SOLA Desktop Applications

Administration Guide

for Release 1503a

Revision History

Date	Version	Description	Author
Nov 25, 2019	1.0	Initial draft	Neil Pullar
Jan 27, 2020	1.1	Updates including change to use OpenJDK, OpenWebStart & restructuring of report requested by FAO	Neil Pullar

Table of Contents

1. About this Guide

1.1 Licensing

The source code for the SOLA Desktop applications and Web Admin is licensed under the Berkeley Software Distribution (BSD) 3-clause license (a.k.a. New BSD or Modified BSD). This guide describing system administration functions and processes associated with Community Server and Web Admin is considered part of these software applications and hence covered by the same license. The details of this license are:

2. Server Installation on Windows

This description assumes the SOLA Desktop server components are being installed on a 64bit Windows Desktop computer running Windows 10 (on either 32bit and 64bit Windows operating systems) If the operating system architecture is 32bit, use 32bit installers in place of the 64bit installers noted in the installation steps.

2.1 Download Installers

The SOLA Desktop deployment package containing the SOLA services EAR file, desktop web-start files and database build files can be downloaded from:

- http://github.com/SOLA-FAO/latest-release [SOLA Registry]
- http://github.com/SOLA-SL-FAO/latest-release [SOLA State Land]
- http://github.com/SOLA-SR-FAO/latest-release [SOLA Systematic]

Alternatively if you decide to do a Docker based installation you only need to download these github repositories:

- http://github.com/SOLA-FAO/docker [SOLA Registry]
- http://github.com/SOLA-SL-FAO/docker [SOLA State Land]
- http://github.com/SOLA-SR-FAO/docker [SOLA Systematic]

Once you have downloaded the docker github repository go directly to Section 3.5 to complete a Docker installation.

You will also require a number of installation packages for the components supporting the SOLA Desktop applications (if you use the first installation option, not a Docker installation):

- Latest update of Java 8 OpenJDK (64bit) (https://adoptopenjdk.net)
- Latest update of Java 8 JRE (32bit) (http://java.com/en/download/manual.jsp). Use the Windows Offline (i586 not 64bit) package. Please note that you will prompted to login using Oracle login credentials and the pending changes to Oracle's charging policy
- Payara Server 5 - Build 193.1 is recommended. The most recent stable version of Payara Server is available at https://www.payara.fish/software/downloads/
- The most recent update for PostgreSQL 11 Download 64bit version (Win x86-64) (http://www.enterprisedb.com/products-services-training/pgdownload)
- The latest update of Postgis 2.5 for PostgreSQL 11. Download the pg11x64 version **executable** (http://download.osgeo.org/postgis/windows/pg11/)
- GeoServer (http://geoserver.org/download/). Version 2.16.0 (WAR file) is used for this guide.
- Notepad++ (https://notepad-plus-plus.org/downloads/) Text editor to help with updates to configuration files.
- If you are installing on a recently built Windows Operating System, you will also need to install the Visual C++ Redistributables. These are required for the Postgres uuid-ossp extension. The Visual C++ Redistributables you require is the vc_redist.x86.exe from https://www.microsoft.com/en-us/download/details.aspx?id=48145.

2.1.1 Visual C++ Redistributable & Notepad++

1) Install the **Visual C++ Redistributable** package on the server by right clicking the vc_redist.x86.exe installer and choosing Run as Administrator.
2) Optionally install **Notepad++** if you don't have a suitable plain text editor already installed.

2.2 OpenJDK 8

1) If there is an existing installation of Oracle Java, uninstall it using the regular Windows uninstall process (Click Start, select Settings-System-Apps & features, select Java program and click on Uninstall button)

2) Download the latest msi installation file for OpenJDK 8 64bit for Windows from https://adoptopenjdk.net
3) Run the AdoptOpenJDK msi file (select, right click, I**nstall**) and include "Set JAVA_HOME variable" in addition to the default "JDK with Hotspot" to be installed when prompted.
4) Check the JAVA_HOME environment setting. In File Manager, right click on **This PC – Properties – Advanced system settings – Environment Variables** and check that
    a. there is no JAVA_HOME in your user variables (top panel)
    b. the JAVA_HOME variable as a system variable (bottom panel) is **C:\Program Files\AdoptOpenJDK\jdk-8.0.232.09-hotspot** - will change as new versions of OpenJDK are released.

2.2.1 Oracle Java

Alternatively to installing OpenJDK Java you can install the Oracle version of Java, noting that as of 2020, Oracle have indicated that there will be a charge to use their version of Java and the *javaws* routine for installing using Java WebStart has already been deprecated by Oracle.

1) Run the Java 8 JDK 64bit installer by right clicking the installer and choosing Run as Administrator. You should exclude the installation of the JDK Source Code. Use the default install location.
2) When downloading the Java 8 JRE 32 bit installer, click on the option to remove older versions.
3) Run the downloaded Java 8 JRE 32bit installer by right clicking the installer and choosing Run as Administrator.
4) Confirm the correct version of JRE as the default JVM on your computer:
    a. From Control Panel, click on Java and go to the Java Control Panel, click About... You should see the Java 8 Standard Edition dialog displayed.

Figure 1 - About Java

    b. If a different version of Java is displayed, remove the older other versions (using standard Windows Program Remove utility) and install / reinstall Java JRE (32 bit version).

Note – The 32bit JRE attempts to install the Java browser plugin and this can be blocked by Internet Explorer Enhanced Security Configuration (ESC), hanging the installer. To disable IE ESC, open IE and enter the following URL *res://iesetup.dll/IESechelp.htm#turnoff*. This will provide instructions on how to disable ESC for your version of Windows. Once disabled, continue with the JRE installation, then re-enable ESC after the JRE is installed.

2.3 PostgreSQL

To run the install for PostgreSQL you will need administrator privileges on your computer.

1) Run the PostgreSQL 12 installer by right clicking the installer and choosing Run as Administrator.
2) Leave the default location for the installation directory (C:\Program Files…), but you can change the data directory to use a folder outside of C:\Program Files … such as C:\PostgreSQLdata\12
3) At the Password form, enter a suitable password for the postgres superuser. You will need to use this password frequently, so ensure it is a password you will remember.
4) Use the default port of 5432 and the default locale
5) Proceed with the installation – it may take several minutes. Once complete, close the installer – **do not run StackBuilder at this time**.
6) Verify PostgreSQL has been installed correctly by starting pgAdmin (Start menu > All Programs > PostgreSQL12 > pgAdmin4) and attempting to connect to the new database [In File Manager locate the pgAdmin4 executable and right click and **Pin to Start**]
    a. If a PostgreSQL 12 (localhost:5432) server is not listed under the Servers node, in the Object Browser view, click the plug tool in the top left of the tool bar and enter the details for your PostgreSQL server.
    b. Right click this database and choose Connect
    c. When prompted, enter your postgres password.
7) You will be informed when there is a new version of pgAdmin4. To install these new updates go to https://www.postgresql.org/ftp/pgadmin/pgadmin4, select the latest version and then the windows version to download the installation file (.exe file type). Right click this file once downloaded, select **Run as administrator** and this will overwrite the old version and install the new version of pgAdmin4.

If you encounter issues while installing PostgreSQL, or you get an unexpected error while trying to connect to the PostgreSQL database, you may need to uninstall and reinstall Postgres. If you do need to reinstall, follow these steps

1) Uninstall PostgreSQL using the Control Panel
2) Locate the PostgreSQL installation directory (e.g. in Program Files) and delete it completely
3) When you re-run the installation, choose the install location to be C:\ rather than C:\Program Files.

2.3.1 PostGIS

1) Run the PostGIS 3 installer by right clicking the installer and choosing Run as Administrator.
2) When installing the PostGIS component, **clear/uncheck** the Create spatial database option on the Choose Components form. If you do not uncheck this option, you can simply delete the additional spatial database at a later time.
3) If prompted to update pgAdmin during the file copy process, you can optionally do so.

2.3.2 Configure PostgreSQL to allow access from other computers

1. By default PostgreSQL is locked down to prevent access from computers other than the localhost. If your database server will be used as the production server, then it may be a good option to leave PostgreSQL in this locked down state. If the database will be used for testing or training, it may be desirable to access the database server from other computers. To allow these connections it is necessary to make some configuration changes.
   Edit the **pg_hba.conf** file (by default in the ..\main subfolder of the PostgreSQL Data folder) and add the following line to this file (depends on the IP address range of your network)
   ```
   host all all 192.168.1.1/24   md5
   ```
2. Configure PostgreSQL to accept connections from SOLA applications.

This can be achieved from psql with the ALTER SYSTEM Command (as Administrator) type:

```
sudo su - postgres
psql
# ALTER SYSTEM SET listen_addresses='*';
\q
$ logout
```

2.3.3 Add to Path Environment

1) In File Manager, right click on **This PC** and navigate Properties => Advanced system settings=> Environment Variables => System Variables> and select **PATH,**
2) Click **Edit,** add then the **New** button and add `C:\Program Files\PostgreSQL\12\bin` and **OK**
3) Click on **New** and `C:\Program Files\PostgreSQL\12\lib` and **OK** (3 times) (to exit Advanced system settings
4) Restart computer

2.3.4 Create SOLA Database

Where there is no configured PostgreSQL database backup and it is the very first time you create a new SOLA database in PostgreSQL, follow these steps:

- Locate the database folder in the Community Server deployment package, copy to the computer hosting PostgreSQL (if necessary) and **Run as Administrator** the **create_database.bat** build script in the **database** folder.
- A Command Prompt will display and ask a sequence of questions about the database connection. Use the default values shown in square brackets, except for the DB Password and Create or Replace the database question (answer Y).
- You should fill the database with sample data initially to help test all components are correctly configured. You can rerun the create_database script at any stage to rebuild a clean database without the sample data.

Figure 2 – Enter create_database script

The script should take about a minute to run. If it only takes a few seconds, it will indicate an issue with script.
Open the build.log file (it will be in the same directory as the create_database script) to see the error/warning messages.

Known error/warning messages

- *The system cannot find the path specified* means the psql.exe could not be located. **Edit** create_database.bat in your preferred text editor (e.g. Notepad++),

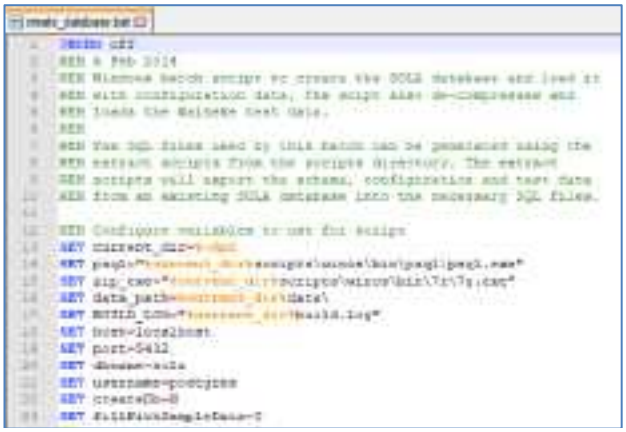


Figure 3 – To edit line 14 of create_database.bat

Go to line 14 and replace that line that includes the folder path for the psql executable in your Postgresql installation.
eg `SET psql="C:\Program Files\Postgresql\12\bin\psql.exe"`
Save the changes and re-run create_database.bat.

2.4 Payara Server 5

Payara Server (Build 5.194.1) was the current version of Payara in January 2020 has been used as the application server for SOLA Community Server & SOLA Web Admin.

Payara Server can be installed in two ways. The easiest way to install an appropriately configured Payara Server instance is to copy an already configured instance (remembering to stop Payara Server before making the copy).

Follow these steps if you want to make a fresh Payara Server installation:

1) Download the Payara Server zip bundle from the web site
   https://www.payara.fish/downloads
   This download is approximately 150Mb
2) Unzip (and move or copy) into:
   `C:\payara5`
3) The .NET Framework 3.5 is required to create the Payara Server Windows Service. Ensure the .NET Framework 3.5 Features are installed/enabled on your computer. This can be done using the Add Roles and Features Wizard on Server 2008 and Server 2012. See https://technet.microsoft.com/en-us/library/dn482071.aspx for details.

   The .NET Framework 3.5 should be enabled on Windows 10 by default. You can check from **Control Panel > Programs and Features** and clicking the **Turn Windows features on or off** link. The Windows Features dialog should show the .NET Framework with a blue square in the checkbox as shown below. To access Programs and Features in Windows 10, right click the Windows icon in the bottom left of the screen and select **Programs and Features** from the popup menu.

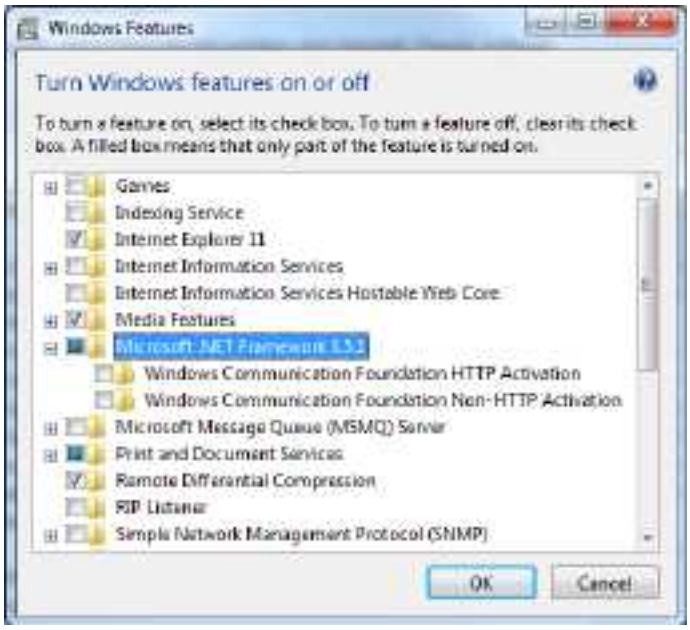


Figure 4 – Turn Windows features on or off

4) Download the appropriate `PostgreSQL jdbc 4.2 jar` file from https://jdbc.postgresql.org/download.html (eg postgresql42.2.9-.jar) and copy it into: `C:\payara5\glassfish\domains\domain1\lib`

5) Download the 4 missing jackson JAR files from https://www.dropbox.com/sh/9uf3rbicau2i1uz/AACuFWjjI31ObaHIuAzELchFa?dl=0 and copy them into this folder: `C:\payara5\glassfish\module`

6) It is advisable to explicitly set the location of the JDK for Payara Server to use. To achieve this edit this file `C:\payara5\glassfish\config\asenv.bat`
Add a new line at the end of this file
`set AS_JAVA=<fully qualified path to your JDK>`
eg. `set AS_JAVA=C:\Program Files\AdoptOpenJDK\jdk-8.0.232.09-hotspot`
OR (if you have used Oracle Java)
`set AS_JAVA=C:\Program Files\Java\jdk1.8.0_231`

7) To establish Payara Server as a service open Windows PowerShell or a Command Prompt as an Administrator and change to the glassfish bin folder
`cd C:\payara5\bin`
`.\asadmin create-service --name PayaraSOLA domain1`
This will create a Windows Service called PayaraSOLA

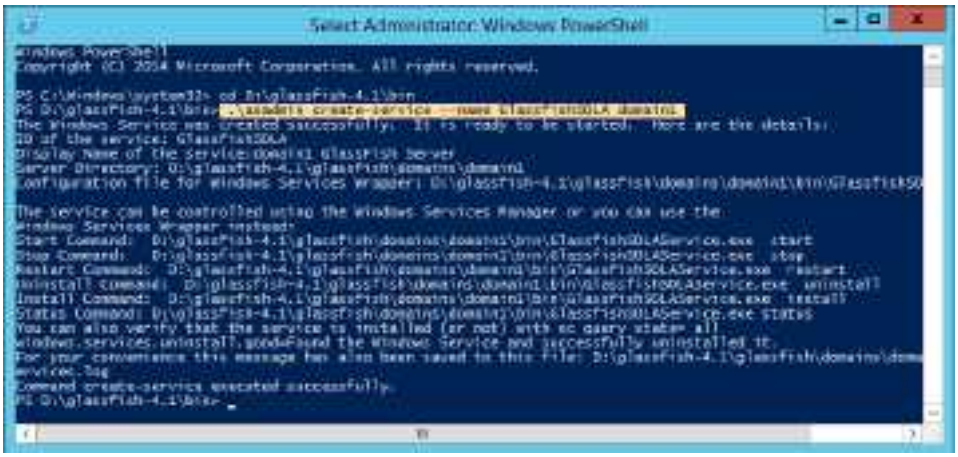


Figure 5 – PowerShell display to create service

8) If you are using Powershell type the following two commands to change the service display name and start it.
   a. `Set-Service -Name PayaraSOLA -DisplayName "Payara SOLA"`
   b. `Start-Service -Name PayaraSOLA`
9) If you are using a Command Prompt type:
   a. `sc config PayaraSOLA DisplayName= "Payara SOLA"`
   b. `sc start PayaraSOLA`

If you are familiar with the Windows Services Control Panel, you will be able to use that to start and stop the new PayaraSOLA service.

2.5 Docker Installation

Docker is operating system level virtualization software. It delivers software in isolated, self contained containers that bundle their own software, libraries and configuration files and communicate with each other through well-defined channels[1].

Docker allows an installation of a SOLA Desktop application to be completed in less than an hour.

Docker Desktop software can be freely download and used and is available on Windows 10 (Professional or Enterprise).

2.5.1 Prerequisites for Docker

Windows

- Windows 10 64-bit: Professional, Enterprise, or Education (Build 15063 or later)
- 4GB system RAM
- Computer with a post 2010 Intel processor and BIOS support for one of the **vt-x**, **vt-d** or **EPT** forms of virtualization. For a more specific check on a computer's ability to run the Docker software use this link to assess the suitability of its Intel processor https://ark.intel.com/content/www/us/en/ark/search/featureFilter.html?productType=873
- BIOS-level hardware virtualization support must be enabled in the BIOS settings. This would typically involve:
    1. Shutdown your computer if it is on
    2. Restart the computer and press **F2** to open the **BIOS** settings. The function key does vary from one motherboard maker to another.
    3. Use the right-arrow key to navigate to the **Advanced** tab.
    4. Select **Virtualization Technology** and press **Enter**
    5. Select **Enable** and press **Enter** again
    6. Save the changes (**F10**) and exit.
    7. Restart your computer.

Figure 6 – Virtualization BIOS Settings

- **Hyper-V** features must be enabled

[1] https://en.wikipedia.org/wiki/Docker_(software)

To check, open the **Start** menu, type in "Turn Windows Features", click on Turn Windows Features on or off hyperlink and make sure both **Hyper-V** checkbox is checked.

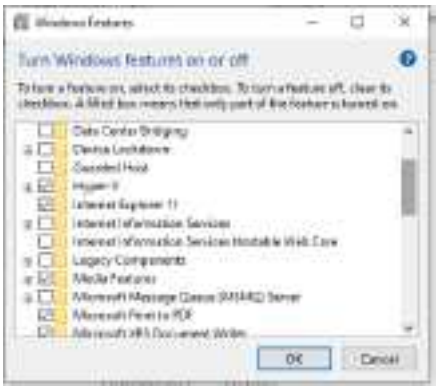


Figure 7 – Enable Windows Hyper-V

2.5.2 Installing Docker Desktop

1. Download the Docker Desktop (Community) software from https://www.docker.com/products/docker-desktop
2. Double click the downloaded file and a standard installation will start
3. **For Windows 10 only**.
   - During installation do **not** click on the checkbox for **Windows container**
   - Run Docker Desktop by clicking on icon in System tray.
   - Select **Settings** and select when you want to **start Docker** and
   - Select Settings-**Shared Drives** and then click the checkbox of the local drive(s) to be accessible to Docker containers (such as the volumes where your SOLA database backup files will be written to)

2.5.3 Online Installation of SOLA Desktop Applications

To install SOLA Desktop applications you need:

- to be connected to the internet and able to download up to 4 Gb of SOLA Desktop installation data
  [available for download from https://github.com/SOLA-FAO/docker - for SOLA Registry; https://github.com/SOLA-SL-FAO/docker for SOLA State Land and https://github.com/SOLA-SR-FAO/docker – for SOLA Systematic]
- to have established a folder (directory) structure to hold the configuration and database backup files such as is illustrated in Figure Figure 19 – Docker Directory Structure for SOLA Desktop Applications.

- to have enrolled as a Docker user here and have a Docker id and password.

docker repositories from github.com
(displaying docker repositories for all SOLA software applications including SOLA Community Server)

sola-docker-data sub-folder
(moved to be directly under $HOME folder)

Figure 8 – Docker Directory Structure for SOLA Desktop Applications

- to have extracted the docker compose (.yml) file appropriate to the SOLA Desktop application from the downloaded SOLA docker github repository and copied it into the compose/registry (or compose/stateland or compose/systemreg) as illustrated in .

Configuration Edits to docker compose (.yml) file:

1. Confirm that Docker is running by typing a simple Docker command in a terminal window such as `$ docker –version`
2. Open the **docker-compose.yml** file in an editor and modify the lines concerning the volume definitions for the automated database backup service (eg reg-backups) and the volumes definition to that service.
   This edit is essential with Windows 10 installations as the volume of the drive to be used must be succeeded by a double forward slash eg /C//sola-docker-data/backups/pg-reg:/backups (and the $HOME/sola-docker-data/backups/pg-reg:/backups deleted or commented out).

   Confirm that the location of the backups folder on the host system is correctly reflected in the (reg-backups) volumes definition in the docker-compose.yml file.

Figure 9 – volume definitions in docker-compose.yml

3. Make further edit changes to the docker-compose.yml file to reflect the **local time zone**. In the downloaded version of docker-compose.yml file, you will note that the multiple references to the docker environment variable **TZ** are set to "Pacific/Auckland" (New Zealand). A list of valid values for the **TZ** variable can be found at https://en.wikipedia.org/wiki/List_of_database_time_zones . Each **TZ** variable reference in the docker-compose.yml file needs to be changed.

