SOLA

Solution for open land administration

SLTR NIGERIA UNIQUE REPOSITORY

Contents

[Revision history 1](#_Toc386536005)

[How to Deliver Updates 2](#_Toc386536006)

[SOla UPDATES 2](#_Toc386536007)

[DROPBOX folder 3](#_Toc386536008)

[SOla DATABASE 3](#_Toc386536009)

[SOla Server 3](#_Toc386536010)

[Deploy SOLA Services 4](#_Toc386536011)

[Deploy SOLA WEB START 4](#_Toc386536012)

[TEST UPDATES 4](#_Toc386536013)

[LOg issues on to lighthouse 4](#_Toc386536014)

# Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | author | date | REASON for revision |
| PA1 | Maria Paola Rizzo | 22/07/2014 | First draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# How to Deliver Updates

In its current implementation, the SOLA is intended to work in a walled garden (e.g. a department network) with the only need for communication between the desktop clients and the SOLA server. The following diagram describes this situation.



Figure 1 SOLA setup

## SOla UPDATES

Sola updates can occur at database level and/or at code level.

* SQL scripts will be used for database changes
* sola\_services\_ear.ear (library containing the services updates) and desktop\_web\_start.war (library containing the client desktop updates) files will be used for code changes

## DROPBOX folder

Dropbox folder is used to store and exchange updates.

The structure of this folder is shown in the following picture:



* Dropbox:
  + **database** folder
    - **yymm<version>\_<LH#>.sql scripts**
    - **Delivered** folder
      * previously delivered sql scripts
    - **base** folder
      * the base sql scripts from which the updates started
  + **code** folder
    - **Delivered** folder
      * previously delivered .ear and .war files
    - desktop-web-start.war file
    - sola-services-ear.ear file
  + **resources** folder
    - any useful resource

Copy database/\*.sql scripts, desktop-web-start.war and sola-services-ear.ear files onto a flash and optionally onto Sola server.

## SOla SLTR NIGERIA

SOLA SLTR NIGERIA uses the standard SOLA 10 Git repositories for managing its code base.

These repositories are:

1) code - Contains the Main POM file and

additional supporting files. This Git repository acts

as the super/parent repositories for all the other SOLA

Git repositories.

2) clients - Contains the Clients Desktop and Clients Admin projects

along with the other client side projects. Attached as

the clients subdirectory of the code repository.

3) common - Contains only the Common Utilities project. Attached as

the common/common subdirectory of the code repository.

4) rules - Contains only the Common Rules project. Attached as the

common/rules subdirectory of the code repository.

5) help - Contains only the Common Help project. Attached as the

common/help subdirectory of the code repository.

6) messaging - Contains only the Common Messaging project. Attached as

the common/messaging subdirectory of the code

repository.

7) boundary - Contains the Web Service Boundary projects. Attached as

the services/boundary subdirectory of the code

repository.

8) services - Conatins the EJB and common service projects. Attached

as the services subdirectory of the code repository.

9) database - Contains the SOLA database scripts. Attached relative to

the code repository in the ../database directory.

10) test - Contains the Fitness and Performance test projects.

Attached in the test subdirectory of the code

repository.

You can read the README.TXT files in the code and database repositories

to get information about how repositories are used and handled.

In addition to the standard 10 repositories there are few other repositories: one for each state

Those contain the database scripts specific of each state.

Thus, at the moment additional repositories are:

11) database-crossriver

12) database-jigawa

13) database-kaduna

14) database-kano

15) database-kogi

16) database-ondo

All changes to the database are captured as a changeset script in the changeset folders (refer to the README.TXT file in the database folder):

\*) database/changeset for the changes that apply to all the state

\*) database-<state>/changeset for the changes that apply only to that specific state

When running the database/create\_soladb\_nigeria.bat you will now be asked also to specify:

- State (this will lead the installation to get the scripts from the specific state repository)

- LGA office (this will properly set the system-id)

A configuration file has been introduced to set the state name, in order to be retrieved when running the Client Applications (both desktop and admin)

It is used at the startup to launch the Splash screen of the state.

This file is: configuration.properties located under SOLA-SLTR-NIGERIA\sola\code folder.

It has to be copied under the <USER-HOME>/sola directory.

The value of the key "state" must be then updated so to properly set the name of the state.

e.g.

state=<name of the state>

to be changed to:

state=Kano

Another customized repository is the client repository.

Under report project=> resources there are few other folders in addition to the standard resources (images,org and reports), one for each state which contain the reports specific to that state:

Thus, at the moment folders under Client Report=>resources are:

- CrossRiver

- Jigawa

- Kaduna

- Kano

- Kogi

- Ondo

- images

- org

- report

Under each of the <state> folder there are the reports customized by each state:

**e.g.**

- PD listings

- CofO

- SLTR PLAN

## DATABASE UPDATES

1. Updates which affect the structure of the database are handled saving scripts under database/changeset
2. Updates which affect only data of a specific state are handled saving script under database-<state>/changeset

## CODE UPDATES

1. Copy on a flash or on a location in the Glassfish Server the desktop-web-start.war file from the **code** folder (**outside delivered**)
2. Browse to the Glassfish Admin Console (localhost:4848).

If Glassfish isn’t currently running, start it from the double clicking on the sola server start icon on the desktop, before attempting to browse to the Admin Console.

1. Select the Applications node and choose Deploy...
2. Set the location to the **desktop-web-start.war** in the folder where you copied it.
3. Click OK to deploy the WAR
4. From your web browser, browse to **localhost:8080/sola**. You will be taken to the SOLA Web Start landing page.
5. Follow the Installation instructions on that page to download and install one or both of the SOLA Client applications.

## ADDING NEW STATE

Log whatever error/issue or new request on to lighthouse.

To create a ticket, use the Create new ticket button on the top right of the screen

Clicking the Create new ticket button will bring you to the New Ticket form.

Here you can enter a Title for the ticket. Keep the title as short and succinct as possible.

Enter a description for the issue. Provide as much detail such as what screen you were working on, what button you pressed and what did or didn’t happen that caused you to raise the ticket.

The Responsible drop down can be left blank unless you want to assign the ticket to someone explicitly.

When creating a new ticket, the Ticket state should be left as Open. The other two tickets states are Fixed and Closed. The Developer will assign the ticket the Fixed state once they have confirmed their fix. The priory of the ticket defaults to Low.

You can set the priority of the ticket to a different value depending on how significant you judge the issue to be.

To help categorize the ticket, you can add one or more tags and you can also upload a file to support the issue. Sometimes a screenshot can be very useful to explain the error.