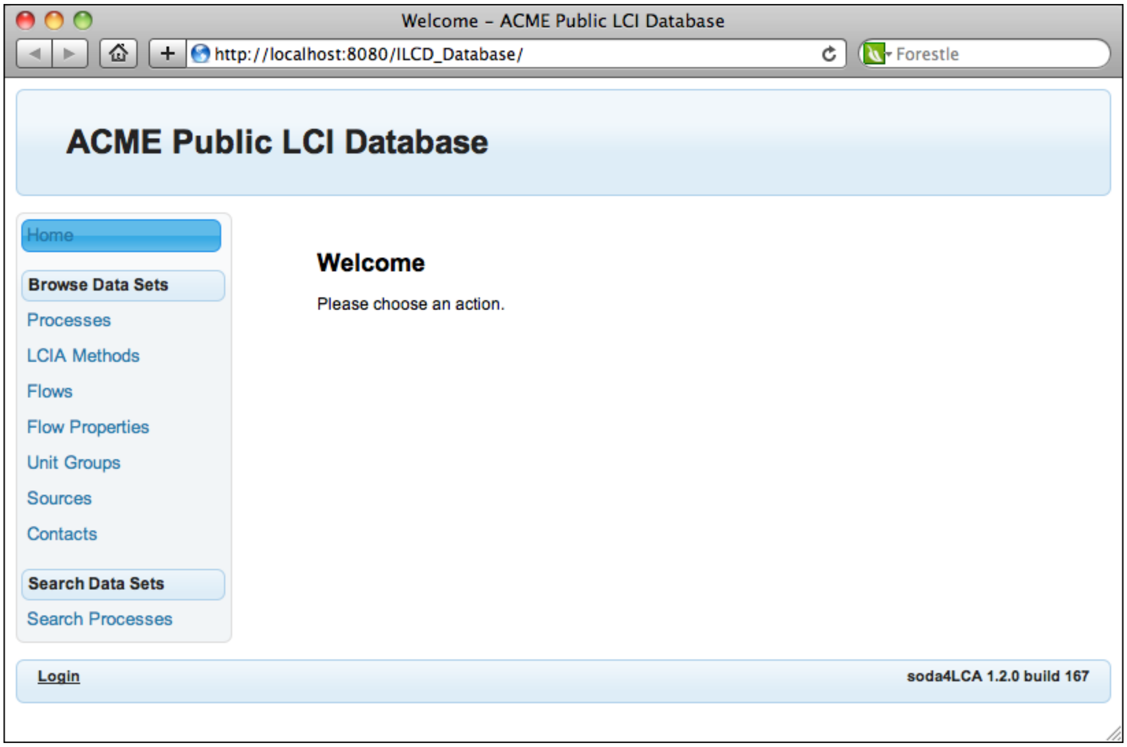
You may specify a custom welcome page that will be shown when as the application's index page. Example:

```
welcomePage = path/to/jumppage.xhtml
```

5.7. Application title (optional)

Use the title property to modify the title of the application that is displayed in the header section as well as in the browser's window title. Example:

```
title = ACME Public LCI Database
```