Figure 10 – Timezone (TZ) definitions in docker-compose.yml

4. If you need to include Geoserver or JasperReports software packages as part of your SOLA Desktop software setup refer to the docker-compose.yml file for SOLA Community Server at Docker Hub (https://hub.docker.com/repository/docker/npullar/sola-community-server) and copy the Geoserver or JasperReports code segment into your SOLA Desktop docker-compose.yml. Note that for both Geoserver and JasperReports there are volumes and corresponding file directory structures on your host system to create – these are useful in that they store configuration details.
5. Once you are satisfied that your docker-compose.yml reflects the installation requirements and host system structures, type these commands to navigate to where your docker-compose file is located
   eg.
   `cd sola-docker-data\compose\registry`
6. In the (same) terminal window type:

   ```
   $ docker-compose up -d
   ```

   This command will complete the installation and start all the SOLA Desktop Docker services. Be patient as it may take 5 – 10 minutes to complete

2.5.4 Useful Docker Commands – with online SOLA installation

In Ubuntu all docker (and docker-compose) **must** be preceded by **sudo**

`docker-compose stop`	Must be run from the folder of the docker-compose.yml file	Stops all SOLA services
`docker-compose start`	Must be run from the folder of the docker-compose.yml file	Starts all SOLA services
`docker ps -all`		Lists all docker containers
`docker image ls`		Lists all docker images

2.5.5 To uninstall online SOLA Docker installation

This uninstall removes the SOLA Docker installation completely (software and data) except for backups made through the automated database service – ***so use with caution***.

`docker-compose down -v --rmi all --remove-orphans`	Must be run from the folder of the docker-compose.yml file	Stops and completely removes all SOLA services & associated data

2.5.6 Accessing SOLA Desktop components in Docker

When the system hosting Docker does not have an associated domain name then the SOLA Desktop system administrator for your organization will inform you of the IP address of the host system (eg http://192.168.1.1)[2] .

The following url assume an associated domain name of http://sola.org identifies and references the Docker system host.

To access SOLA Desktop application landing page (for client installation)	SOLA Registry SOLA State Land SOLA Systematic	http://sola.org:8080/sola http://sola.org:8080/sola_sl http://sola.org:8080/sola_sr
To access SOLA Web Admin	SOLA Registry SOLA State Land SOLA Systematic	http://sola.org:8080/sola/admin http://sola.org:8080/sola_sl/admin http://sola.org:8080/sola_sr/admin
To access PostgreSQL Admin Console		http://sola.org:80
To access Payara Server 5 Admin Console		http://sola.org:4848

2.5.7 SOLA Database Backups & Restore

Automated PostgreSQL Backups

Currently daily automated database backups are scheduled and these are kept for 7 days before being deleted. Weekly backups are kept for 4 weeks and monthly backups kept for 6 months. These scheduling parameters are configurable.

Backup files are stored to folders outside the Docker containers using a Docker volume. The external location of these backup folders is configurable and typically C:\sola-docker-data\backups\pg-reg (or pg-sl or pg-sys).

Manual PostgreSQL Backups

A manual database backup can be initiated by users with the following Docker command in a terminal window:

```
$ docker run -e POSTGRES_HOST=192.168.1.71 -e POSTGRES_DB=sola -e
POSTGRES_USER=postgres -e POSTGRES_PASSWORD=sola -e POSTGRES_EXTRA_OPTS='--
format=c' -e TZ=Pacific/Auckland -v $HOME/sola-docker-data/backups/pg-
reg:/backups prodrigestivill/postgres-backup-local /backup.sh
```

Manual backups are stored in the ../docker-sola-data/backups/daily sub-folder.

PostgreSQL Restores

Using PGadmin 4 – http://localhost:80

1. Select Database and right click to **Create** a new Database (eg sola2)
2. Select new Database, right click and select **Restore**

[2] IP address can be determined using the following terminal command while logged on to the host system. In Windows10 `ipconfig`

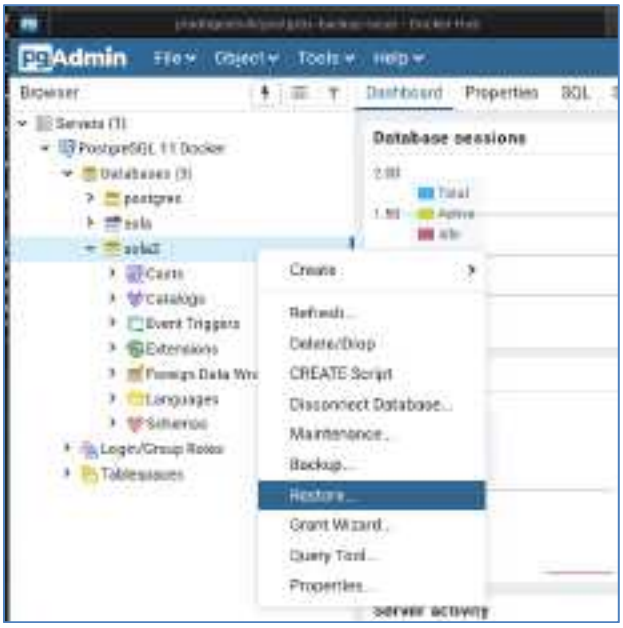


Figure 11 – PGadmin - Restore

3. Click on Filename **….** Button
4. Click on **All Files** & then the toolbar **upload** icon

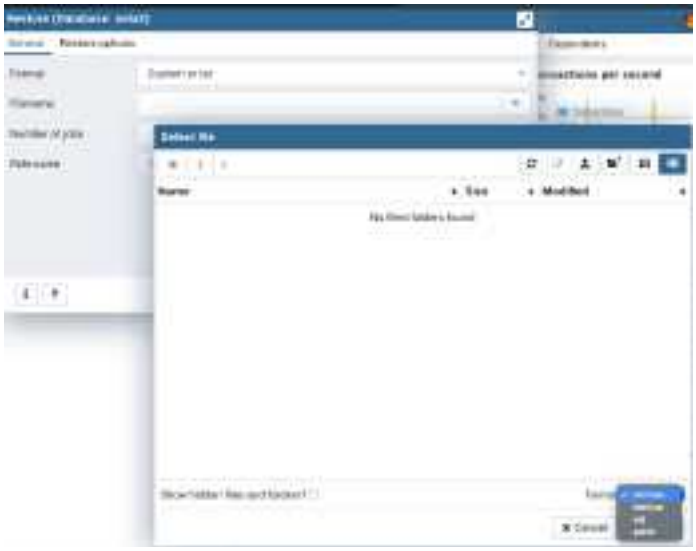


Figure 12 – PGadmin – Restore – Select All files

5. Drag the backup file into PGadmin4 upload panel
6. Click on upload icon and then click on the backup file and the **Select** box

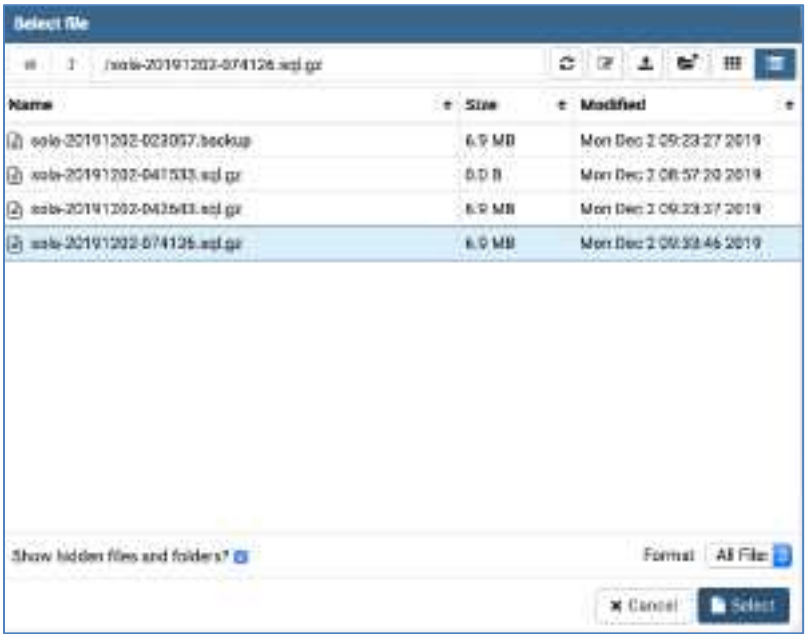


Figure 13 – PGadmin – Restore – Select backup file

7. Click on **Restore** button

2.5.8 Known issues concerning SOLA Docker installations

1. Sometimes with SOLA implementations, Docker appears to “forget” the url of the system hosting Docker and defaults to a Docker container id. As an interim solution, when this happens (more often when logging in to SOLA Web Admin), edit the url in your web browser and manually add the name or IP of the hosting system.

2.5.9 Offline Installation of SOLA Desktop Applications

The commands provided in the following process are docker commands and should be applicable to both Windows and Ubuntu[3].

In preparation for an offline installation, while connected to the internet:

1. On a test Docker installation clear any existing docker images from the system
2. Download the Docker image for the SOLA Desktop server software from Docker Hub
   Example commands are for SOLA Registry (for SOLA State Land use mcdowella/sola-stateland-server & for SOLA Systematic use mcdowella/sola-systemreg-server and correspondingly different output (tar) file names)
   ```
   $ docker pull mcdowella/sola-registry-server
   $ docker save –output sola-reg-srv.tar
   ```
3. Download the Docker image for the SOLA Desktop PostgreSQL database from Docker Hub
   Example commands are for SOLA Registry (for SOLA State Land use mcdowella/sola-stateland-server & for SOLA Systematic use mcdowella/sola-systemreg-server and correspondingly different output (tar) file names)
   ```
   $ docker pull mcdowella/sola-registry-db
   $ docker save –output sola-reg-db.tar
   ```
4. Download the Docker image for the PostgreSQL PGadmin4 software from Docker Hub
   ```
   $ docker pull dpage/pgadmin4
   $ docker save –output pgadmin4.tar
   ```
5. Download the Docker image for the automated PostgreSQL database backup software from Docker Hub
   ```
   $ docker pull prodrigestivill/postgres-backup-local
   ```

[3] This process has been tested successfully on Ubuntu only.

```
$ docker save –output pgBackup.tar
```

6. Copy the (5) output files onto a USB stick
7. Clear the test Docker environment of all downloaded Docker images
   ```
   $ docker image ls
   ```
   Note the first 3 numbers of the IDs of each of the Docker images needs to replace the illustrative IDs in this following command
   ```
   $ docker image rm 432 8a1 9a8 678 09b
   ```
8. Restart (or close down) the Docker software using the Docker icon in the toolbar on your computer
9. Download the Docker software (appropriate to the future host system operating system) and save the installation file to the USB stick with the output (tar) files.

On the host system not connected to the internet:

1. To establish a folder (directory) structure to hold the configuration and database backup files such as the following screenshot illustrates for a SOLA Registry implementation

docker repositories from github.com
(displaying docker repositories for all SOLA software applications including SOLA Community Server)

sola-docker-data sub-folder
(moved to be directly under $HOME folder)

Figure 14 – Docker Directory Structure for SOLA Desktop Applications

2. Install the Docker software as described earlier in Section 3.5.2
3. Copy the output (tar) files into the ..\sola-docker-data\offline-install folder on the host system using this terminal command:
   ```
   $ cd sola-docker-data\offline-install
   ```
   – **Windows**
4. Run the following Terminal commands to load the output (tar) files
   [commands are for SOLA Registry and need to be modified accordingly for SOLA State Land and SOLA Systematic]
   ```
   $ docker load sola-db.tar
   ```
   - modify for SOLA State Land & SOLA Systematic
   ```
   $ docker load sola-svr.tar
   ```
   - modify for SOLA State Land & SOLA Systematic
   ```
   $ docker load pgadmin4.tar
   $ docker load pg-backup-local.tar
   ```
   - modify for SOLA State Land & SOLA Systematic
5. Start the SOLA database service
   ```
   $ docker run -p 5432:5432 --name sola-db -e POSTGRES_PASSWORD=sola -e
   TZ=Pacific/Auckland mcdowella/sola-registry-db
   ```
6. Wait until “Finished load at .. date …time” message has displayed after several minutes. Then open a new terminal window and start the SOLA Desktop server service
   ```
   $ docker run -p 4848:4848 -p 8080:8080 -p 8181:8181 --name sola-srv
   -e TZ=Pacific/Auckland --link sola-db:database mcdowella/sola-
   registry-server
   ```

7. Start the PostgreSQL PGadmin4 service
```
$ docker run -p 80:80 -e 'PGADMIN_DEFAULT_EMAIL=db@flossola.org' -e
'PGADMIN_DEFAULT_PASSWORD=sola' -e TZ=Pacific/Auckland -d
dpage/pgadmin4
```
8. Finally start the PostgreSQL automated database backup service
```
$ docker volume create --driver local --name sola_reg_data
$ docker run -e POSTGRES_HOST=192.168.1.71 -e POSTGRES_DB=sola -e
POSTGRES_USER=postgres -e POSTGRES_PASSWORD=sola -e SCHEDULE=@daily -
e BACKUP_KEEP_DAYS=7 -e BACKUP_KEEP_WEEKS=4 -e BACKUP_KEEP_MONTHS=6 -
e HEALTHCHECK_PORT=85 -e POSTGRES_EXTRA_OPTS='--format=c'  -e
TZ=Pacific/Auckland --name sola-db-backups -v $HOME/sola-docker-
data/backups/pg-reg:/backups prodrigestivill/postgres-backup-local
```
9. Check that all the Docker services are all up and running
```
$ docker ps -all
```
And this is the display you should see

2.5.10 Useful Docker Commands – with offline SOLA installation

The commands below are for SOLA Registry and need to be modified accordingly for SOLA State Land and SOLA Systematic.

`docker ps -all`	Lists all docker containers
`docker stop $(docker ps -a q)`	Stops all SOLA services
`docker ps -all` Note the first 3 digits of container ID of sola-reg-db `docker start 123` Substitute first 3 digits from last command above `docker start $(docker ps -a q)`	To start all SOLA services
`docker image ls`	Lists all docker images

2.5.11 To uninstall offline SOLA Docker installation

This uninstall removes the SOLA Docker installation completely (software and data) except for backups made through the automated database service – ***so use with caution***.

`docker stop $(docker ps -a q)`	Stops all SOLA services
`docker rm $(docker ps -a q)`	Removes all Docker containers
`docker image rm $(docker image ls)`	Removes all Docker images
`docker image ls`	To check there are no Docker images remaining

3. Server Installation on Linux

This installation description steps through installing the SOLA Desktop server software on Ubuntu 18.04. These instructions are for installing OpenJDK 8 (update 232), Payara Server 5 (Build 193.1), PostgreSQL 12 and PostGIS 3 on Ubuntu 18.04.

3.1 Installation Preparation

Ubuntu uses the **apt** tool (Application Package Tool) to install and upgrade applications. **apt** is capable of downloading pre-configured installation packages from various online repositories. The process to do an **apt** installation involves the following Terminal commands:

```
$ sudo apt update
$ sudo apt install <<application name>>
```

For example to install the following applications used in SOLA related routines:

```
$ sudo apt update
$ sudo apt install nautilus-admin
$ sudo apt install git
$ sudo apt install p7zip-full
$ sudo apt install openjdk-8-jre
$ sudo apt -y install icedtea-netx icedtea-plugin
```

However, please note that some of the packages required for SOLA are not directly available through these repositories due to licensing restrictions and/or the lack of a suitable install package and the installation of those packages will require different methods.

3.2 OpenJDK Java

OpenJDK is freely available for Linux distributions and can be installed using apt. Due to licensing restrictions (and future user licence charges, the Oracle Java JDK is not recommended. Steps that need to be followed to download and install Java 8[4] are as follows:

1. To remove Oracle JDK (if previously installed) – ascertain where it has been installed. In the following instructions it is assumed that it has been installed in /opt/jdk/[version] (eg [version] = “jdk1.8.0_231”
   ```
   $ sudo update-alternatives –-remove “java”
   “/opt/jdk/[version]/bin/java”
   $ sudo update-alternatives –-remove “javac”
   “/opt/jdk/[version]/bin/javac”
   $ sudo update-alternatives –-remove “javaws”
   “/opt/jdk/[version]/bin/javaws”
   $ sudo rm -r /opt/jdk/[version]
   ```
2. To install OpenJDK, open a terminal and type
   ```
   $ sudo apt update
   $ sudo apt install openjdk-8-jdk
   $ java -version
   ```
   The following should be displayed:
   ```
   neil@CS1-STK2M364CC:~$ java -version
   openjdk version "1.8.0_232"
   OpenJDK Runtime Environment (build 1.8.0_232-8u232-b09-0ubuntu1~18.04.1-b09)
   OpenJDK 64-Bit Server VM (build 25.232-b09, mixed mode)
   ```
3. To make sure OpenJDK is the default JVM run these instructions:
   ```
   $ sudo update-alternatives –-config java
   ```
   If it is the only version of java you will see the following displayed:
   ```
   There is only one alternative in link group java (providing
   /usr/bin/java): /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
   Nothing to configure.
   ```

[4] The use of Java 9, Java 10 or Java 11 is not recommended for SOLA (January 2020)

If there is more than one version of java, you will be prompted to specify which version should be used as the default JVM (the Java 8 version)

4. If the /etc/environment file does not exist create it with this command:
   `$ sudo touch /etc/environment`
5. Now to set the JAVA_HOME environment variable, (using a Terminal window), make the following commands:
   ```
   $ JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
   $ export JAVA_HOME
   $ JRE_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre
   $ export PATH=$PATH:$JAVA_HOME/bin
   ```
6. Check this change has been applied
   ```
   $ source /etc/environment
   $ echo $JAVA_HOME
   ```
   This should display:
   `/ usr/lib/jvm/java-8-openjdk-amd64`

3.3 PostgreSQL

Download the latest version of PostgreSQL using the apt repository following these steps:

1. Remove any old versions of PostgreSQL and PostGIS
   `$ dpkg -l | grep post` to list the PostgreSQL and PostGIS packages
   `$ sudo apt --purge remove` <<each of the PostgreSQL and PostGIS packages>>
2. Identify the optimal combination combination of different versions of both PostgreSQL and PostGIS by referring to
3. Confirm that for your preferred combination of Ubuntu, PostgreSQL and PostGIS there is the required apt packages by a search of “postgresql-nn1-postgis-nn2-scripts” (where nn1 is the version of PostgreSQL and nn2 is the version of PostGIS) on the https://www.ubuntuupdates.org website to check
   (for instance in December 2019 the optimal combination was Ubuntu 18.04 (bionic), PostgreSQL 12 and PostGIS 3)
3. Import the repository signing key, and update the package lists with this Terminal command
   ```
   $ sudo wget --quiet -O -
   https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add
   –
   $ sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/
   `lsb_release -cs`-pgdg main" >> /etc/apt/sources.list.d/pgdg.list'
   $ sudo apt update
   ```
4. Install PostgreSQL 12 use this Terminal command.
   `$ sudo apt -y install postgresql-12 postgresql-contrib-12`
5. Add a password to the default PostgreSQL user postgres
   ```
   $ sudo su – postgres
   $ psql
   # \password postgres
   ```
   Enter and re-enter your new password
   Exit from psql and revert to Ubuntu root user
   ```
   # \q
   $ logout
   ```
   And this screenshot should be what is displayed

6. Configure PostgreSQL to allow access from other computers
   By default PostgreSQL locked down to prevent access from computers other than the localhost. If your database server will be used as the production server, then it may be a good option to leave PostgreSQL in this locked down state. If the database will be used for testing or training, it may be desirable to access the database server from other computers. To allow these connections it is necessary to make some configuration changes.

   Edit the **/etc/postgres/12/main/pg_hba.conf** file and add the following line (the first in the range of IP assigned by your wifi hotspot/router) to this file
   ```
   host all all 192.168.1.1/24   md5
   ```
   and check that this line is present in the pg_hba.conf file
   ```
   local all all peer
   ```
7. Configure PostgreSQL to accept connections from SOLA applications
   This can be achieved from psql with the ALTER SYSTEM command as follows:
   ```
   $ sudo su - postgres
   $ psql
   # ALTER SYSTEM SET listen_addresses='*';
   \q
   $ logout
   ```
8. Run PostgreSQL as a service
   This command may take some time to complete as it will download and install the PostgreSQL 12 database server. Once the installation is complete, PostgreSQL will be running as a service.

   You can start, stop and restart PostgreSQL, reload the configuration files and list the status of the server using the ***service*** command:

   `$ sudo service postgresql` *[status, reload, start, restart and stop].*

   Therefore make the following commands:
   ```
   $ sudo service postgresql reload
   $ sudo service postgresql restart
   ```
   The configuration files for PostgreSQL will be installed in **/etc/postgres/12/main**
   The data directory for PostgreSQL will be **/var/lib/postgres/12/main**

3.3.1 PostGIS

Using the PostgreSQL and PostGIS version numbers identified at the beginning of the PostgreSQL installation

```
$ sudo apt install postgis postgresql-12-postgis-3 postgresql-12-postgis-3-scripts
```

3.3.2 PostgreSQL extensions

The **uuid-ossp** PostgreSQL extension provides GUID generation functions. It must be installed as an extension into PostgreSQL. The source files for this extension are included in the postgresql-contrib-11 package which is installed as part of the initial PostgreSQL installation.

It is also necessary to create a postgis extension in order to complete the PostGIS installation (note that with this extension, the extension name does not need to be enclosed with double quotes (“).

To create an extension in the database you must be logged in as **postgres**.

```
$ sudo su - postgres
$ psql
# CREATE EXTENSION “uuid-ossp”;
```

```
# CREATE EXTENSION postgis;
# /q
$ exit
```

3.3.3 *PostgreSQL Admin Console (PGadmin4)*

1. Install PGadmin4 (uses the same repository key as PostgreSQL)
   ```
   $ sudo apt install pgadmin4 pgadmin4-apache2 -y
   ```

Figure 15 – Enter email address as PGadmin4 user name

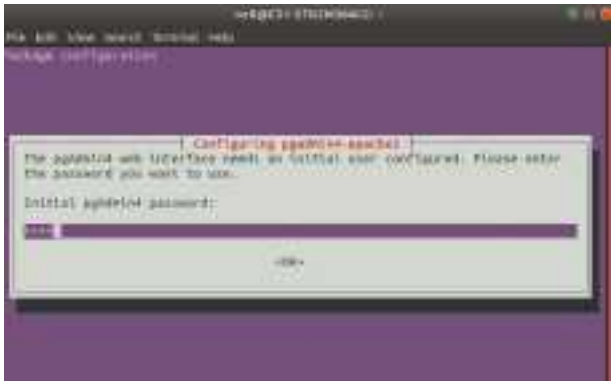


Figure 16 – Enter PGadmin4 password

2. To run PGadmin4, first determine the IP address of the host server with the terminal command:
   ```
   $ hostname -I
   ```

   Using the first IP address from the display resulting from the hostname command, type the url **Error! Hyperlink reference not valid.** into your web browser (eg http://192.168.1.74/pgadmin4 in the screenshot example above). This URL address will work for ALL devices connecting to the same LAN (wireless hotspot).

   The following web page should display.

Figure 17 – PGadmin4 Logon

If the PGadmin logon page does not display, type in http://localhost/pgadmin4 in a web browser on your server. If that displays OK, check that you have made the necessary changes to pg_hba.conf described in Section 5 remembering to reload and restart (using the `$ sudo service postgresql reload` (and then `restart`)

3. If it still does not work you should delete and completely remove all the pgadmin4 packages and reinstall pgadmin4. In December 2019 these packages are **pgadmin4 pgadmin4-apache2 pgadmin4-common pgadmin-doc**

   To confirm what the pgadmin packages are, type the terminal command

   `$ dpkg –list pgadmin4*` to confirm the 4 packages identified above – if there are different packages, note the package names for the next command

   ```
   $ sudo apt  -- purge remove pgadmin4 pgadmin4-apache2 pgadmin4-common
   pgadmin-doc
   ```

   ```
   $ sudo rm -r  /var/lib/pgadmin4
   ```

3.3.4 Creating SOLA Database

Where there is no configured PostgreSQL database backup and it is the very first time you create a new SOLA database in PostgreSQL, follow these steps:

1) As **root**, restart the Postgresql database to release all connections to the template database
   ```
   $ sudo service restart postgresql
   ```
2) To create the database you must be logged in as **postgres**.
   ```
   $ sudo su - postgres -c "createdb -T postgres sola"
   ```
3) SOLA database name is usually assumed to be sola
4) Locate the database folder in the Community Server deployment package, copy this folder to the computer hosting PostgreSQL (if not stored there already).
   For those installing in Ubuntu, the following changes are required to create_database.sh
   Line 19 – check and modify to reflect installed version of Postgresql eg
   `psql="/usr/lib/postgresql/12/bin/psql"`
   AND
   Line 24 will probably need to be modified to reflect runtime version 7zip eg
   `zip_exe="/usr/lib/p7zip/7zr"`
   In this sequence of steps the database folder has been copied to ~/Desktop/install/database-master
   ```
   $ chmod -R +rwx ~/Desktop/install/database-master
   $ cd ~/Desktop/install/database-master
   $ ./create_database.sh
   ```

Remember to check the build.log file to make sure the database build has run without problems.

3.4 Payara Server 5

Payara Server (Build 5.194.1) was the current version of Payara in January 2020 has been used as the application server for SOLA Community Server & SOLA Web Admin.

Payara Server can be installed in two ways. The easiest way to install an appropriately configured Payara Server instance is to copy an already configured instance (remembering to stop Payara Server before making the copy).

Follow these steps if you want to make a fresh Payara Server installation:

1) Download the Payara Server zip bundle from the web site
   https://www.payara.fish/downloads
   This download is approximately 150Mb
2) Unzip downloaded zip file into the /etc/payara5 folder
   `$ sudo unzip payara-5.194.zip -d /etc`
3) Now provide appropriate world/other rights for the /etc/payara5 folder:
   `$ sudo chmod –R ug+rwx /etc/payara5`
   `$ sudo chmod –R ug+rwx /etc/payara5/glassfish/domains/domain1`
   Restart your computer
4) Download the appropriate PostgreSQL jdbc 4.2 jar file from https://jdbc.postgresql.org/download.html (eg postgresql42.2.2-.jar) and copy it into `/etc/payara5/glassfish/domains/domain1/lib` folder
5) Download the 4 missing jackson JAR files from https://www.dropbox.com/sh/9uf3rbicau2i1uz/AACuFWjjl31ObaHIuAzELchFa?dl=0 and copy them into this folder: `/etc/payara5/glassfish/module`
6) It is advisable to explicitly set the location of the JDK for Payara Server to use within the C:\payara5 (or /etc/payara5) software setup. Edit this file in:
   Windows: C:\payara5\glassfish\config\asenv.bat
   Ubuntu: ./etc/payara5/glassfish/config/asenv.conf file
   Add a new line at the end of the applicable file
   Windows: `set AS_JAVA=<fully qualified path to your JDK>`
   eg. `set AS_JAVA= C:\Program Files\AdoptOpenJDK\jdk-8.0.232.09-hotspot`
   OR (if you have used Oracle Java)
   `set AS_JAVA=C:\Program Files\Java\jdk1.8.0_231`
   Ubuntu: `AS_JAVA="<fully qualified path to your JDK> "`
   eg. `AS_JAVA="/usr/lib/jvm/java-8-openjdk-amd64"`
   OR (if you have used Oracle Java)
   `AS_JAVA=="/usr/lib/jvm/jdk1.8.0_231"`
7) Establish Payara Server as a service
   `$ sudo touch /etc/systemd/system/payara_sola.service`
   `$ sudo chmod +rw /etc/systemd/system/payara_sola.service`
   Edit as Administrator the **/etc/systemd/system/payara_sola.service** file and add these lines:

```
[Unit]
Description = Payara Server 5
After = syslog.target network.target
[Service]
ExecStart=/etc/payara5/bin/asadmin start-domain
ExecReload=/etc/payara5/bin/asadmin restart-domain
ExecStop=/etc/payara5/bin/asadmin stop-domain
TimeoutStartSec=0
Type = forking


[Install]
```

```
WantedBy = multi-user.target
```

Create Payara Server Service and Start on Start-up

```
$ sudo systemctl daemon-reload
$ sudo systemctl enable payara_sola
```

To start this service (and similarly stop or restart)

```
$ sudo systemctl start payara_sola
```

To check Payara is running

```
$ sudo systemctl status payara_sola
```

At this point you will have a Paraya Server instance that is configured to run as a service under the root user account.

3.5 Docker Installation

Docker is operating system level virtualization software. It delivers software in isolated, self contained containers that bundle their own software, libraries and configuration files and communicate with each other through well-defined channels[5].

Docker allows an installation of a SOLA Desktop application to be completed in less than an hour.

Docker Desktop software can be freely download and used and is available MacOS. Docker can also be run on Ubuntu using the **Docker Engine** software.

3.5.1 Prerequisites for Docker

MacOS

- Mac hardware must be a 2010 or newer model
- macOS must be 10.13 or newer
- At least 4 GB of RAM.
- VirtualBox prior to version 4.3.30 must **not** be installed as it is not compatible with Docker Desktop.

Ubuntu

To install Docker Engine - Community, you need the 64-bit version of one of these Ubuntu versions:

- Disco 19.04
- Cosmic 18.10
- **Bionic 18.04 (LTS)**
- Xenial 16.04 (LTS)

The Ubuntu 18.04 version runs SOLA Desktop application containers on a computer with 4GB of RAM.

3.5.2 Installing Docker

Mac

1. Download the Docker Desktop (Community) software from https://www.docker.com/products/docker-desktop
2. Double click the downloaded file and a standard installation will start

Ubuntu

[From https://docs.docker.com/install/linux/docker-ce/ubuntu/]

For Unbuntu, you need to install **Docker Engine - Community Edition**. There are two ways of installing Docker Engine, depending on your needs:

- Most users set up Docker's repositories and install from them, for ease of installation and upgrade tasks. **This is the recommended approach**

[5] https://en.wikipedia.org/wiki/Docker_(software)

- Some users download the DEB package, install and manage upgrades completely manually. This is useful in situations such as installing Docker on systems with no access to the internet.

Install using the Docker repositories

Before you install Docker Engine - Community Edition for the first time on a new host machine, you need to set up the Docker repository on your computer. Afterwards, you can install and update Docker from that repository. To set up the repository:

1. Update the `apt` package index:
   ```
   $ sudo apt update
   ```
2. Install packages to allow `apt` to use a repository over HTTPS:
   ```
   $ sudo apt install apt-transport-https ca-certificates curl gnupg-agent software-properties-common
   ```
3. Add Docker's official GPG key:
   ```
   $ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
   ```

   Verify that you now have the key with the fingerprint`9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88`, by searching for the last 8 characters of the fingerprint.
   ```
   $ sudo apt-key fingerprint 0EBFCD88
   ```
   And the following should be displayed:
   ```
   pub   rsa4096 2017-02-22 [SCEA]
         9DC8 5822 9FC7 DD38 854A  E2D8 8D81 803C 0EBF CD88
   uid           [ unknown] Docker Release (CE deb) <docker@docker.com>
   sub   rsa4096 2017-02-22 [S]
   ```
4. Use the following command to set up the **stable** repository.
   ***Note**: The `lsb_release -cs` sub-command below returns the name of your Ubuntu distribution, such as `18.04 bionic`. Docker does not offer any guarantees on untested and unsupported Ubuntu distributions.*
   ```
   $ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
   ```

To install Docker Engine – Community:

3. Update the `apt` package index.
   ```
   $ sudo apt update
   ```
4. Install the *latest version* of Docker Engine - Community and containerd:
   ```
   $ sudo apt install docker-ce docker-ce-cli containerd.io
   $ sudo apt install docker-compose
   ```
5. Reboot the computer

Verify Docker Engine - Community installation

To check the installation and that the Docker service is now running:

```
$ sudo systemctl status docker
```

And the following should display if the Docker service is properly installed and running.

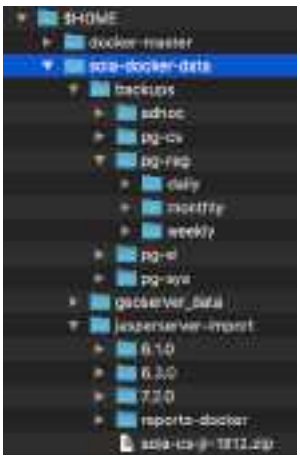
Figure 18 – Post Docker service installation - Ubuntu

3.5.3 Online Installation of SOLA Desktop Applications

To install SOLA Desktop applications you need:

- to be connected to the internet and able to download up to 4 Gb of SOLA Desktop installation data
  [available for download from https://github.com/SOLA-FAO/docker - for SOLA Registry; https://github.com/SOLA-SL-FAO/docker for SOLA State Land and https://github.com/SOLA-SR-FAO/docker – for SOLA Systematic]
- to have established a folder (directory) structure to hold the configuration and database backup files such as is illustrated in Figure Figure 19 – Docker Directory Structure for SOLA Desktop Applications.
- to have enrolled as a Docker user here and have a Docker id and password.

docker repositories from github.com
(displaying docker repositories for all SOLA software applications including SOLA Community Server)



sola-docker-data sub-folder
(moved to be directly under $HOME folder)

Figure 19 – Docker Directory Structure for SOLA Desktop Applications

- to have extracted the docker compose (.yml) file appropriate to the SOLA Desktop application from the downloaded SOLA docker github repository and copied it into the compose/registry (or compose/stateland or compose/systemreg) as illustrated in .

Configuration Edits to docker compose (.yml) file:

7. Confirm that Docker is running by typing a simple Docker command in a terminal window such as `$ sudo docker -version`
8. Open the **docker-compose.yml** file in an editor and modify the lines concerning the volume definitions for the automated database backup service (eg reg-backups) and the volumes definition to that service.

   Confirm that the location of the backups folder on the host system is correctly reflected in the (reg-backups) volumes definition in the docker-compose.yml file.

Figure 20 – volume definitions in docker-compose.yml

9. Make further edit changes to the docker-compose.yml file to reflect the **local time zone**. In the downloaded version of docker-compose.yml file, you will note that the multiple references to the docker environment variable **TZ** are set to "Pacific/Auckland" (New Zealand). A list of valid values for the **TZ** variable can be found at https://en.wikipedia.org/wiki/List_of_database_time_zones . Each **TZ** variable reference in the docker-compose.yml file needs to be changed.

Figure 21 – Timezone (TZ) definitions in docker-compose.yml

10. If you need to include Geoserver or JasperReports software packages as part of your SOLA Desktop software setup refer to the docker-compose.yml file for SOLA Community Server at Docker Hub (https://hub.docker.com/repository/docker/npullar/sola-community-server) and copy the Geoserver or JasperReports code segment into your SOLA Desktop docker-compose.yml. Note that for both Geoserver and JasperReports there are volumes and corresponding file directory structures on your host system to create – these are useful in that they store configuration details.
11. Once you are satisfied that your docker-compose.yml reflects the installation requirements and host system structures, type these commands to navigate to where your docker-compose file is located
    eg.
    ```
    $ cd neilpullar/sola-docker-data/compose/registry
    ```

    In the (same) terminal window type:

    ```
    $ sudo docker-compose up -d
    ```

    This command will complete the installation and start all the SOLA Desktop Docker services. Be patient as it may take 5 – 10 minutes to complete

3.5.4 Useful Docker Commands – with online SOLA installation

In Ubuntu all docker (and docker-compose) **must** be preceded by **sudo**

`docker-compose stop`	Must be run from the folder of the docker-compose.yml file	Stops all SOLA services
`docker-compose start`	Must be run from the folder of the docker-compose.yml file	Starts all SOLA services
`docker ps -all`		Lists all docker containers
`docker image ls`		Lists all docker images

3.5.5 To uninstall online SOLA Docker installation

This uninstall removes the SOLA Docker installation completely (software and data) except for backups made through the automated database service – ***so use with caution***.

`docker-compose down -v --rmi all --remove-orphans`	Must be run from the folder of the docker-compose.yml file	Stops and completely removes all SOLA services & associated data

3.5.6 Accessing SOLA Desktop components in Docker

When the system hosting Docker does not have an associated domain name then the SOLA Desktop system administrator for your organization will inform you of the IP address of the host system (eg http://192.168.1.1)[6] .

The following url assume an associated domain name of http://sola.org identifies and references the Docker system host.

To access SOLA Desktop application landing page (for client installation)	SOLA Registry SOLA State Land SOLA Systematic	http://sola.org:8080/sola http://sola.org:8080/sola_sl http://sola.org:8080/sola_sr
To access SOLA Web Admin	SOLA Registry SOLA State Land SOLA Systematic	http://sola.org:8080/sola/admin http://sola.org:8080/sola_sl/admin http://sola.org:8080/sola_sr/admin
To access PostgreSQL Admin Console		http://sola.org:80
To access Payara Server 5 Admin Console		http://sola.org:4848

3.5.7 SOLA Database Backups & Restore

Automated PostgreSQL Backups

Currently daily automated database backups are scheduled and these are kept for 7 days before being deleted. Weekly backups are kept for 4 weeks and monthly backups kept for 6 months. These scheduling parameters are configurable.

Backup files are stored to folders outside the Docker containers using a Docker volume. The external location of these backup folders is configurable and typically C:\sola-docker-data\backups\pg-reg (or pg-sl or pg-sys) – Windows or ~/sola-docker-data/backups/pg-reg – Ubuntu or MacOS.

Manual PostgreSQL Backups

A manual database backup can be initiated by users with the following Docker command in a terminal window:

[6] IP address can be determined using the following terminal command while logged on to the host system. In Windows10 `ipconfig` or in Ubuntu or MacOS `ifconfig`

```
$ docker run -e POSTGRES_HOST=192.168.1.71 -e POSTGRES_DB=sola -e
POSTGRES_USER=postgres -e POSTGRES_PASSWORD=sola -e POSTGRES_EXTRA_OPTS='--
format=c' -e TZ=Pacific/Auckland -v $HOME/sola-docker-data/backups/pg-
reg:/backups prodrigestivill/postgres-backup-local /backup.sh
```

Manual backups are stored in the ../docker-sola-data/backups/daily sub-folder.

PostgreSQL Restores

Using PGadmin 4 – http://localhost:80

1. Select Database and right click to **Create** a new Database (eg sola2)
2. Select new Database, right click and select **Restore**

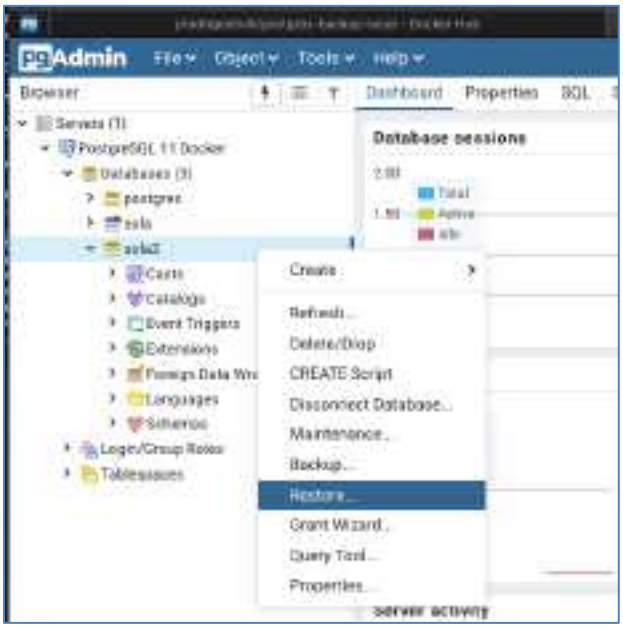


Figure 22 – PGadmin - Restore

3. Click on Filename **….** Button
4. Click on **All Files** & then the toolbar **upload** icon 

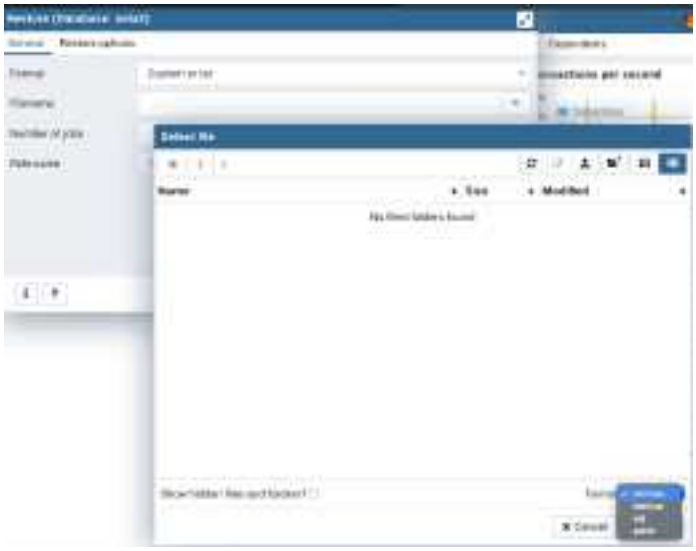


Figure 23 – PGadmin – Restore – Select All files

5. Drag the backup file into PGadmin4 upload panel
6. Click on upload icon  and then click on the backup file and the **Select** box

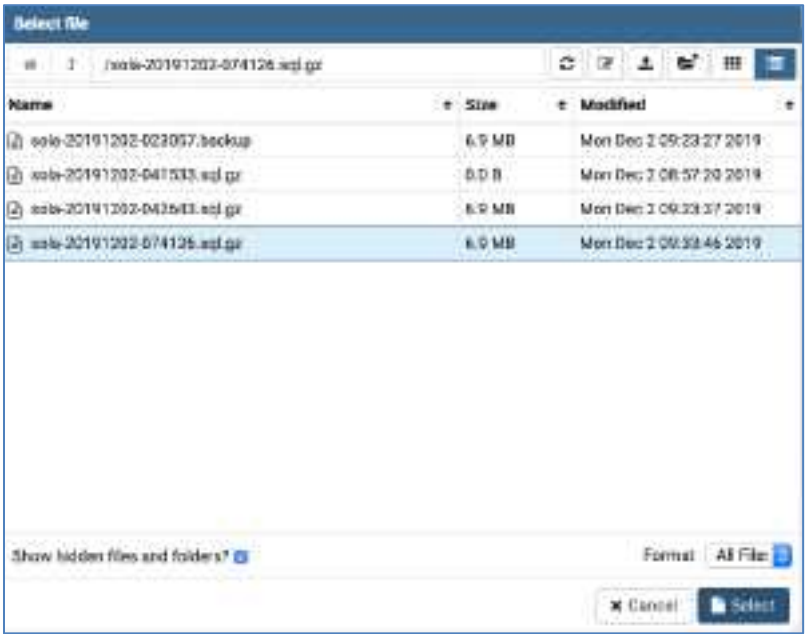


Figure 24 – PGadmin – Restore – Select backup file

7. Click on **Restore** button

3.5.8 Known issues concerning SOLA Docker installations

1. Sometimes with SOLA implementations, Docker appears to “forget” the url of the system hosting Docker and defaults to a Docker container id. As an interim solution, when this happens (more often when logging in to SOLA Web Admin), edit the url in your web browser and manually add the name or IP of the hosting system.