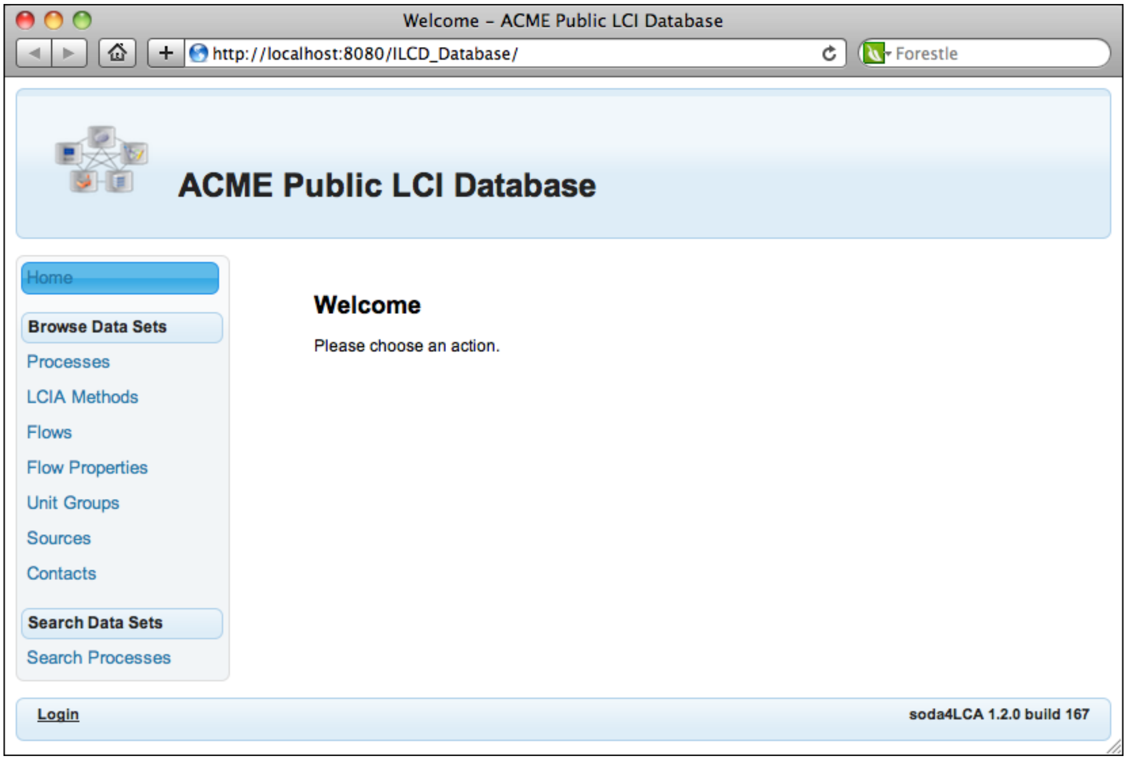
Welcome

Please choose an action.

5.8. Logo (optional)

Use the logo property to make a custom logo appear next to the application title in the header section. If you put the image file with the logo relative to the application's context path, use the expression %contextPath% as replacement. Example:

```
logo = %contextPath%/logo.png
```



Welcome

Please choose an action.

Use "false" to disable display of a logo. Example:

```
logo = false
```

5.9. Preferred Languages (optional)

As datasets may contain information in multiple languages, the handling of those when displaying datasets can be controlled with this setting.

By default, the preferred languages used by the application are en, de and fr (in this order, en being the default language). If information within a dataset is not available in the default language, the next language on the list will be used, and so forth.

You may configure your own specific languages and their precedence, the first language in the list being the default one, the second the first alternative, etc. The language codes have to be separated by commas. Example:

```
preferredlanguages = de, en, fr, es, ru
```

5.10. Data and temporary directories

External files attached to source datasets are currently stored on the filesystem. The default location is `WEB-INF/var/files` but can (and is strongly recommended to) be changed with the `files.location.datafiles` option. Example:

```
files.location.datafiles = /usr/local/soda4LCA/datafiles
```

Important

The option `files.location.datafiles` must point somewhere outside the directory containing the web application, otherwise the data may be lost on re- or undeploy!

For temporary storage of uploads and ZIP files, separate configuration options are available, these default to `WEB-INF/var/uploads` and `WEB-INF/var/zips`, respectively, and do not necessarily need to be set. Example:

```
files.location.uploads = /tmp/uploads
files.location.zipfiles = /tmp/zips
```

Now you have configured all necessary options in order to run the application. Start the servlet container as described in section Startup the application.

For a complete list of all available configuration options, refer to the following section.