3.5.9 Offline Installation of SOLA Desktop Applications

In preparation for an offline installation, while connected to the internet:

1. On a test Docker installation clear any existing docker images from the system
2. Download the Docker image for the SOLA Desktop server software from Docker Hub
   Example commands are for SOLA Registry (for SOLA State Land use mcdowella/sola-stateland-server & for SOLA Systematic use mcdowella/sola-systemreg-server and correspondingly different output (tar) file names)
   ```
   $ docker pull mcdowella/sola-registry-server
   $ docker save –output sola-reg-srv.tar
   ```
3. Download the Docker image for the SOLA Desktop PostgreSQL database from Docker Hub
   Example commands are for SOLA Registry (for SOLA State Land use mcdowella/sola-stateland-server & for SOLA Systematic use mcdowella/sola-systemreg-server and correspondingly different output (tar) file names)
   ```
   $ docker pull mcdowella/sola-registry-db
   $ docker save –output sola-reg-db.tar
   ```
4. Download the Docker image for the PostgreSQL PGadmin4 software from Docker Hub
   ```
   $ docker pull dpage/pgadmin4
   $ docker save –output pgadmin4.tar
   ```
5. Download the Docker image for the automated PostgreSQL database backup software from Docker Hub
   ```
   $ docker pull prodrigestivill/postgres-backup-local
   $ docker save –output pgBackup.tar
   ```
6. Copy the (5) output files onto a USB stick
7. Clear the test Docker environment of all downloaded Docker images
   ```
   $ docker image ls
   ```

Note the first 3 numbers of the IDs of each of the Docker images needs to replace the illustrative IDs in this following command
`$ docker image rm 432 8a1 9a8 678 09b`

8. Restart (or close down) the Docker software using the Docker icon in the toolbar on your computer
9. Download the Docker software (appropriate to the future host system operating system) and save the installation file to the USB stick with the output (tar) files.

On the host system not connected to the internet:

10. To establish a folder (directory) structure to hold the configuration and database backup files such as the following screenshot illustrates for a SOLA Registry implementation

docker repositories from github.com
(displaying docker repositories for all SOLA software applications including SOLA Community Server)

sola-docker-data sub-folder
(moved to be directly under $HOME folder)

Figure 25 – Docker Directory Structure for SOLA Desktop Applications

11. Install the Docker software as described earlier in Section 3.5.2
12. Copy the output (tar) files into the ..\sola-docker-data\offline-install folder on the host system using this terminal command:
    `$ cd sola-docker-data\offline-install` – **Windows**
    `$ cd sola-docker-data/offline-install` – **Ubuntu or MacOS**
13. Run the following Terminal commands to load the output (tar) files
    [commands are for SOLA Registry and need to be modified accordingly for SOLA State Land and SOLA Systematic]
    `$ docker load sola-db.tar` - modify for SOLA State Land & SOLA Systematic
    `$ docker load sola-svr.tar` - modify for SOLA State Land & SOLA Systematic
    `$ docker load pgadmin4.tar`
    `$ docker load pg-backup-local.tar` - modify for SOLA State Land & SOLA Systematic
14. Start the SOLA database service
    `$ docker run -p 5432:5432 --name sola-db -e POSTGRES_PASSWORD=sola -e TZ=Pacific/Auckland mcdowella/sola-registry-db`
15. Wait until "Finished load at .. date …time" message has displayed after several minutes. Then open a new terminal window and start the SOLA Desktop server service
    `$ docker run -p 4848:4848 -p 8080:8080 -p 8181:8181 --name sola-srv -e TZ=Pacific/Auckland --link sola-db:database mcdowella/sola-registry-server`
16. Start the PostgreSQL PGadmin4 service

```
$ docker run -p 80:80 -e 'PGADMIN_DEFAULT_EMAIL=db@flossola.org' -e
'PGADMIN_DEFAULT_PASSWORD=sola' -e TZ=Pacific/Auckland -d
dpage/pgadmin4
```

17. Finally start the PostgreSQL automated database backup service

```
$ docker volume create --driver local --name sola_reg_data
$ docker run -e POSTGRES_HOST=192.168.1.71 -e POSTGRES_DB=sola -e
POSTGRES_USER=postgres -e POSTGRES_PASSWORD=sola -e SCHEDULE=@daily -
e BACKUP_KEEP_DAYS=7 -e BACKUP_KEEP_WEEKS=4 -e BACKUP_KEEP_MONTHS=6 -
e HEALTHCHECK_PORT=85 -e POSTGRES_EXTRA_OPTS='--format=c'  -e
TZ=Pacific/Auckland --name sola-db-backups -v $HOME/sola-docker-
data/backups/pg-reg:/backups prodrigestivill/postgres-backup-local
```

18. Check that all the Docker services are all up and running

```
$ docker ps -all
```

And this is the display you should see

3.5.10 Useful Docker Commands – with offline SOLA installation

The commands below are for SOLA Registry and need to be modified accordingly for SOLA State Land and SOLA Systematic.

`docker ps -all`	Lists all docker containers
`docker stop $(docker ps -a q)`	Stops all SOLA services
`docker ps -all` Note the first 3 digits of container ID of sola-reg-db `docker start 123` Substitute first 3 digits from last command above `docker start $(docker ps -a q)`	To start all SOLA services
`docker image ls`	Lists all docker images

3.5.11 To uninstall offline SOLA Docker installation

This uninstall removes the SOLA Docker installation completely (software and data) except for backups made through the automated database service – ***so use with caution***.

`docker stop $(docker ps -a q)`	Stops all SOLA services
`docker rm $(docker ps -a q)`	Removes all Docker containers
`docker image rm $(docker image ls)`	Removes all Docker images
`docker image ls`	To check there are no Docker images remaining

4. Configure Payara Server 5

4.1 Start Payara Server

1) Check that there are no Payara Server or Glassfish instances already running:
   In Windows (running Command Prompt as Administrator):
   ```
   $ cd \payara5\bin
   $ .\asadmin list-domains
   ```
   For Ubuntu type:
   ```
   $ cd /etc/payara5/bin
   $ sudo ./asadmin list-domains
   ```
   A message should display
   ```
   domain1 not running
   list-domains executed successfully
   ```
2) Start the SOLA domain, **domain1**.
   Using the `cd` command, navigate to the ..\payara5\bin folder
   In Windows (running Command Prompt as Administrator):
   ```
   $ cd \payara5\bin
   $ .\asadmin start-domain domain1
   ```
   For Ubuntu type:
   ```
   $ cd /etc/payara5/bin
   $ sudo ./asadmin start-domain domain1
   ```
3) Browse to the Payara Server Admin console at **http://localhost:4848** in your web browser.

4.2 Change password (for copy of previously SOLA configured Payara Server 5)

If you are working with a freshly downloaded version of Payara Server then complete all the steps in sections 4.3 - 4.6.

Alternatively, if you are using a copy of Payara Server which has been pre-configured for SOLA, then you only need to change the password for your SOLA PostgreSQL database:

1) Stop the SOLA domain, **domain1**.
   In Windows (running Command Prompt as Administrator):
   ```
   $ cd \payara5\bin
   $ .\asadmin stop-domain domain1
   ```
   For Ubuntu type:
   ```
   $ cd /etc/payara5/bin
   $ sudo ./asadmin stop-domain domain1
   ```
2) Edit the **domain.xml** file as Administrator to be found:
   Windows: C:\payara5\glassfish\domains\domain1\config directory
   Ubuntu: /etc/payara5/glassfish/domains/domain1/config directory
   Search for the string "**<property name="Password" value=**" about line 84

Figure 26 – domain.xml (password defined on line 84)

In the screenshot above, the password is shown as **sola2018**. You need to edit the password to reflect your chosen password (and then make sure you remember it).

3) Do **not** start the (SOLA) Payara Server domain, **domain1**.

4.3 JDBC Connection

1) Make sure you have copied the correct version of PostgreSQL JDBC driver file (dependent on version of Java JDK) into the:
   Windows
   `C:\payara5\glassfish\domains\domain1\lib` folder
   Ubuntu
   `/etc/payara5/glassfish/domains/domain1/lib` folder
2) Start the (SOLA) Payara Server domain, **domain1**.
   In Windows, (running Command Prompt as Administrator)
   Then type the following commands:
   ```
   cd \payara5\bin
   .\asadmin start-domain domain1
   ```
   For Ubuntu:
   ```
   $ cd /etc/payara5/bin
   $ sudo ./asadmin start-domain domain1
   ```
3) Browse to the Payara Admin console at **http://localhost:4848**.
4) In the Payara Server Admin Console, locate the **Resources > JDBC > JDBC Connection Pools** node
5) Click the New button
6) In the New **JDBC Connection Pool** (**General** tab)
   a. Pool Name = sola
   b. Resource Type =javax.sql.ConnectionPoolDataSource
   c. Datasource Classname = org.postgresql.ds.PGConnectionPoolDataSource
7) Click on the **Additional Properties** tab and set the following values:
   i. AllowEncodingChanges = false
   ii. BinaryTransfer = true
   iii. ConnectTimeout = 10
   iv. DatabaseName=<your database name>
   v. [ADD] DefaultRowFetchSize = 0
   vi. DisableColumnSanitiser = false
   vii. GssLib = auto
   viii. HostRecheckSeconds = 10
   ix. LoadBalanceHosts = false
   x. LoginTimeout = 0
   xi. [ADD] LogLevel = 0
   xii. LogUnclosedConnections = false
   xiii. Password = <your postgres password>
   xiv. PortNumber = 5432
   xv. PreparedStatementCacheQueries = 256
   xvi. PreparedStatementCacheSizeMiB = 5
   xvii. PrepareThreshold = 5
   xviii. ProtocolVersion = 0
   xix. ReadOnly = false
   xx. ReceiveBufferSize = -1
   xxi. [ADD] SendBufferSize = -1
   xxii. ServerName = localhost
   xxiii. SocketTimeout = 0
   xxiv. Ssl = false
   xxv. SspiServiceClass = POSTGRES
   xxvi. TargetServerType = any
   xxvii. TcpKeepAlive = false

xxviii. UnknownLength = 2147483647
xxix. Url = jdbc:postgresql://localhost/<database name>
xxx. User = postgres
xxxi. UserSpNego = false

8) Click Finish, then select the Connection Pool you just created and click **Ping**. You should get a Ping Succeeded message.
9) It is now necessary to add a new JNDI resource for the connection pool. The JNDI name is the value used by the Java Naming Directory Interface (JNDI) resolution to reference/locate the sola connection pool resource. Go to the **JDBC > JDBC Resources** node and click **New**...
10) Set the **JNDI Name** to jdbc/sola. It is important to use this name as this is the value referenced by the MyBatis connection configuration files.
11) Select **sola** as the **Pool Name** and click OK to create the new JDBC Resource.

4.4 Security Realm

SOLA delegates authentication of user credentials to a Payara JDBC Security Realm. It is possible to configure a variety of Security Realms in Payara, and any of these could be used for SOLA. The JDBC Security Realm is used because it references the user credentials directly from the sola database which greatly simplifies the administration of user details using SOLA Web Admin.

1) From the **Configurations > server-config > Security > Realms** node, select **New**...
2) Set the following values. All other values should remain blank.
    a. Name = SolaRealm
    b. Class Name = com.sun.enterprise.security.auth.realm.jdbc.JDBCRealm
    c. JAAS Context = jdbcRealm
    d. JNDI = jdbc/sola
    e. User Table = system.appuser
    f. User Name Column = username
    g. Password Column = passwd
    h. Group Table = system.user_roles
    i. Group Name Column = rolename
    j. Digest Algorithm = SHA-256
3) Click **OK** to save the new security realm.
4) Click the **Configurations > server-config > Security** node
5) Check the **Default Principal To Role Mapping** checkbox to enable this setting. SOLA does not include any specific role mapping configuration and relies on the default mapping provided by Payara Server (i.e. roles from the security realm have a one to one mapping to roles in the application based on role name).
6) **Save** this change

4.5 JVM Settings

This section customizes a minimal set of JVM settings for Community Server.

1) Open the Payara Admin Console at http://localhost:4848
2) Click the **Configurations > server-config > JVM Settings** node
3) Click the **JVM Options** tab
4) Change -client to -server. This will ensure the Server JVM is used to run the application domain rather than the client JVM.
5) Change the -Xmx512m option to -Xmx1280m. This will allow the JVM to use up to 1.25GB of RAM.
6) Click the **Add JVM Option** button and enter -Xms640m in the blank Value field
7) **Save** your changes. You should get a message indicating the save was successful. Changes to the JVM settings will also require a **restart** of Payara Server, however you should complete all of the configurations below before restarting.

4.6 Logger Settings

For development purposes it is useful to capture the SQL queries that are sent from Payara Server to the database in the Payara Log. The Payara Log is displayed in the Output View by Netbeans making it very convenient to follow the SQL the application sends to the database as the application is running. Note that these settings are not required in production unless it is necessary to capture the database queries for debugging purposes.

1) Click the **Configurations > server-config > Logger Settings** node
2) Click the **Log Levels** tab
3) Confirm there is a logger name **java.sql** and set the Log Level to **FINE**.
4) Confirm there is a logger called **java.sql.Connection** with the Log Level set to **FINE**. Note that the java.sql and java.sql.Connection loggers must both be set to FINE to log SQL statements.
5) Confirm there is a logger called **org.sola.services** with the Log Level set to **INFO**. This is the logger used by the LogUtility in SOLA. Exception messages and other useful details are recorded in the Glassfish Log through this logger.
6) Confirm there is a logger called **java.sql.ResultSet** with the Log Level set to **OFF**. This logger will capture all results returned from the database however logging every result has a significant performance impact. This logger should only be turned to the FINE level for debugging specific issues and then turned OFF again.
7) **Restart** Payara Server

4.7 Deploy SOLA applications

Installation and database creation files for each of the 3 SOLA Desktop applications (SOLA Registry, SOLA State Land, SOLA Community Server Community Server) along with SOLA Web Admin and the appropriate webstart software for each of the SOLA Desktop applications are available from these Github repositories:
https://github.com/SOLA-FAO/latest-release (SOLA Registry)
https://github.com/SOLA-SL-FAO/latest-release (SOLA State Land)
https://github.com/SOLA-SR-FAO/latest-release (SOLA Systematic)

The following steps assume you have downloaded the relevant SOLA Desktop deployment package.

To deploy the relevant Services and Desktop applications;

SOLA Services ear file Installation

1) Open the Payara Server administration console for domain1 at **http://localhost:4848** and go to **Applications** node.
2) Click the **Deploy** button. On the Deploy Applications or Modules page, click the Choose File button, browse to the SOLA Desktop deployment package and select the relevant SOLA services ear file and click **Open**.
3) On the Deploy Applications or Modules page click the **OK** button. Deployment will take 1 to 2 minutes. **Warning** - Do not navigate away from the Deploy Applications or Modules page during this time otherwise the deployment will be aborted.

SOLA Desktop web-start war file Installation

4) Click the **Deploy** button again. On the Deploy Applications or Modules page, click the Choose File button, browse to the SOLA Desktop deployment package and select the relevant **web-start war** file and click **Open**.
5) On the Deploy Applications or Modules page, change the **Context Root** field from web-start war file name to **reg** (for SOLA Registry), sl (for SOLA State Land) or sys (for SOLA Systematic).
6) Click the **OK** button. Deployment will take 1 to 2 minutes. **Warning** - Do not navigate away from the Deploy Applications or Modules page during this time otherwise the deployment will be aborted.

SOLA Web Admin war file Installation

7) Click the **Deploy** button again. On the Deploy Applications or Modules page, click the Choose File button, browse to the SOLA Desktop deployment package and select the relevant **web-admin war** file and click **Open**.
8) On the Deploy Applications or Modules page, change the **Context Root** field from web-admin war file name to **admin**.
9) Click the **OK** button. Deployment will take 1 to 2 minutes. **Warning** - Do not navigate away from the Deploy Applications or Modules page during this time otherwise the deployment will be aborted. All 3 applications should deploy successfully and show as per the screenshot below.

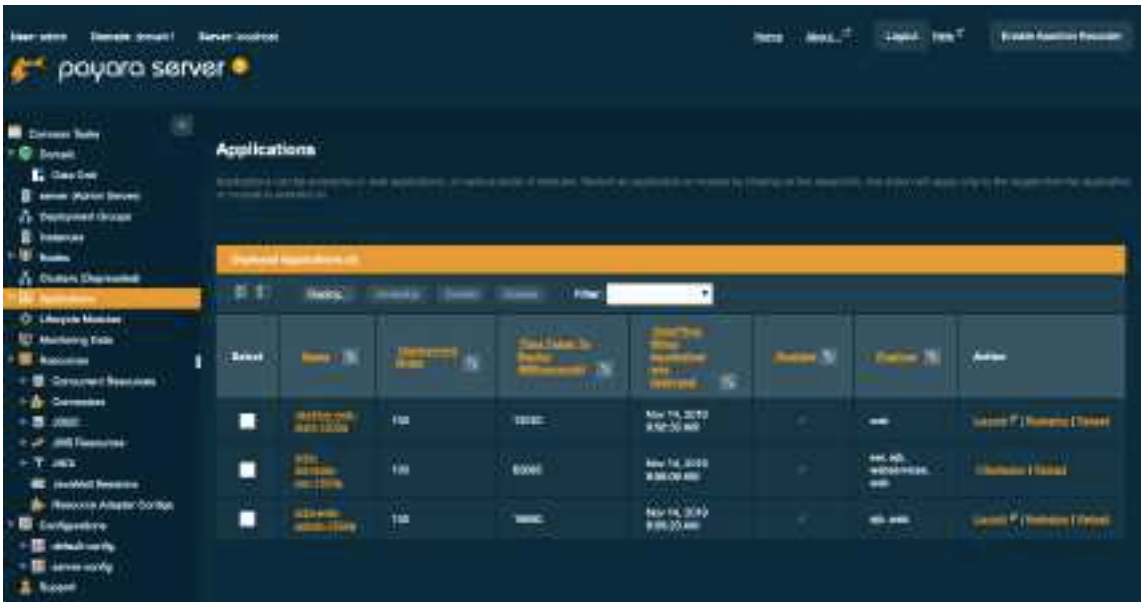


Figure 27 – Payara Admin Console with 3 SOLA Desktop applications deployed

4.8 Deploy GeoServer

In simple implementation scenarios for the SOLA Desktop applications (for instance, one satellite imagery file to be published on a local area network), Geoserver can be installed beside the SOLA Desktop application in the same Payara domain (domain1). For more complicated implementation scenarios, you should consider a separate Payara domain for Geoserver.

Please note that Community Server uses Geoserver version **2.16.0** which is downloadable from http://geoserver.org/download - use the **Web Archive** package

1) Extract the geoserver.war file from the geoserver zip file.
2) Open your web browser and enter the following URL – http://localhost:4848 to open the Payara Admin Console for the domain1 domain.
3) Navigate to the Applications node and click the Deploy button
4) Click Browse and navigate to the geoserver.war file, select and click Open.
5) Click OK to deploy the WAR. Geoserver will process the file and after 1 to 2 minutes you will be returned to the Applications page indicating that the geoserver application is deployed. **Note**: do not navigate away from the Glassfish processing page otherwise the deployment will be aborted.

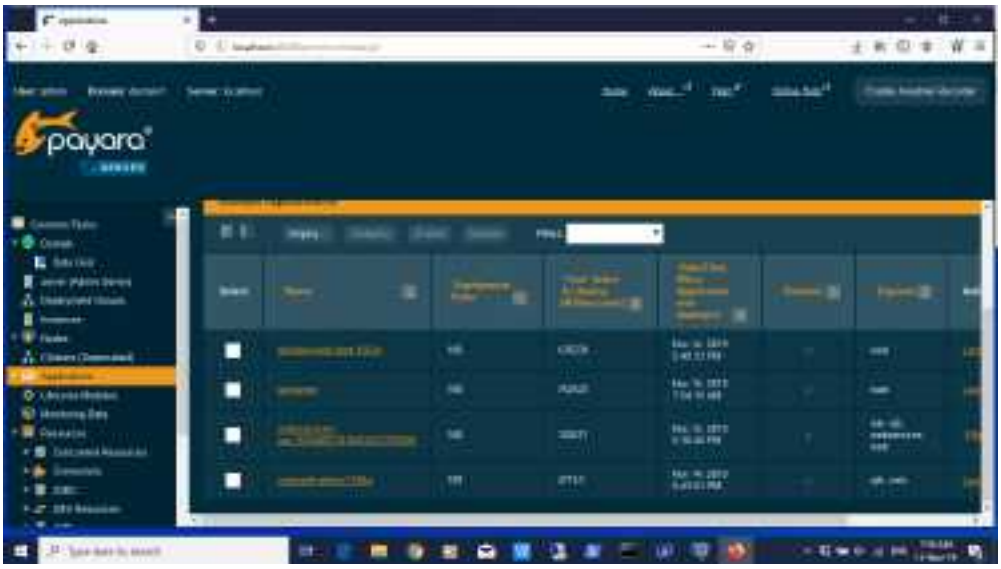


Figure 28 – Payara Admin Console with Geoserver deployed

4.8.1 Publishing GeoServer Map Layers

SOLA Desktop applications uses Geoserver to publish the imagery layers through a WMS service which can then be used within the SOLA Desktop applications. For instance, Geoserver can be used to host satellite / orthophoto imagery[7] relevant to the SOLA Desktop

[7] Note that the possibilities of using Geoserver (data sources that GeoServer can access) are more than what has been described. The standard GeoServer installation supports the loading and serving of the following data formats (list of possible raster layers you can configure in GeoServer):

- GeoTIFF
- GTOPO30
- WorldImage
- ImagePyramid
- ImageMosaic
- Other data sources are supplied as GeoServer extensions. Extensions are downloadable modules that add functionality to GeoServer. Extensions are available at the GeoServer download page.

application. It can also be used to create a SQL view of any table within the SOLA database that has a geometry – for instance parcels over 100 hectares. The symbology of such SQL views is defined using a SLD style definition which is loaded, along with the SQL definition of the view in Geoserver.

SOLA Web Admin (http://localhost:8080/admin) is used to incorporate the WMS service layers (published by Geoserver) into the SOLA Desktop application.

It is also possible to use Geoserver to publish wms layers based on SQL views from the PostgreSQL sola database such as is used to publish the Claims layer in SOLA Community Server.[8]

4.8.2 Publish Satellite Imagery

The following steps assume the satellite (or orthophoto or drone) imagery has been supplied in GeoTIFF format and that the supplied file is less than 2 Gigabytes[9]

1) Start Geoserver console Open the Geoserver admin console by going to http://localhost:8080/geoserver.
   Note: the port number 8080 may vary depending on how you have configured your Glassfish domain.
2) Logon to GeoServer Admin console (default account credentials are username:admin, password:geoserver).
3) Go to Stores in the left side panel.
   - Click Add **New** Store.
   - Under category Raster data sources click **GeoTIFF** (if the imagery file is a geoTIFF file).
     Note: Geoserver can utilise the Image Mozaic plug-in from the open source Geotools library. This plug-in allows for an image pyramid format Geoserver Data Store to be created from a GeoTIFF file and this will enhance performance particularly when the imagery is displayed through the Open Tenure software on mobile devices. Where there is a image pyramid structure select ImageMosaic (rather than WorldImage)
   - For **Workspace** choose **opentenure**,
   - For **Data Source Name** enter community or locality name covered by the imagery.
   - For **URL** choose file:coverages button and use the Browse link to find the geoTIFF file (or the mosaic file folder).
   - Click **Save.** The screen New Layer will appear.
4) Add Layer After saving the Store the screen New Layer will show up.
   - Click **Publish**.
   - For **Name** enter <<community or locality name>>.
   - Make sure both **Enabled** and **Advertised** check boxes are ticked
   - For **Title** enter the same community or locality name.
   - In **Declared SRS** click **Find**. Search for **4326** which is the SRID of WGS84 (geographic coordinates) datum.
5) Make sure **SRS handling** has the value "**Force Native to Declared**".
6) Click on **Compute** from data link and then **Compute from native bounds** link
7) Click **Save**.
8) Test the Layer by navigating to **Layer Preview** in the left side panel.
   - Find in the list of layers the imagery layer name.
   - Under column Common Formats click on **OpenLayers**.

[8] Refer to the SOLA Community Server Administration Guide
[9] If the imagery GeoTIFF file is greater than 2 Gbytes, then it should be converted into a Mosaic file structure that is recognized by Geoserver. There are Geoserver plug-ins that perform this conversion

- You will be brought to a new web/tab page and if everything is fine you should see the image.

4.8.3 Other Useful Map Features

Although not essential, sometimes it is very useful in the map windows of SOLA Desktop applications to have some additional map features to help users to "locate" themselves when they are viewing imagery displayed in both software applications. Typically these map features include roads, rivers, village names and administrative boundaries but this will vary from place to place and their inclusion will depend on availability.

In the following descriptions of how to incorporate these other map features into SOLA Desktop applications, it is assumed that the map features are available in ESRI Shapefile format and that implementing these as wms layers through Geoserver (rather than loading them into the sola PostgreSQL database and customizing the SOLA Desktop software) is the best approach.

OpenStreetMap Data

Often a good source for these map features is from the OpenStreetMap which allows its data to be copied, distributed, transmited and adapted, as long as you credit OpenStreetMap and its contributors. A suggested credit is: "© OpenStreetMap contributors".

If you use this data you must also make it clear that the data is available under the Open Database License, and if using their map tiles, that the cartography is licensed as CC BY-SA. You may do this by linking to the OpenStreetMap copyright page.

Often on the internet you can find a series of OpenStreetMap data extracts for a country nicely organized by data themes that largely coincide with the suggested map feature types. Alternatively, you can use a GIS (such as the open source Q-GIS with the Quick OSM Plugin) to extract the suggested map feature types yourself.

Publishing Shapefile Map Layers in GeoServer

For each map feature shapefile follow these steps in the Geoserver Admin Console (http://localhost:8080/geoserver):

1. For each map feature to be included in Open Tenure and Community Server, go to **Stores** and add new store. Click on **Shapefile** in the **Vector Data Source** list.
2. Select **opentenure** workspace and enter **<<*mapFeature1*>>** as a **Data Source Name** and for URL choose file:data button and use the Browse link to find the Shapefile file. Click Save. The screen New Layer will appear.
3. Add Layer After saving the Store the screen New Layer will show up.
    - Click Publish.
    - For Name enter <<***mapFeature1***>>.
    - Make sure both Enabled and Advertised check boxes are ticked
    - For Title enter the same community or locality name.
    - In Declared SRS click Find. Search for 4326 which is the SRID of WGS84 (geographic coordinates) datum.
    - Add Layer After saving the Store
4. Make sure SRS handling has the value "Force declared".
5. Click on Compute from data link and then Compute from native bounds link
6. Go to the Publishing tab and select an appropriate value from the Default Style values
7. Click Save.
8. Test in the Layer Preview in the left side panel

Once a Geoserver layer has been created for each shapefile, create a new Layer Group incorporating each of these new layers as well as the imagery to be used with Open Tenure:

1. Click on Layer Groups (under Data in left hand panel), and on Add new layer grou[
2. Enter **Name** (eg ImageryWithRefLayers) and **Title** (eg. Imagery with Reference Layers) leave **Workspace** blank.
3. Under Layers, click on **Add Layer** and add the imagery layer
4. Above the Layers, click on the Find Button and enter the appropriate SRID/EPSG code for the coordinate system used by the SOLA Desktop map control. Then click on the returned value and check that the desired SRID/EPSG code is displayed.
5. Click on the Generate Bounds button so that the 4 Bounds fields are filled.
6. Continue and **Add Layer** for each of the Shapefile layers.
7. Click on Save
8. Test in the Layer Preview in the left side panel.

5. Configure SOLA Desktop applications using SOLA Web Admin

To run SOLA Web Admin application open the following link in your web browser
http://localhost:8080/sola/admin for SOLA Registry
http://localhost:8080/sola_sl/admin for SOLA State Land
http://localhost:8080/sola_sr/admin for SOLA Systematic
By default SOLA Web Admin has the login credentials user **test** and password **test**.

5.1 User Management

5.1.1 New Admin User

The first step you should take to configure the Community Server is to add a new admin user account and disable the test account.
In SOLA Web Admin

1) Select **Security => Users** from the main menu. You will see 5 user accounts, but only one account (test) will be active.
2) Click **+ Add** and add a new user account for yourself. On the General tab, remember to click **Active** to activate the user and select the Admin Community Server Security Group to allow the user account to administer the SOLA Desktop server. You can select other Security Groups for the user account depending on your role in the office where the SOLA Desktop application is being implemented.
3) Click Save to save the new user account and verify the new user account is shown on the Users page and that it has a tick in the Active column.
4) On the Users page, you should do one of the following to effectively disable the test user account;
    a. Edit the test user account by clicking the pencil icon and deselecting the Active checkbox, or
    b. Edit the test user account and change the test user password, or
    c. Delete the test account completely.

    If you disable the test user account and forget to active your own account, you can edit the system.appuser table in the database and manually set active to true.

5.1.2 Roles

A key aspect of security is controlling which functions users are able to execute. All SOLA applications, including the SOLA Desktop applications support Role Based Access Control (RBAC). With RBAC, a role is used to represent a function (or group of functions) within the application that requires access control. For a user to execute that function, their user account must be assigned the role that corresponds to the function. As there can be a large number of functions within a system, there can also be a large number of roles. To simplify management of the roles, SOLA supports groups that combine multiple roles into logical groupings. Often job positions are used as the basis for groups as users can then be assigned the groups that correspond to their job position. To allow for people that have multiple roles within an organization, SOLA allows one or more groups to be assigned to a user. The user's access to system functions is then determined from all groups they are associated with.

To view the roles that have been configured for use in the SOLA Desktop applications:
In SOLA Web Admin

1. Select **Security > Roles** page from the main menu. This gives a description for each role.

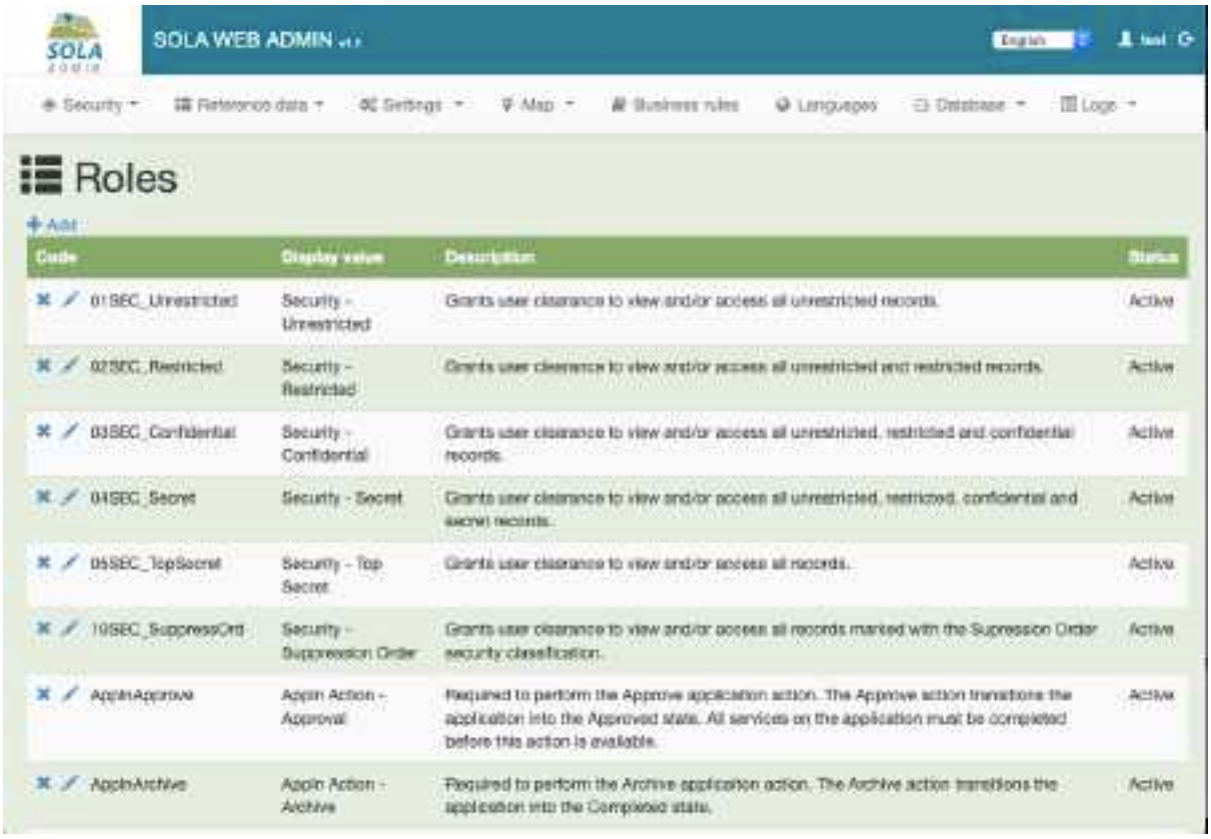


Figure 29 – Web Admin Roles

2. It is possible to add (using the Add link) and edit roles (using the **pencil icon** for the role to be changed) , but *it is recommended that you do not modify these roles* unless a change to the SOLA Desktop software has caused a new role to be created or an existing role to be removed and you have been instructed by your software support specialist to make such a change.

5.1.3 Create User Groups

Groups in SOLA are groups of users who have the same roles. In SOLA Registry typically the Groups that are necessary are:

- Public Counter – receive applications & make enquiries
- Cadastral Processing – undertake cadastral processes
- Registration Processing – undertake registration processes
- Approvers - roles that approve registration & cadastral processes
- Statutory Approvals – roles that incorporate any statutory approval/clearance

Review the business processes that will be supported by the SOLA Desktop software and decide what Groups will be needed in your office and the Roles that are applicable to each group.

To create a new User Group
In SOLA Web Admin

1. Using SOLA Web Admin **Security => Groups** menu option,
2. Click on the **Add** link,
3. Enter the Groups name and tick roles that apply to that Group.
4. Then **Save**

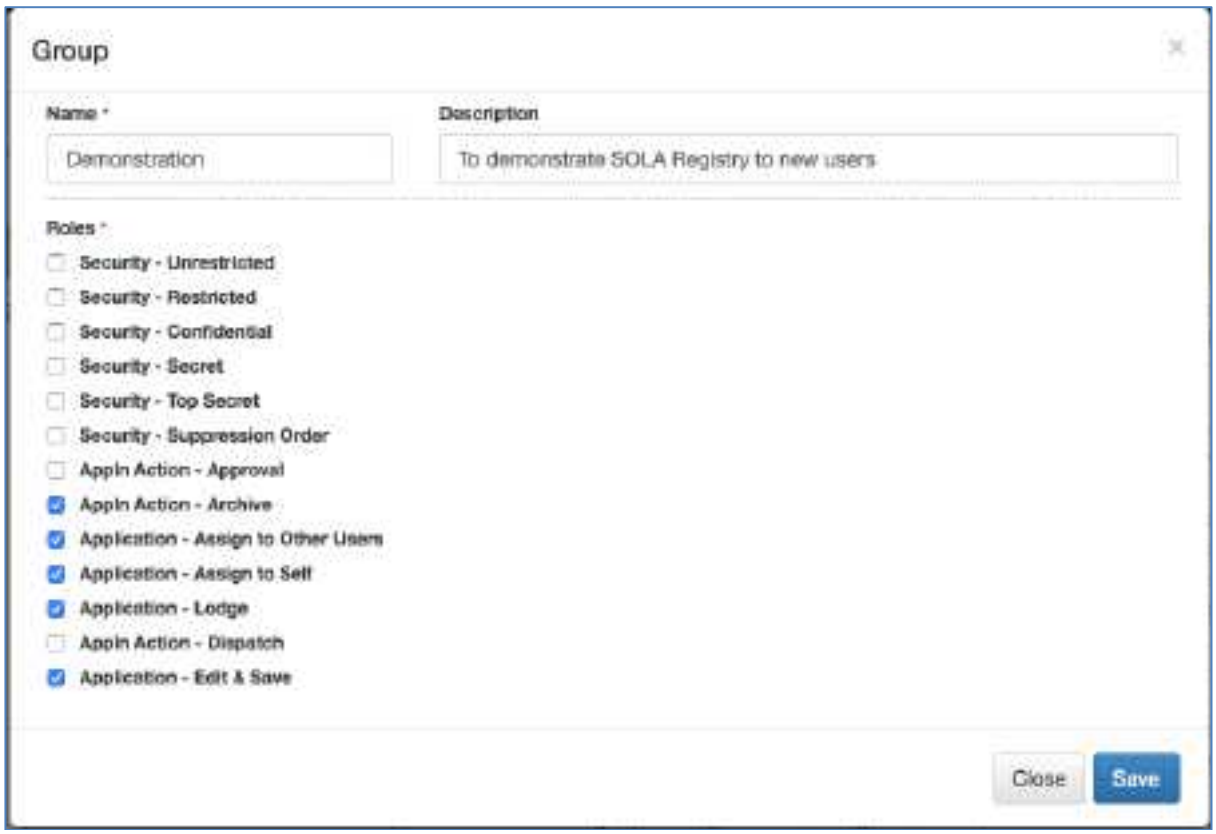


Figure 30 – Web Admin New Group

5.1.4 Enrol Users

User enrolment will be done by the system administrator using SOLA Web Admin.
User name and password should be selected with care to ensure:

- the user remembers the user name and password
- in countries where the language involves a unique font or there are some ambiguities in unicode definitions, user names and password in local font may not work. In such cases, consider using user names and passwords formed from Basic Latin numbers (eg mobile phone number, birthdays)

In SOLA Web Admin
The system administrator will :

1. Select **Security => Users** on the main menu
2. Click on the **Add** link (top left hand side of screen

Figure 31 – Web Admin Add User to Group

3. Fill in the user details and tick the **Active** checkbox

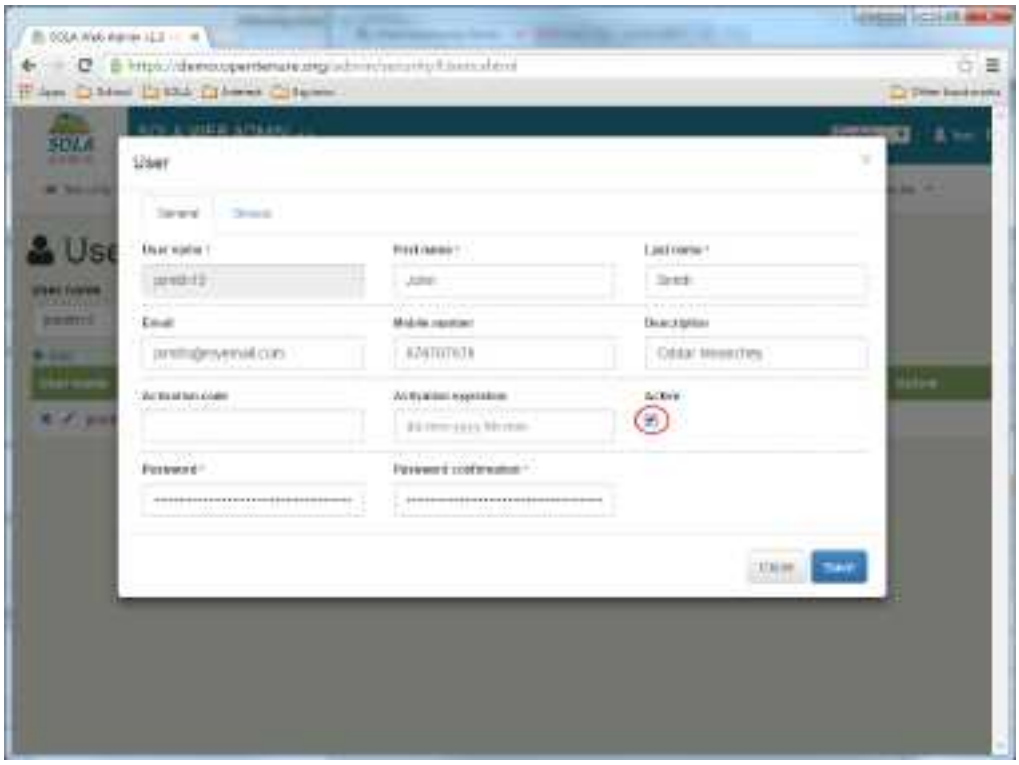


Figure 32 – Web Admin – Make User Active

4. Change to the **Groups tab** and identify which Groups the user will belong to
5. **Save** changes to enable the new user account..

Be sure to configure the appropriate security groups on the Groups tab otherwise the user will not be able to perform any actions in the SOLA Desktop application even if their user account has been activated.

5.1.5 Editing User Details

To edit user details, click the pencil icon beside the username in the Users page. If the Community Server has a large number of users, you can search for specific users by username, first name, last name or security group from the top section of the Users page.

5.1.6 User Passwords

There are three methods for resetting user passwords:

1.	User Reset	Users can change their own password from the User Profile icon in the toolbar of the Desktop app No administrator intervention is necessary.
2.	Administrator Reset	The system administrator can reset a user's password from the User Details page (SOLA Web Admin). This approach should only be used if the user cannot use the Change Password functionality to reset their password such as when their email account is disabled or no longer accessible. After resetting a user's password, the Administrator will need to inform the user of their new password.

Passwords must be one or more characters in length. Users should be encouraged to create secure passwords of at least eight characters; however the SOLA servers do not enforce any password complexity constraints.

5.1.7 Disabling a User

Users can be disabled by unticking the Active checkbox on the User Details popup.
Users will also effectively become disabled if they do not have any security groups assigned to their user account. The User Details popup will not let you remove all security groups from a user, but it is possible to remove security groups from the Groups page which can lead to some users being deallocated from a security group.

5.1.8 Deleting a User

You should only delete a user when a mistake is made in the enrolment of a new user
To delete a user:

1. go to the Users page (SOLA Web Admin main menu select the **Security->** Users menu option)
2. Click the **x icon** beside the username in the Users page.
3. A popup message will confirm the action before deleting the user account.

When you wish to withdraw access to a particular user:

1. go to the Users page (SOLA Web Admin main menu select the **Security->** Users menu option)
2. Click the **Pencil icon** beside the username in the Users page
3. Disable the user account by unticking the **Active checkbox** in the User Details popup.
4. Click on **Save**

5.1.9 User Management details stored in SOLA Database

The SOLA database tables and views from System schema used to store user management details are:

system.appuser	The list of users that can have access to the SOLA application.
system.approle	Contains the list of application security roles used to restrict access to different parts of the application, both on the server and client side.
system.appgroup	Groups application security roles to simplify assignment of roles to individual system users.
system.approle_appgroup	Associates the application security roles to the groups. One role can exist in many groups.
system.appuser_appgroup	Associates users to groups. Each user can be assigned multiple groups.
system.active_users	View. Identifies the users currently active in the system. If the users password has expired, then they are treated as inactive users, unless they are System Administrators. This view is intended to replace the system.appuser table in the SolaRealm User Table property in Payara.
system.user_roles	View. Determines the application security roles assigned to each user. Referenced by the SolaRealm security configuration in Payara.
system.user_pword_expiry	View. Determines the number of days until the user's password expires. Once the number of days reaches 0, users will not be able to log into SOLA Desktop applications unless they have the ManageSecurity role (i.e. role to change manage user accounts) or the NoPasswordExpiry role. To configure the number of days before a password expires, set the pword-expiry-days setting in system.setting table. If this setting is not in place or is deactivated, then a password expiry does not apply.