5.11. Overview of configuration options in soda4LCA.properties file

Table 5.1. Configuration options in soda4LCA.properties file

Option	Required	Possible Values	Explanation
service.url.hostname	optional (default: auto-detect)	valid hostname or IP address	The hostname or IP address for this node as it is supposed to appear in the service URL. Will be usually be auto-detected by the application, so this only needs to be specified if the node is behind a proxy.
service.url.port	optional (default: 80)	valid port number	Port number of the application.
service.node.baseURL	optional	full URL	If you are behind a proxy and auto-detect does not work for your specific configuration, but you need to specify an exact custom URL, use this optional property.
service.node.id	yes	string, but MUST NOT contain any spaces!	This is the name to identify the node in the network.
service.node.name	yes	string	Name for this node
service.node.description	yes	string	Description for this node
service.node.operator	yes	string	Name of the organization operation this node
service.admin.name	yes	String	name of the administrative contact for this node
service.admin.phone	yes	String	phone number of the administrative contact for this node
service.admin.email	yes	String	email address of the administrative contact for this node
service.admin.www	yes	String	web site of the administrative contact for this node
template	yes	template folder name	This is the XHTML template that will be used for displaying the site. You can build and use your custom template.

Option	Required	Possible Values	Explanation
theme	yes	String	Specifies the jQueryUI theme, consult the PrimeFaces manual for details on how to change themes.
welcomePage	optional	path to jump page, relative to web directory	This can be used to make the application show a custom jump page instead of the default browse view.
title	yes	String	The title of the database instance that will be shown on top of the page.
logo	yes	full qualified path to logo, or "false" to disable	This can be used to display a custom logo in the page header. Use "false" to disable the logo. For specifying a path relative to the application's context, path, you may use the expression %contextPath% as replacement, for instance "%contextPath%/my/path/to/image.png" .
security.guest .metadataOnly	yes	true, false	By default, only metadata is publicly available. If this is set to false, then full data sets will be publicly available.
user.registration .activated	optional	true, false	Allow users to register themselves.
user.registration .selfActivation	optional	true, false	Allow new users to activate their accounts upon registration.
user.registration .activationEmail	optional	true, false	Check new users' email addresses by sending an activation email.
user.registration .spam.protection	optional	true, false	
user.registration .registrationAddress	optional	String	An email address where notifications about newly registered users are sent.
user.registration .admin.email	optional	String	Email address to use for the From header of registration emails.
mail.sender	yes	String	Email address to use for the From header for

Option	Required	Possible Values	Explanation
			emails sent by the application.
mail.hostname	yes	String	Hostname for the SMTP server.
mail.port	optional, defaults to 25	String	Port number for the SMTP server.
mail.sitename	yes	String	Application title to use for emails sent by the application.
mail.auth	optional, defaults to false	Boolean	Whether to use authentication information for the SMTP server.
mail.user	yes if mail.auth=true	String	SMTP server username.
mail.password	yes if mail.auth=true	String	SMTP server password.
files.location.datafiles	optional, defaults to WEB-INF/var/files	full qualified path name	Directory where external files attached to source datasets will be stored. IMPORTANT: This should point to somewhere outside the directory containing web application, otherwise you will lose your data on re- or undeploy!
files.location.uploads	optional, defaults to WEB-INF/var/uploads	full qualified path name	Directory where uploaded files are temporarily stored.
files.location.zipfiles	optional, defaults to WEB-INF/var/zips	full qualified path name	Directory where zip files are temporarily stored.
feature.browse.processes	yes	true, false	Enable/disable browsing of process datasets
feature.browse .lciamethods	yes	true, false	Enable/disable browsing of LCIA method datasets
feature.browse.flows	yes	true, false	Enable/disable browsing of flow datasets
feature.browse .flowproperties	yes	true, false	Enable/disable browsing of flow property datasets
feature.browse .unitgroups	yes	true, false	Enable/disable browsing of unit group datasets
feature.browse.sources	yes	true, false	Enable/disable browsing of source datasets
feature.browse.contacts	yes	true, false	Enable/disable browsing of contact datasets
feature.search.processes	yes	true, false	Enable/disable search of process datasets