5.2 Reference Data

The configuration of the SOLA reference data is an essential process in preparing a SOLA Desktop application for use. Reference data irrelevant to your implementation should be marked “inactive” and those elements of reference data deemed relevant should be described in terms of local terminology.
To review the SOLA codelists :
In SOLA Web Admin:

1. Select the **Reference Data** menu option and navigate through the sub-menu options to page corresponding to the reference data page
2. Click on the **Pencil** icon for each row (codelist value) and determine if it is relevant for your SOLA Desktop implementation. If it is click the ensure the appropriate **active checkbox** or leave this checkbox clear is not relevant
3. If the codelist value is relevant review the **Display Values** and amend to reflect the local land administration terminology.
4. Click on **Save** button

The following subsections identify the reference data tables that should be reviewed and modified as described above. It should be noted that all other reference data tables should be left unchanged (these tables areonly be modified by the software developers responsible for the updating of the SOLA Desktop software).

5.2.1 Application Reference Data

Figure 33 – Web Admin – Application Reference data

The **Request type** reference table is key to the customization of the SOLA Desktop software in so far as identifying the customer service requests actioned by your office, ensuring they are described according to local terminology and details such as fees are recorded.
The other Application reference tables should not be modified unless requested by your software support person.

5.2.2 Administrative Reference Data

Figure 34 – Web Admin – Administrative Reference data

The Administrative reference data tables should be reviewed and modified as described in earlier in Section 5.2:

- Condition types – if the land certificate is used subject to certain land use conditions
- Mortgage types
- RRR types – very important to review

5.2.3 Source Reference Data

Figure 35 – Web Admin – Source Reference data

The Source reference data tables should be reviewed and modified as described in earlier in Section 5.2:

- Administrative source (Document) types – very important to review
- Availability statuses (of Document types)

5.2.4 Cadastre Reference Data

Figure 36 – Web Admin – Cadastre Reference data

The Cadastre reference data tables should be reviewed and modified as described in earlier in Section 5.2:

- Land use types

5.2.5 Party Reference Data

Figure 37 – Web Admin – Party Reference data

The Source reference data tables should be reviewed and modified as described in earlier in Section 5.2:

- Gender types
- ID types – very important to review
- Party roles -very important to review
- Party types -very important to review. Please note that a non-Natural Person is an organization or company

5.2.6 Transaction Reference Data

Figure 38 – Web Admin – Transaction Reference data

No changes required for Transaction reference data

5.2.7 System Reference Data

Figure 39 – Web Admin – System Reference data

No changes required

5.2.8 Business Rules Reference Data

Figure 40 – Web Admin – Business rules

No changes required.

5.2.9 Open Tenure Reference Data

This reference data is only relevant to SOLA Community Server web application and not the SOLA Desktop applications. For details refer to the SOLA Community Server Administration Guide.

5.3 System Settings

The (System) settings table in the SOLA database that controls or customizes certain aspects of your SOLA Desktop application

The system settings are:

account-activation-timeout	Account activation timeout in hours. After this time, activation should expire.
command-consolidate	The command for running the consolidation
command-extract	The command for running the extraction
db-utilities-folder	Full path tp PostgreSQL utilities (bin) folder (eg. C:\Program Files\PostgreSQL\12\bin). Used for backup/restore implementation of SOLA Web Admin application
map-east	Eastern most coordinate for map control
map-north	Northern most coordinate for map control
map-shift-tolerance-rural	The shift tolerance of boundary points used in cadastre change in rural areas
map-shift-tolerance-urban	The shift tolerance of boundary points used in cadastre change in urban areas
map-south	Southern most coordinate for map control
map-tolerance	The tolerance that is used while snapping geometries to each other. If two points are within this distance are considered being in the same location.
map-west	Western most coordinate for map control
max-file-size	Maximum file size in KB for uploading
max-uploading-daily-limit	Maximum size of files uploaded daily
moderation-days	Duration of moderation time in days
network-scan-folder	Scan folder path used by digital archive service. This setting is

	disabled by default. It has to be specified only if specific folder path is required (eg network drive). By default, system will use user's home folder + /sola/scan
ot-community-area	Not relevant to SOLA Desktop software
path-to-backlog	The path of the extracted files
path-to-process-log	The path of the process log files
product-code	SOLA product code. reg-SOLA Registry, sys-SOLA Systematic, sl-SOLA State Land
product-name	SOLA product name
public-notification-duration	The notification duration for the public display
pword-expiry-days	The number of days a user's password remains valid.
system-id	A unique number that identifies the installed SOLA system. This unique number is used in the business rule that generate unique identifiers.

These values can be edited using SOLA Web Admin:
In SOLA Web Admin

1. In the **Settings->System settings** menu option review each row
2. Where a change is required click on the **Pencil icon** and edit accordingly
3. Click on the **Save** button where an edit is made

Open Tenure Settings

These settings are only relevant to SOLA Community Server web application and **not** the SOLA Desktop applications. For details refer to the SOLA Community Server Administration Guide.

5.4 Map Layers

Figure 41 – Web Admin – Configure Map Layers

5.4.1 Setting the SRID

Used to define the coordinate system to be used in the SOLA Desktop map control. Uses the SRID (EPSG) code. - https://spatialreference.org/ref/epsg/

5.4.2 Layer Queries

Any changes to these predefined queries should only be done by your software developer support staff

5.4.3 Layers

Figure 42 – Web Admin – Map Layers

A number of predefined layers are provided when a SOLA Desktop software application is first installed. The parcel related layers should not be changed except by the software developers providing software support. However all the other layers will need to be changed to reflect the availability of appropriate geospatial datasets or because the details of local data are different. For instance the url of the orthophoto (or satellite imagery) is different.

5.5 Business Rules

SOLA Desktop applications include various business rules to control the work flow and to protect the integrity of the business processes associated with individual services and applications. The business rules also protect the integrity of the data generated or changed as a result of services and applications.

The SOLA Desktop applications business rules can be defined in Drools[10] or (PostgreSQL) SQL. All the default business rules currently (November 2019) used by the SOLA Desktop applications are defined in SQL.

As part of the initial configuration of a new implementation of one of the SOLA Desktop applications, these business rules should be reviewed. As you can see from the following examples, there are a number of business rules that may need to be changed.

area-check-percentage-newareas-oldareas	sql	The difference between the total of the new parcels official areas and the total of the old parcels official areas should not be greater than 0.1%
ba_unit-spatial_unit-area-comparison	sql	The difference between the property area (_$property_area) and the parcel area (_$parcel_area) must be less than 1%
public-display-check-complete-status	sql	At least 90% of the parcels must have an associated Systematic Application with complete status.
spatial-unit-group-inside-other-spatial-unit-group	sql	Spatial unit groups that are not of the top hierarchy must be spatially inside another spatial unit group with hierarchy which is a level up. Tolerance of 0.5 m is applied.

[10] https://www.drools.org

Figure 43 – Example SOLA Desktop Business Rules

To review the business rules:

In SOLA Web Admin

1. Click on the **Business rules** menu option review
2. Click on the **Search** button

Figure 44 – Web Admin – Business Rule Search

3. Where a change is required click on the **Pencil icon** for that business rule

Figure 45 – Select Business Rule for Editing

4. Make the change to the Feedback field at the bottom of the page in the applicable language (in the screenshot below English)

Figure 46 – Business Rule Details

5. Click on the **Definitions** tab and make the change to the business rule (in the screen shot the allowable difference in compensation paid and the initial offer was changed to 15%)

Figure 47 – Modify SOLA Business Rule SQL Definition

6. Click on **Save** icon and return to the **General** tab
7. The business rule **Validation** tab provides the means to modify the working of the business rule when validation is run (all the applicable business rules at that particular stage of the service and application). *Changes in this tab should only be made when instructed by your software support person*.

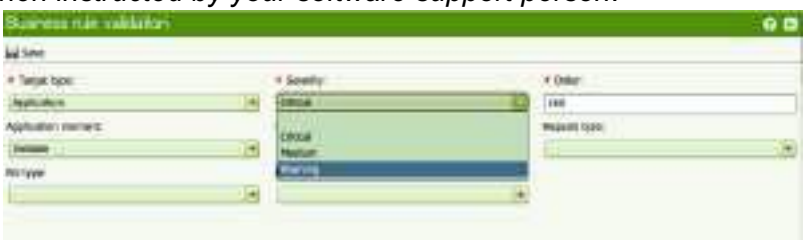


Figure 48 – Modify SOLA Business Rule Severity

8. On the **General** tab when all changes have been made to the business rule, click on the **Save & Close** button

Figure 49 – Save SOLA Business Rule changes

9. Click on the **OK** button

Please note that at the last revision of this Administration Guide (January 2020), a software fault prevented users from accessing the Business Rules pages in SOLA Admin Web. The temporary work-around is to use the old desktop version of SOLA Admin – go to the landing page for the SOLA Desktop application of interest to you (eg *http://localhost:8080/sola* or *http://localhost:8080/sola_sl* etc) and use the other link to install the old Admin desktop application.

5.6 Language Configuration

Where a language localization has been done for a SOLA Desktop application (see Section 8) then you need to make sure that all the localized versions are recognized as "**active**" and to define the **default** language to be used in your implementation.

To complete this language configuration:

In SOLA Web Admin

1. Click on the **Languages** menu option review

Figure 50 – Web Admin – Languages

2. Click on the Pencil icon for each language that is to be made active, made inactive or made the default language
3. Make the appropriate edits of the applicable checkbox items

Figure 51 – Web Admin – Language Details

4. Click on **Save** button

5.7 Enable Mail Service

If the mail service is to be used (and you have configured the mail service as described in Section 6) you must enable it with these steps:

1) Select on the main menu **Settings => System** settings.
2) Look for email-enable-email-service item and click the pencil icon.
3) Make sure **Active** checkbox is ticked and enter **1** as a value.
4) Click **Save** button.
5) Locate **email-admin-address** item and edit it. Provide system administrator's email address.
6) Locate **email-admin-name** and edit it. Provide system administrator's full name.

5.8 Web Admin Database Backup

You can also setup database backup settings. These values will be used if you want to do adhoc database backup and restore from SOLA Web Admin.

Adhoc database backups and restores can also be done using PGadmin or with a Docker command.

1) Select on the main menu Settings => System settings.
2) Look for **db-utilities-folder** item and click edit button on it (pencil icon).
3) Provide full path to the PostgreSQL bin folder (e.g. C:\Program Files\PostgreSQL\12\bin [Windows] or for Ubuntu/etc/postgresql/12/bin) and click **Save** button.

Please note that for Docker implementations, the database backup through SOLA Web Admin will not work.

In all implementations (including Docker implementations), the PGadmin application provides database backup and restore functionality.

6. SOLA Registry Email Service

It is possible to configure SOLA Registry to send system generated messages to users for certain events.

6.1 Email variables

In SOLA Web Admin
Click on the **Settings-System settings** menu option.

1. Then select each of these email setting variables and enter appropriate details and messages for your SOLA Registry implementation
2. Make sure the **Active** checkbox is ticked.
3. Click on the Save button when each variable edit is complete

email-admin-address	Email address of server administrator. If empty, no notifications will be sent
email-admin-name	Name of server administrator
email-body-format	Message body format. text - for simple text format, html - for html format
email-enable-email-service	Enables or disables email service. 1 - enable, 0 - disable
email-mailer-jndi-name	Configured mailer service JNDI name
email-msg-failed-send-body	Message text for delivery failure
email-msg-failed-send-subject	Subject text for delivery failure of message
email-msg-notifiable-subject	Action on Interest subject text
email-msg-notifiable-submit-body	Action on Interest body text
email-send-attempts1	Number of attempts to send email with first interval
email-send-attempts2	Number of attempts to send email with second interval
email-send-attempts3	Number of attempts to send email with third interval
email-send-interval1	Time interval in minutes for the first attempt to send email message.
email-send-interval2	Time interval in minutes for the second attempt to send email message.
email-send-interval3	Time interval in minutes for the third attempt to send email message.
email-service-interval	Time interval in seconds for email service to check and process scheduled messages.

Figure 52 – Email Settings in Web Admin Settings

If one of these email variables is not recorded in the Web Admin System Settings form, then the default value will apply. In those cases when you want to apply some other value, then you will need to add a new record and make sure the setting name is entered exactly as is given below.

6.2 Payara Server JNDI Mail Service

In addition to creating or editing email variables as system settings, it is also necessary to make certain changes to the Payara Server setup.
In the example below settings for Google mail server are given. If you intend to use a different email provider, modify the properties listed in step 5 to match your email provider details.

1. Open Payara Server administration console at http://localhost:4848 and go to **Resources => JavaMail Sessions**
2. Click "**New**" button to create new mail service.
3. Enter the following settings:
    a. JNDI Name = mail/sola
    b. Mail Host = smtp.gmail.com

c. Default User = SOLA Mailer
d. Default Sender Address = <*your_mailbox@gmail.com*>

4. Leave **Advanced** settings with default values
5. In the **Additional Properties** table add the following properties:
   a. mail-smtp-host = smtp.gmail.com
   b. mail-smtp-password = <*your_mailbox_password*>
   c. mail-smtp-socketFactory-class = javax.net.ssl.SSLSocketFactory
   d. mail-smtp-auth = true
   e. mail-smtp-socketFactory-port = 465
   f. mail-smtp-port = 465
   g. mail-smtp-starttls-enable = true
   h. mail.smtp.connectiontimeout = 60000
   i. mail.smtp.timeout = 180000
   j. mail-smtp-user = <your_mailbox@gmail.com>
   k. mail-smtp-socketFactory-fallback = false
6. Save settings by clicking "**Save**" button

6.3 Email Log

In the SOLA Web Admin application, under the **Logs – Mail** queue menu option it is possible to view a table of email messages generated by the server with details regarding the recipients, the subject line and body of the message and whether the email had been successfully sent.
If users report that they are not receiving SOLA Registry emails, this log is a good starting point in trouble-shooting such problems.

6.4 SOLA Database Table for Email

The SOLA database table used with system generated emails is:

system.email	Contains all emails that have been logged, but are yet to be sent by the server. Emails that are sent successfully by server are deleted from this table. To view the history of all emails that have been sent by the server, see the Sent folder of the email account that the server uses for the system generated email.

7. SOLA Desktop Client Installation

This section describes the steps to install the Release 1503a Web Start Application on your computer. The WebStart application installs the SOLA Desktop Client software**.**

7.1 Install Java Runtime Environment (JRE)

The Java SE Runtime Environment 8 is required for the SOLA Desktop Applications. All versions of Java 8 are compatible with the SOLA Desktop Applications. If you do not have Java 8 installed, you can download the latest update from : https://adoptopenjdk.net/releases.html.

- **Scroll down** to select the Java Runtime Environment (JRE) 32bit **.msi** file for Windows or (64bit) **.pkg** file for macOS and download the file appropriate to your client platform (*except for Ubuntu – see below*). This download should be approximately 36Mb.
- **For Ubuntu 18.04** make the following Terminal commands

```
$ sudo apt update
$ sudo apt install openjdk-8-jre
```

7.2 Install OpenWebStart

In order to use a Java WebStart (JWS) installation a Java package that includes the *javaws* routine needs to be installed. Previously this was part of the Oracle JRE and JDK but with the deprecation of Oracle JWS from version 9, its removal altogether from version 11, versions 7 & 8 now no longer being maintained and Oracle about to institute some form of licence charges, there is a need for SOLA users to transition to non-Oracle versions of Java. This is particularly critical for SOLA Desktop application users where the main installation method uses JWS.

Currently (January 2020):

- Only **OpenWebStart** provides a non-Oracle JWS option for Windows, Mac and Ubuntu users but it is new product (released in second half of 2019) and based on the JNLP JSR-56 specification (Appendix A) which has had 6 maintenance releases since its launch in 2000. There are concerns that the JNLP format used in SOLA is not completely compatible with the JNLP specification used by OpenWebStart.
- **IcedTea-Web** provides a reliable JWS option for Ubuntu users through an apt installation although the SOLA software code does need to be reviewed to ensure current JNLP compatibility. **IcedTea-Web** is used with the `itweb-settings` command line to open a console to change settings (including security). It can also be used to make other changes to facilitate the use of **IcedTea-Web** with the SOLA jnlp file (sola-desktop.jnlp and sola-desktop-test.jnlp)
  eg. `itweb-settings set deployment.manifest.attributes.check NONE`

When these current (January 2020) JWS options or Oracle Java is unable to complete a web start installations, then the alternative SOLA Desktop client installation (described in Section 7.5) should be used as an interim solution.

~~*7.2.1 Windows Installation*~~

~~1) Go to the OpenWebStart download area (https://openwebstart.com/download/) and download an installer.~~
   - ~~Chose the `OpenWebStart_windows-x64_1_1_1.zip` if you have a default Windows installation.~~
   - ~~Chose the `OpenWebStart_windows-x32_1_1_1.zip` if you have a 32-bit Windows installation.~~

~~2) Run the installer.~~
~~3) Click **Next** to start the OpenWebStart installation.~~
~~4) Enable the checkbox to associate the .JNLP suffix with OpenWebStart, and click **Next**.~~
~~5) Wait while Setup installs OpenWebStart on your computer.~~

6) Click **Finish** on the completion screen to close the wizard.

7.2.2 Ubuntu Installation

1) Go to the OpenWebStart download area (https://openwebstart.com/download/) and download OpenWebStart_linux_1_1_1.deb.
2) Change into the Downloads directory and run the installer from the terminal:

```
$ cd Downloads
$ sudo dpkg -i OpenWebStart_linux_1_1_1.deb
```

7.3 Upgrading the SOLA Desktop Client

If you are upgrading from an earlier version of the SOLA Web Start application, simply launch SOLA from the desktop shortcut. Java Web Start will automatically download the new release and upgrade the SOLA Desktop Client for you. Be aware that the upgrade will also replace your desktop shortcut and you may need to reconfigure the shortcut (Section 7.4.1).

If you receive an “Unable to launch the application” error after attempting to upgrade SOLA, it is recommended that you uninstall then reinstall the SOLA web start application. Refer to the Uninstall description (Section 7.7) for information on how to correctly uninstall a Java Web Start application.

7.4 Installation

These installation instructions are for the client SOLA Desktop application using Java Web Start.

1. Using your web browser, navigate to the server SOLA Desktop initial “landing page” url where the SOLA Desktop server software is running
   eg http://192.168.1.100/sola for SOLA Registry;
   http://192.168.1.100/sola_sl for SOLA State Land and
   http://192.168.1.100/sola_sr for SOLA Systematic

Figure 53 – SOLA Desktop initial landing page

2. Right click the SOLA Desktop Web Start link and choose **Save link as...** and save the sola-desktop.jnlp file to a known location on your local file system. [*Ignore and do not try and install the now obsolete SOLA Admin desktop app]*
3. Once the file has been saved on your file system, browse to the sola-desktop.jnlp file using Windows Internet Explorer or equivalent and double click the file. You may see a Java 8 splash screen displayed.
4. Followed by the Starting application... dialog

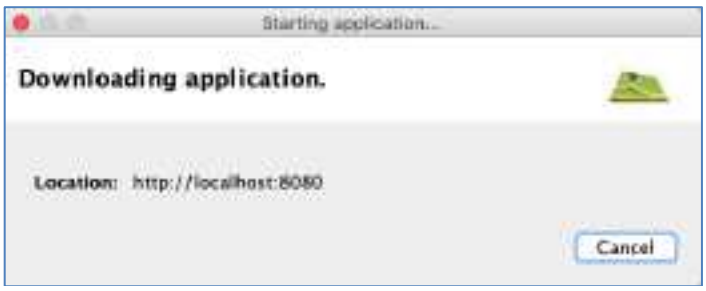


Figure 54 – Client installation starting ... dialog

5. Once the download is complete, the installation may pause for several minutes while it performs verification of the installation package. Please wait until this action has completed.
6. When prompted with the **Do you want to run this application?** warning, tick *Do not show this again…* and choose **Run**

Figure 55 – Client installation warning dialog

7. The SOLA Desktop will start automatically. At the login screen, enter User **test** and Password **test.** You will be presented with the SOLA Dashboard.

7.4.1 Configuring the Desktop Shortcut

The installation will place a shortcut to the SOLA Desktop application on your computer desktop. This shortcut may be configured to use an older version of Java which will prevent the application from being launched. Test the shortcut by double clicking it. If you get the **Unable to launch application** error displayed due to the requested version of the JRE, perform the following steps. Note that the Properties dialog illustrated below may vary in appearance depending in the operating system you are using.

1. Right click the SOLA Desktop Web Start shortcut and choose Properties

Figure 56 - SOLA Desktop Web Start desktop shortcut

2. Update the Target: field on the Shortcut tab by removing any path (e.g. C:\Windows\System32\) before the javaws.exe command. This will ensure the shortcut uses the default Java installation on the computer. You may need to scroll left to reach the start of the Target: field.

Figure 57 - SOLA Desktop Web Start Properties

7.5 Alternative Installation

This approach should be viewed as an interim solution only to be used when there are difficulties in completing a webstart installation and there will be a delay in resolving those difficulties.

It assumes that:

1. the SOLA Desktop server based software has been installed successfully and that the IP address of the host server is known.
   - To determine the IP address of the server, logon to the server and in a Command (Terminal) window, type the following command:
     `ipconfig` **Windows**

Figure 58 – Windows ipconfig display

In the display above, the IP address is the value for the Ethernet adapter Ethernet, IPv4 Address – 192.168.1.76

In **Ubuntu** the most useful command to determine the IP address is:
`$ hostname -I`

Figure 59 – Ubuntu hostname display

In the display above, the IP address is the first value – 192.168.1.74

- To check this IP value (and the LAN connectivity) logon to one of the workstations where you will be using the SOLA Desktop client software. Open a Command (Terminal) window and type:
  `ping 192.168.1.76` modify IP address to reflect your server's IP address.

  You should receive a display like in the following figure (the `$ ping` command is the same in **Ubuntu**)

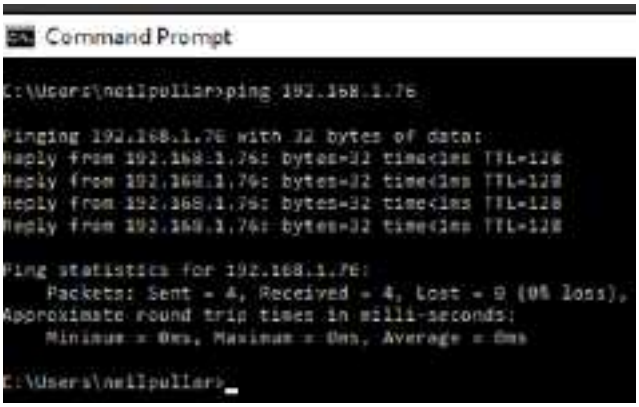


Figure 60 – Successful ping of server IP address

2. a custom build of the **Clients Desktop** Netbeans project has been completed and the resulting sola-desktop.jar file and associated lib folder has been copied onto the client workstation (on the same LAN as the server) where the SOLA Desktop software will run. Ideally, the custom build of the **Clients Desktop** Netbeans project should be done by a SOLA software specialist but if that is not feasible, these are the steps using Netbeans to follow:
   - obtain a complete copy of your SOLA Desktop software possibly using the git-slave process described in Section 8.1 but noting that if your version of the SOLA Desktop software is a customized version, the references will be different to the uncustomized versions (which include "FAO" in the name of the github repositories)

- In Netbeans open the **Clients Desktop** project (having navigated from code-clients-swing-**desktop** in the Open Project process)

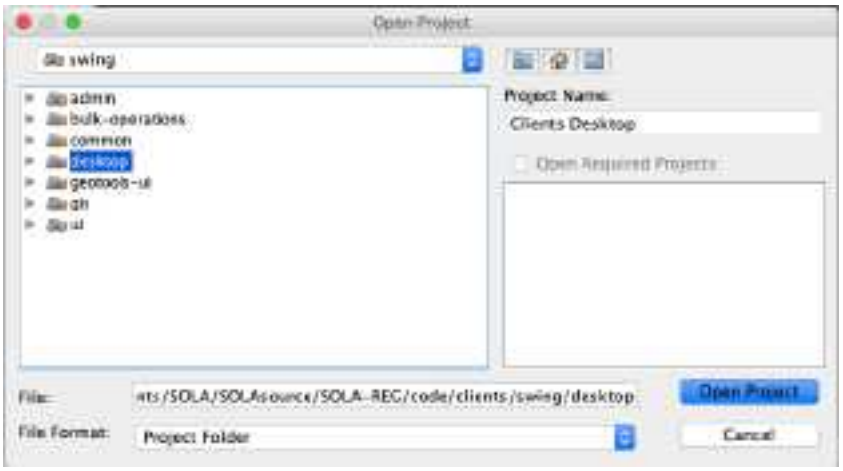


Figure 61 – Netbeans – open Clients Desktop project

- In Netbeans, navigation panel, navigate Clients Desktop – Other Sources – config and open the **wsconfig.properties** with a double click

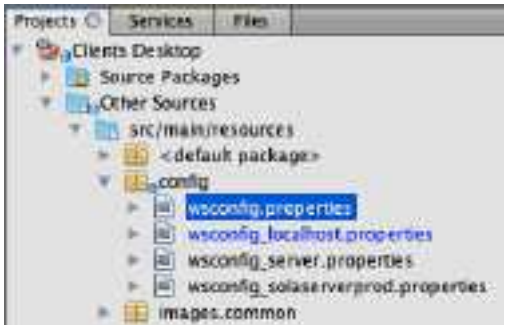


Figure 62 – Netbeans – open wsconfig.properties

- In Netbeans, edit the wsconfig.properties file to replace all references to "localhost" with the servers IP address. **Save**

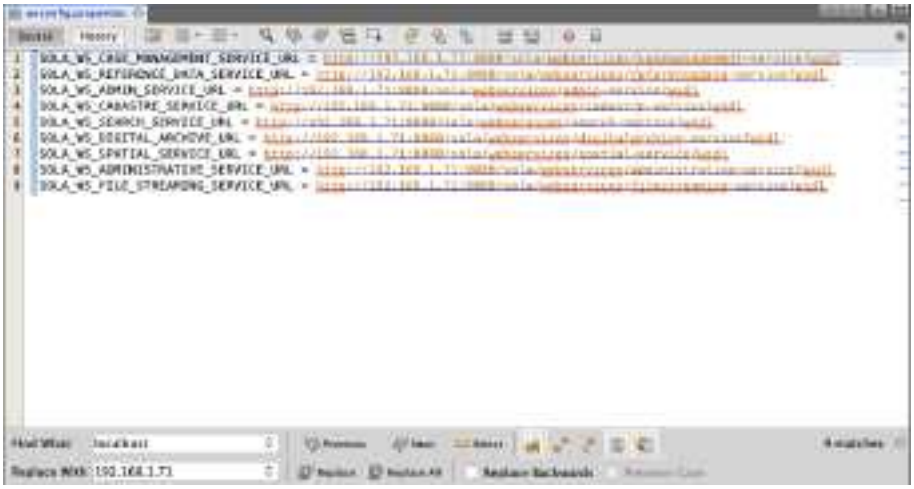


Figure 63 – Netbeans – replace "localhost" with server ip address in wsconfig.properties

- In Netbeans, select Clients Desktop project (in navigation panel), right mouse click and select Set Configuration - **Runnable**

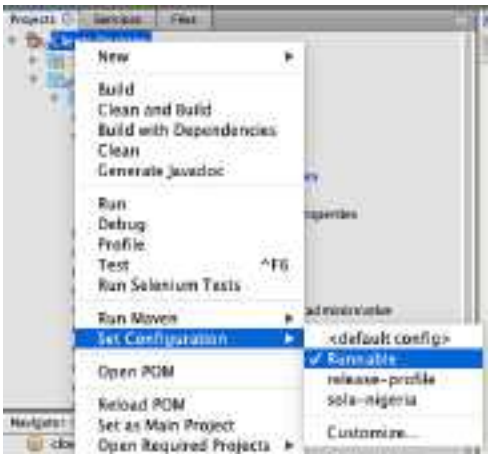


Figure 64 – Netbeans – Set configuration of Clients Desktop to Runnable

- In Netbeans, select Clients Desktop project (in navigation panel), right mouse click and select **Clean and Build**. The following screenshot of messages panel (bottom right hand side of the screen) indicates a successful build

Figure 65 – Netbeans – Successful Clean and Build

- Take a copy of the sola-desktop.jar and associated lib directory for copying and copy it into a C:\sola-registry-client folder ready for workstation client installations

Figure 66 – File Manager Client Desktop Installation Files

3. In the C:\sola-registry-client folder, create a **sola-registry.bat** file (in the same folder as the sola-desktop.jar file) with this one line instruction **`java -jar sola-desktop.jar`**
4. Select the **sola-registry.bat** file in File Manager, right click and create a **short-cut** to the sola-registry.bat file. Modify the shortcut file with an appropriate label and icon
5. For each client workstation, copy the sola-registry-client folder to the C:\ drive. Then copy the shortcut into the workstation's Desktop. To start the SOLA Desktop client software double click and enter user name and password (for installation use login credentials test/test)

7.6 Installation Troubleshooting

This section discusses some of the issues you may encounter when installing the SOLA Registry Desktop Web Start. This may include:

1. LAN connectivity
2. Unable to launch the application
3. Error: The application has requested a version of the JRE (version 1.8+)...
4. Microsoft Jscript compilation error
5. Download/launching of the SOLA Web Start Application hangs/freezes
6. Where is the Java Control Panel?
7. Application Blocked by Java Security (Oracle)
8. Passive FTP Mode is not supported by OpenWebStart
9. Could not read or parse JNLP file (OpenWebStart)

7.6.1 LAN connectivity

If you cannot display the SOLA Desktop initial landing page on a web browser (Figure 53) on the client workstation:

- Using a web browser and using the same url, try to display the landing page on the server
- If the landing page does not display on the server, use the procedure described in Section 7.5, to determine the IP address of the server and modify the landing page url accordingly
- If the landing page does display on the server, return to the client workstation and in a Command (Terminal) window type the `ping <<IP address of server>>` (as described in Section 7.5).
- If the **ping** is try turning off your firewall temporarily and try again on the client workstation to display the landing page using a web browser. If the landing page does display then you should ask your support specialist to reconfigure the firewall to allow server access from LAN connected workstations on ports 8080 and 8181.
- If the landing page still does not display on the client workstation, you should ask your support specialist to check the LAN cabling (or wifi configuration if that is how the client workstation connects to the LAN and server)
- You might also consider turning the client workstation and server anti-virus software off temporarily to identify what is blocking communications involving the SOLA software between the client workstations and the server. Once identified discuss configuration options or other antivirus software solutions with your support specialist

7.6.2 Unable to launch the application

Figure 67 – Application Error dialog

Click the Details button to get the More Information dialog. The relevant error message should be displayed at the top of the dialog. You can also view the Exception tab for more details.

Figure 68 - More Information dialog

7.6.3 Error: The application has requested a version of the JRE (version 1.8 +)...

Check the following to get the SOLA Desktop client software running using Java 8:

1. Confirm you have Java 8 installed as the default JVM on your computer
   - From Control Panel, click on Java and go to the Java Control Panel, click About... You should see a Java 8 splash displayed.

Figure 69 – Oracle Java splash display

   - If a different version of Java is displayed, install / reinstall Java 8.
2. Java version of your web browser
   - Although Java 8 is installed as the default JVM, unfortunately it does not update the Java version used by your web browser. If you left click the SOLA Registry Desktop Web Start link rather than right clicking and using Save link as... Java Web Start will be automatically triggered using Java configured for your web browser.

     To avoid this issue, right click the **SOLA Registry Desktop Web Start** link on the Welcome page and choose **Save As**... to save the sola-desktop.jnlp file to your local file system. Once saved, use Windows Explorer or equivalent to launch the application as described in the Installation section.
3. Desktop shortcut configuration

- If you get this error when launching the application using the desktop shortcut you will need to configure the shortcut. Refer to Configuring the Desktop Shortcut in the Installation section.

7.6.4 Microsoft JScript compilation error

This is a known issue with Java[11].

Figure 70 - JScript Compilation Error Message

To avoid this issue

1. Open the Java Control Panel
2. Choose Network Settings... in the Network Settings section of the General tab
3. Choose the Direct Connection radio button and click OK

Figure 71 – To set Java Direct Connection setting

7.6.5 Download/launching of the SOLA Registry Web Start Application hangs/freezes

If the download or launching of the SOLA Registry Web Start Application hangs / freezes, this may indicate that Java Web Start is being blocked by a firewall or anti-virus software. Try temporarily disabling your anti-virus and attempt to download the web start application again. This issue is known to occur with later versions of AVG Anti-virus (v9.0+).

7.6.6 Where is the (Oracle) Java Control Panel?

On Windows you should find “Java” in Windows Control Panel. You may need to change from Category View to an Icon View to make it easier to find. You should also be able to search for “Java Control Panel” using the Start Menu search.

7.6.7 Application Blocked by Java Security (Oracle Java)

If at a late stage of the installation, the installation stops with the following display open the **Java Control Panel**:

[11] http://bugs.sun.com/view_bug.do?bug_id=6780968

Figure 72 – Blocked by Java Security

1. In the **Security** tab, change Security to **High**

Figure 73 – Oracle Java Admin Console Security tab

2. Add http://ip-address-of-server:8080 and https://ip-address-of-server:8181 to the Exception Site List

7.6.8 Passive FTP Mode not supported by OpenWebStart

Figure 74 – OpenWebStart Passive FTP Mode message

No known solution.

7.6.9 Could not read or parse the JNLP file (OpenWebStart)

Figure 75 – OpenWebStart Could not read or parse JNLP file

Figure 76 – OpenWebStart Admin console JNLP file parse error details

No known solution.

7.7 Uninstall

To uninstall the SOLA Desktop or SOLA Admin from your computer

1. Launch the Java Control Panel
2. Choose View... in the Temporary Internet Files section of the General tab

Figure 77 - Java Control Panel

3. In the Java Cache Viewer, select the SOLA Desktop Web Start application or the SOLA Admin Web Start application and remove it using the Remove tool. This will remove your desktop shortcut to SOLA and the main jar for the application.

Figure 78 - Java Cache Viewer > Applications

4. To completely remove all references and jar's used by the SOLA Application you should also clear the Resources cache. To do this, choose Resources in the Show: drop down, select all of the resources listed and remove them using the Remove tool.

Figure 79 - Java Cache Viewer > Resources

8. Language Localization

8.1 Prerequisites

To add a new language to one of the SOLA Desktop applications you will ideally need to have familiarity with the SOLA Development environment[12].
Essential tools to install and files to download are:

- Netbeans IDE 8.2 from https://netbeans.org/downloads/8.2/
- Gitslave from https://sourceforge.net/projects/gitslave/files/
  UnTar the download file and copy resulting folder to C:\ (eg C:\gitslave-2.0.2)
  Add C:\gitslave-2.0.2 to the Environment variable Path (right click **This PC – Properties – Advance System Settings – Environment (button)**)
- Git (available for Windows, Mac & Linux/Ubuntu) from https://git-scm.com/download and accept all default options except :
  - select Icon on the Desktop
  - select Notepad++ as your editor for Windows installations (not VIM)
  - use Git from Git Bash only
  - untick Release notes
  - select Launch Git Bash
- The source code for the SOLA Desktop application to be localised. There are 10+ github repositories for each of the SOLA Desktop applications and so you :
  - Create the following directory structure on C:\drive
    - C:\SOLAsource\registry
    - C:\SOLAsource\sl
    - C:\SOLAsource\sr
  - Run Git Shell and navigate to the super repository (code) root folder by typing **cd C:\SOLAsource\registry** [or cd C:\SOLAsource\sl or cd C:\SOLAsource\sr]
  - Then type **git clone git://github.com/SOLA-FAO/code** to create the main repository for SOLA Registry. [or git clone git://github.com/SOLA-SL-FAO/code or git clone git://github.com/SOLA-SR-FAO/code]
  - Type cd code
  - Type ./**gits –v populate**. If you lose your connection during this action repeat the gits populate command until you are successful for all 10 git repositories.

8.2 Add new language into SOLA database language table

Currently the following languages are incorporated into some (but not all) of the SOLA software applications:

code	Display Value	Active	Default	Order	Left to Right
en-US	English	true	true	1	true
ru-RU	Русский	true	false	2	true
ar-JO	عربي	true	false	3	false
fr-FR	Français	true	false	4	true
es-ES	Español	true	false	5	true
pt-BR	Português	true	false	6	true
zh-CN	中国	true	false	7	true
km-KH	ខ្មែរ	false	false	8	true
sq-AL*	Shqip	false	false	9 (8)	true

[12] Refer to “SOLA Development Environment” - https://www.dropbox.com/s/sgqj6qx0arpo3wq/FAO%20FLOSS%20SOLA%20Development%20Environment.docx?dl=0

code	Display Value	Active	Default	Order	Left to Right
my-MM	မြန်မာ	false	false	10	true
am-ET*	አማርኛ	false	false	11 (5)	true

1. When a new language localization is required for a SOLA Desktop application, check the table above and see if this language has already been used in SOLA software.
2. Confirm this by launching SOLA Web Admin and clicking on the **Language** menu option
3. If you confirm that the new language is not yet used by SOLA, click on the Add link and completing the following fields:
    a. **code** – language and country code separated by dash symbol "**-**" [ie <ISO 639 2 character code for language>-<ISO 3166 2 character code for country> e.g. en-US for English language, USA country). You can find supported locales by the following link http://www.oracle.com/technetwork/java/javase/locales-137662.html.
    b. **display_value** – language name, which will be shown on the languages list for selection. Follow languages item order when entering localized language name.
    c. **active** – boolean value, indicating whether to show language in the selection list or not. Set True value.
    d. **is_default** – boolean value indicating whether the language is default or not. True value must be set only to one language. Set **False** value.
    e. **item_order** – integer value to define the order of localized strings in the reference data tables. Set **next available number**.
    f. **ltr** - If the language is not a "left-to-right" language (such as English) but is a "right-to-left" languages (such as Arabic), change the new language record ltr field to **false**

If the new language record is also the default language, the display value fields for each of the other language records must be reviewed and modified to reflect the default language.

Warning! Enter next sequential number in item order field for the new language. Do not change existing order. It will result into wrong localization of existing reference data values.

8.3 Add language flag icon

When the SOLA Desktop applications displays the languages list the relevant flag icon is also displayed together with language name. Select the flag icon for the new language from the collection of flag icons in the **Resources/Flags/** folder or you can source a png format map icon (16x16 size) from the internet.

1. Rename flag icon according with the language code (e.g. **en.png**). Make sure you use the **png** extension.
2. Copy the icon into Clients Desktop project, under Other Sources → src/main/resources → images.flags
3. Copy the icon into Clients SWING UI project, under Other Sources → src/main/resources → images.flags
4. In Netbeans, run the **Clean and Build** command for both projects.

Run **Clients Desktop** application to check the result.

1. In Netbeans, right click on the **Clients Desktop** project and select **Run** option.
2. On the main form, go to **View** → **Language** menu and check if new language is shown. Select the new language and agree to restart application.

8.4 Forms

8.4.1 Creating new Locale

Using Netbeans:

1) Open Clients Desktop project
2) Expand Other Sources → org.sola.clients.swing.desktop
3) Right click on the **Bundle.properties** file and select **Add → Locale…**
4) Select **Language code** and **Country code** (or search in predefined locales list)
5) Click **OK** button to generate new resource bundle for selected locale

Figure 80 – Netbeans New Locale form

8.4.2 Localize Forms

In Netbeans:

1 Select the Bundle.properties file for the new locale

Figure 81 – Netbeans table with Bundles translations

Translate all items in New Locale column of Bundle.properties

2 Right mouse click and select **Open**

3 The column containing all the form's labels for the new locale will initially be populated with the default language values. Translated values for each of the new locale labels need to be entered into this table.

4 Click **Save** button to save the changes

This process needs to be repeated for all forms in the Clients Desktop project. See the highlighted forms in the Clients Desktop project in the following screenshot:

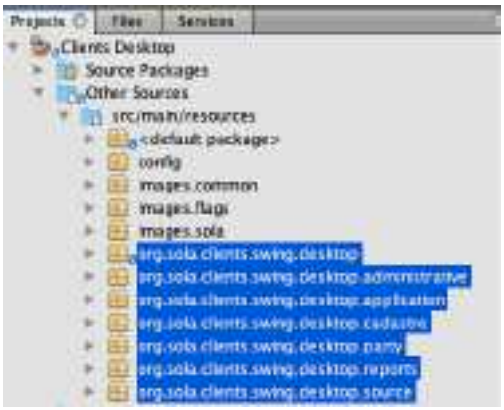


Figure 82 – Netbeans table – Resources to be localized

5 **Build** the Clients Desktop project
6 **Run** SOLA desktop client by right clicking on the **Clients Desktop** project and selecting **Run** option.
7 On the main form, go to **View** → **Language** menu and select the new language. Agree to restart application and check main form and dashboard items to see translation results.

8.4.3 Localize any date fields

Identify all forms with date format fields. For instance the Application search form.
In Netbeans:

1 Enter “ApplicationSearchPanel” into the Netbeans search field (top right hand corner) and select to open this form
2 Click on **Design** button
3 Check lodging date fields of the search criteria. They must show date format pattern, relevant to your locale. On the dashboard lodging dates must be also shown in the localized format, relevant to the country for the selected locale.

8.5 Messages

Keep in mind two simple rules, when doing messages localization:

1) Don’t localize or change keys with **.type** postfix (e.g. **clignrl002.type**). They always should stay the same as in default language.
2) Some messages may have parameters, they are defined with **{n}**, starting with zero (e.g. **There are {0} running tasks currently…**). Don’t remove them and place in accordance with semantic of sentence.

In Netbeans:

1) Open Common Messages project
2) Expand Other Sources → src/main/resources → org.sola.messages.client
3) Right click on the **Bundle.properties** file and select **Add** → **Locale…**
4) Select **Language code** and **Country code** (or search in predefined locales list)
5) Click **OK** button to generate new resource bundle for selected locale

To localize popup messages buttons and progress bar messages, follow these steps:

1) Open created bundle file by right click and selecting **Open** option
2) Find the keys with **dialog.options** prefix and translate their values in the new locale column
3) Find the key cliprgs001.message
4) Translate strings in the column of new locale finishing with **cliprgs041.message** (or the last message starting with **cliprgs** prefix**)**
5) Click **Save** button to save the changes
6) Right click on the **Common Messages** project and select **Clean and Build** context menu

Run the SOLA Desktop client by right clicking on the **Clients Desktop** project and selecting **Run** option. When you click **Login** button, open Dashboard panel or run any search. Check progress bar messages, located at the right bottom corner of the main form.

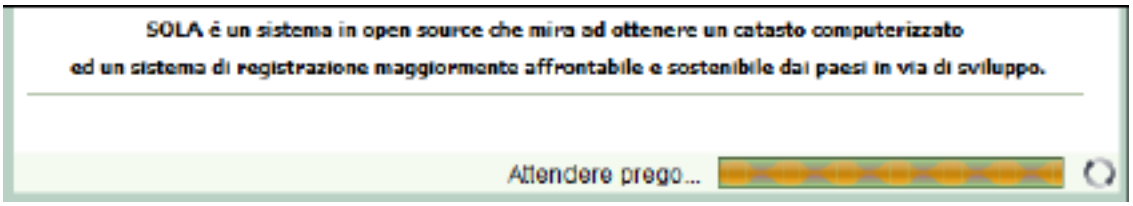


Figure 83 – Localized toolbar message (in Italian)

Repeat this process for all client messages in the Client Messages project and once you have saved the translated messages, Clean and Build and then Run the SOLA Desktop application to check how they will be displayed after localization.

8.6 Reports

As for the Client Messages project (in the previous section) you need to create a new locale for the **Clients Reports** project.

1) Open Clients Reports project
2) Expand Other Sources → src/main/resources → org.sola.clients.reports.locale
3) Right click on the ApplicationPrintingForm.properties file and select Add → Locale…
4) Select **Language code** and **Country code** (or search in predefined locales list)
5) Click **OK** button to generate new resource bundle for selected locale

To localize report, follow these steps:

1) Open created bundle file by right click and selecting **Open** option
2) Translate all values in the new locale column
3) Click **Save** button to save the changes
4) Right click on the **Clients Reports** project and select **Clean and Build** context menu
5) Run SOLA desktop client by right clicking on the **Clients Desktop** project and selecting **Run** option.
6) **Check** your localized reports by opening any available application from the **Dashboard** and click "**Print invoice**" button to generate invoice form which is based on Application printing form. If localization was successful you should see the localized application invoice.

Figure 84 – Example of Localized Invoice produced by SOLA (Italian)

8.7 Reference data tables

Launch **SOLA Web Admin** (http://localhost:8080/sola/admin) in your web browser:

1) Log in to the system

For each menu option available under the main **Reference data** menu option:

2) Open each available **Reference** table and click the "**Edit**" (Pen) icon at the left of each row
3) Add the translated value for the new language in the new language field
4) Click the **Save** button

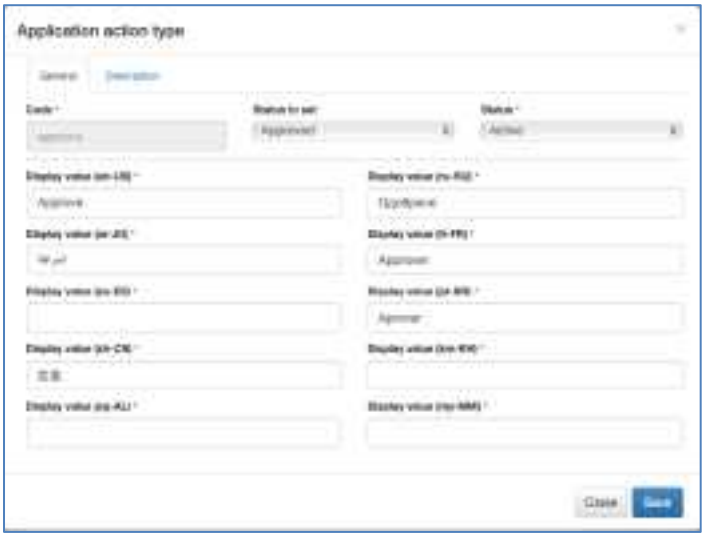


Figure 85 – Web Admin Application Action Type (note translations)

8.8 Business rules

Launch **SOLA Web Admin** (http://localhost:8080/sola/admin) in your web browser:

1) Log in to the system
   Click the main **Business rule** menu option and then click the **Search** button. For each listed business rule displayed:
2) Open each available **Reference** table and click the "**Edit**" (Pen) icon at the left of each row
3) Add the translated value for the new language in the new language field
4) Click the **Save** button

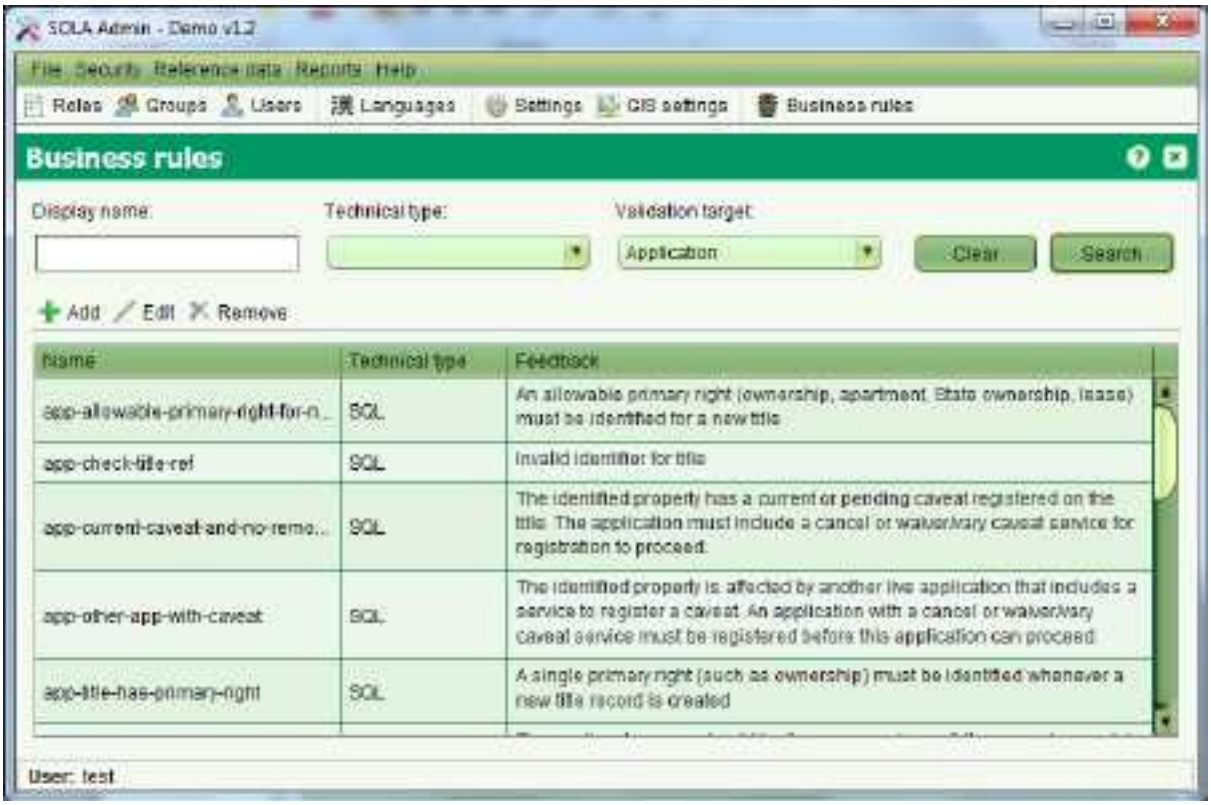


Figure 86 – Web Admin SOLA Business Rules

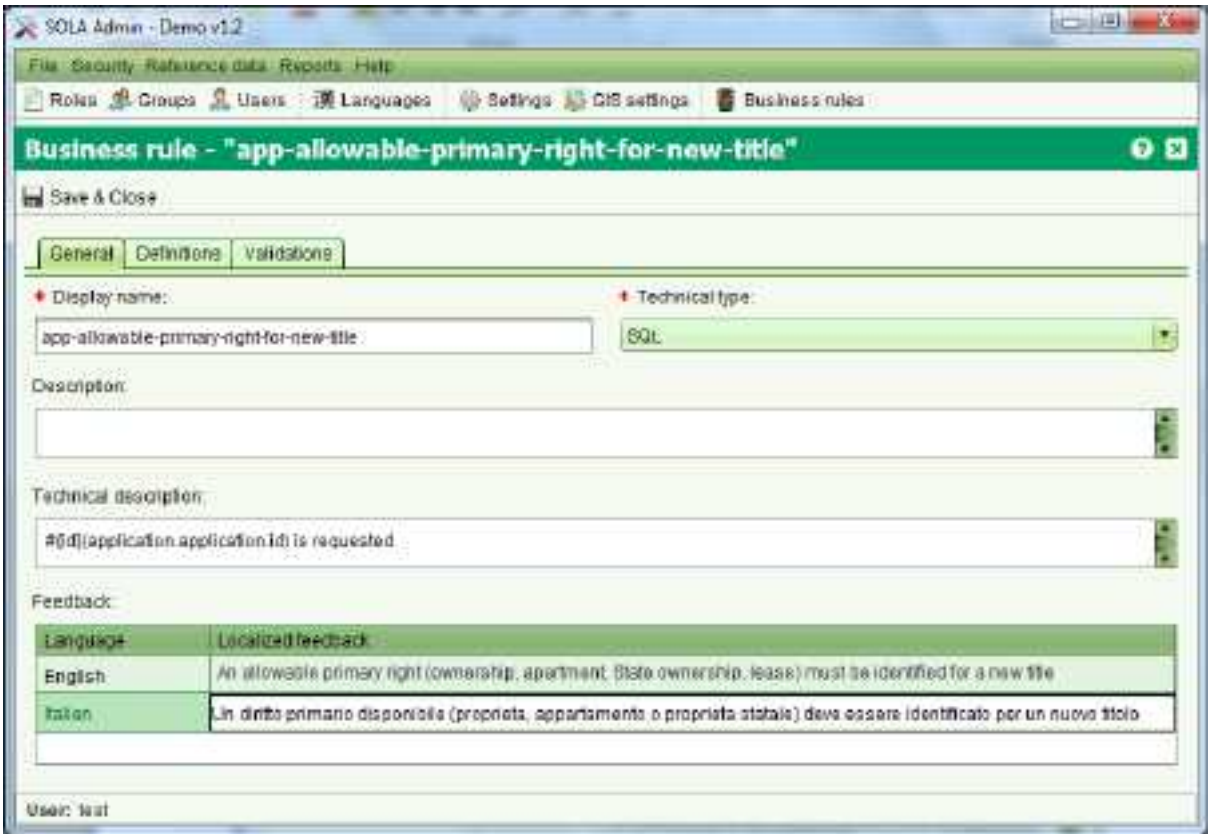


Figure 87 – Web Admin SOLA Business Rules -General Tab (with translations)

Once you have finished with BRs translation for application object, check results by running **SOLA Desktop** client. Select your language and open any existing application. Click **Validate** button on the main toolbar of application form. Feedback text in the validation list should be in selected language for BRs related to application object.

8.9 Help

In Netbeans:

1) Open ..code/Common **Help** project and expand **Other sources** → **src/main/resources** package.
2) Select src/main/resources and right click on New → Java Package
3) Enter **org.sola.common.help.*new_language_code*** as a package name, where ***new_language_code*** is a 2 characters (lower case) language code. Click **Finish** button.
4) Select the created package for the new language and right click and click on **New** → **Java Package.** Enter **MainTopics** name instead of **newpackage**. Click **Finish** button
5) Select the created package for the new language and right click and click on **New** → **Java Package.** Enter **howtotopics** name instead of **newpackage**. Click **Finish** button
6) Select the created package for the new language and right click and click on **New** → **Java Package.** Enter **images** name instead of **newpackage**. Click **Finish** button
7) Select the created package for the new language and right click and click on **New** → **Java Package.** Enter **javahelpsearch** name instead of **newpackage**. Click **Finish** button
8) In the Projects-Files-Services panel (top left hand side), select all files in the **org.sola.common.help.defaultlang** package and copy and paste files into the org.sola.common.*new_language_code* package created in step 3 above.

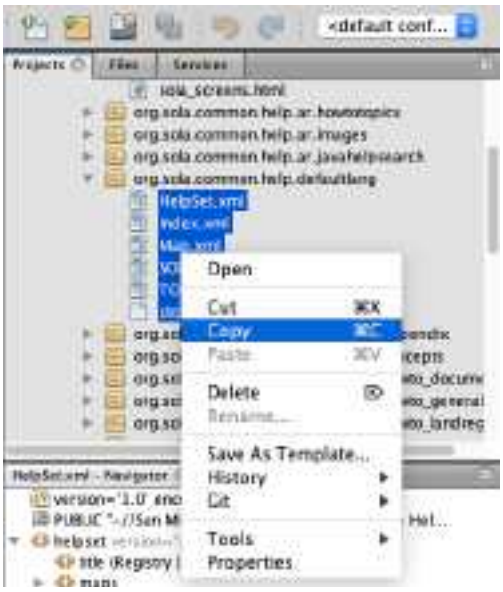


Figure 88 – Netbeans – Help Messages copying

Repeat for:
org.sola.common.help.defaultlang.MainTopics into package created in step 4
org.sola.common.help.defaultlang.howtotopics into package created in step 5
org.sola.common.help.defaultlang.images into package created in step 6 and
org.sola.common.help.defaultlang.javahelpsearch into package created in step 7

9) For the (new) org.sola.common.*new_language_code* package folder, Open **HelpSet.xml** file and translate just the string inside <title> tag. **Save**
10) For the (new) org.sola.common.*new_language_code* package folder, Open **TOC.xml** file and translate just the string inside the strings of **text=** property in the tags. **Save**
11) For the (new) org.sola.common.*new_language_code*.MainTopics sub-package folder, Open every html file and translate all general text [retain untranslated tags and style descriptions].
When you will translate html files, you may have problems with displaying non-Latin characters. Add the following tag between **<header>** tags, if it doesn't exist:
`<meta http-equiv="content-type" content="text/html; charset=UTF-8">`
12) For the org.sola.common.*new_language_code*.howtotopics sub-package, Open the **dashboard_and_main_menu.html** file and translate all general text [retain untranslated tags and style descriptions].
13) The **dashboard_and_main_menu.html** file has linkages to a number of images, described with **<img>** tag. Find all these images in the org.sola.common.*new_language_code***.images** package and replace them with screenshot images showing localized forms in the language. Try to keep the same image size as was generated for the org.sola.common.defaultlang.images package.
14) To check this localization, Clean&Build **Common Help** project and run **SOLA Desktop** client. Open Help from the main menu and review main topics as well as dashboard and main menu description.

8.10 SOLA Web Admin

The Web Admin source code is stored and downloadable from github (https://github.com/SOLA-FAO/admin) and contains the existing resource files that can be used as a starting point for a set of resource files for a new language localization.
Please note that this github repository needs to be separately downloaded from Github and is not included in git-slave process described at the beginning of this chapter.

In Netbeans:

1 Open the **SOLA Web Admin** Web project and navigate to Other Sources > src/main/resources > org.fao.sola.admin.web

2 Select the **errors.properties** file in the Project File-Services panel (top left corner of Netbeans window) right click **Add > Locale** and select the new language from the list of predefined locale in the Locale window. Click **OK**

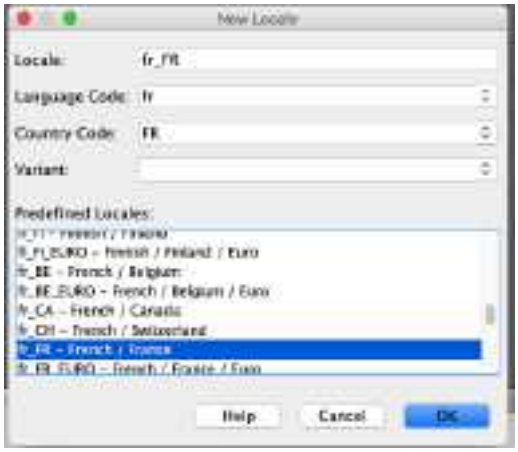


Figure 89 – Netbeans New Locale form

3 **Open** the *new language* **errors.properties** file and translate all the words and phrases in the column for the new language. **Save**

4 Select the **messages.properties** file in the Project File-Services panel (top left corner of Netbeans window) right click **Add > Locale** and select the new language from the list of predefined locale in the Locale window. Click **OK**

5 **Open** the *new language* **messages.properties** file and translate all the words and phrases in the column for the new language. **Save**

8.11 Re-compile SOLA Software

The final step of a new language localization of a SOLA Desktop application is to use Netbeans IDE to **Clean&Build** all the Netbeans projects you have altered in the localization process. The Clean & Build should be done in Netbeans and in this order:

1. Common Help project
2. Common Messaging project
3. Clients Swing Desktop project (right mouse click on project – Set configuration – Runnable. And after Clean & Build reset Set configuration to default)
4. Clients Reports project
5. Clients project
6. Code (Main) project
7. Admin (SOLA Web Admin) project

Collect the newly generated deployable files so that you can do a fresh installation and thoroughly test this new version that includes the new language.
These deployable files are:

- ..\code\clients\desktop-web-start\target**desktop-web-start.war**
- ..\code\services\services-ear\target**sola-services-ear.ear**
- ..\code\admin\target\ **sola-web-admin-1504a-SNAPSHOT.war**

9. Software Changes

Although the SOLA Desktop software applications are open source software it is strongly recommended that only experienced Java programmers attempt making any changes to the SOLA software. In the past FAO has also delivered a 5 day training program for potential software developers involved in the software customisation of SOLA Registry software based on the SOLA Development Environment Manual which remains a valuable resource although a little dated. Netbeans IDE 8.2 is the IDE used for the SOLA Desktop applications.

Source code for the SOLA Desktop applications is available from a series of Github repositories. The super github repositories for each of the SOLA Desktop applications are:

1. SOLA-FAO/code [SOLA Registry]
2. SOLA-SL-FAO/code [SOLA State Land]
3. SOLA-SR-FAO/code [SOLA Systematic]

9.1 Software Release Management Process

When you make software changes that need to be incorporated back into github repositories it is important that you follow this process:

1. Releases are identified by a 5 character code - YYMM<sequential alpha> eg first release in November 2019 would be 1911a, the second release 1911b
2. Check the Github super repository website of the SOLA Desktop application you have made changes to (eg https://github.com/SOLA-FAO/code) to see if there have been any other software releases in this month. Based on that, determine what your release code will be based on the rule provided above.
3. While at the SOLA Desktop super repository website, update the Issues log to note that the issue was resolved and reference the resolution of the issue with the new release code. Also take note of the IDs of resolved issues as recorded in the Issues Log
4. As part of the git “push” process (by way of a git or gits command or using the Git Desktop app) identify the release code and the issues resolved and/or a brief description of the changes (eg “Albanian language localization”).

9.2 Building the Web Start

9.2.1 Install the Web Start Application

1) Open the Glassfish Admin Console (localhost:4848)
2) From the Applications node, choose **Deploy**
3) Set the location to the **desktop-web-start.war**
4) Set the Context Root to **sola**
5) Click OK to deploy the WAR
6) From the Applications node, click the **Redeploy** hyperlink for the desktop-web-start application.
7) On the Redeploy page, click the **Java Web Start** checkbox option and click OK again. This option is not available during the initial deploy, hence it is necessary to redeploy the application.
8) From your web browser, browse to **localhost:8080/sola**. You will be taken to the SOLA Web Start landing page.
9) Follow the Installation instructions on that page to download and install one or both of the SOLA Client applications.

9.2.2 Uninstall the Web Start Application

These steps are for information only and are optional. The describe how to uninstall the web start application from your local PC.

1) To uninstall a web start application, open Windows Control Panel and select the Java option.

2) On the General tab, click the View button in the Temporary Internet Files section.
3) Select the web start application to uninstall and then click the red cross icon.

9.3 Building the Standalone Client

Java Web Start is one option for deploying the SOLA Client applications. They can also be deployed as Standalone client applications. The standalone versions require all dependent JARs to be packaged and available to the main client application JAR. This can be achieved using the maven-dependency-plugin and the maven-jar-plugin.

Maven includes the concept of Profiles to allow different build outputs to be generated dependent on the profile selected. A Runnable profile has been added to the Clients Desktop POM file to support building the standalone application. A copy of the profile is listed in the table below.

```
<profiles>
        <profile>
            <!-- This profile packages the SOLA Desktop dependencies
                 in a lib directory and sets the classpath in
                 sola-desktop jar. The output can be run independently
                 of the Netbeans environment. -->
            <id>Runnable</id>
            <build>
                <plugins>
                    <plugin>
                <!-- Copies all of the external sola-desktop dependencies
                     to the lib directory -->
                       <groupId>org.apache.maven.plugins</groupId>
                       <artifactId>maven-dependency-plugin</artifactId>
                       <version>2.3</version>
                       <executions>
                           <execution>
                               <id>copy-dependencies</id>
                               <phase>prepare-package</phase>
                               <goals>
                                   <goal>copy-dependencies</goal>
                               </goals>
                               <configuration>

<outputDirectory>${project.build.directory}/lib</outputDirectory>

<overWriteReleases>false</overWriteReleases>
<overWriteSnapshots>false</overWriteSnapshots>
<overWriteIfNewer>true</overWriteIfNewer>
<stripVersion>false</stripVersion>
<outputAbsoluteArtifactFilename>true</outputAbsoluteArtifactFilename>
                               </configuration>
                           </execution>
                       </executions>
                    </plugin>
                    <plugin>
                <!-- Configures the manifest for the sola-desktop.jar
                    including a classpath entry for all lib jars. -->
                       <groupId>org.apache.maven.plugins</groupId>
                       <artifactId>maven-jar-plugin</artifactId>
                       <version>2.3.1</version>
                       <configuration>
                           <archive>
                               <manifest>
                                   <addClasspath>true</addClasspath>
```

```
                                        <classpathPrefix>lib/</classpathPrefix>

<mainClass>org.sola.clients.swing.desktop.DesktopApplication</mainClass>
                                    </manifest>
                                </archive>
                            </configuration>
                        </plugin>
                    </plugins>
                </build>
            </profile>
        </profiles>
```

You can build the Runnable profile from Netbeans by

1) Open the **wsconfig.properties** file in the **Other Sources > src/main/resources > config** package of the Clients Desktop application.
2) Update each of the URL refrences to point to the location of the Glassfish Application Server. You can replace localhost with either the name of the computer hosting Glassfish or an IP address.
3) Right click the Clients Desktop project and choose **Set Configuration > Runnable**
4) Right click the Clients Desktop project again and choose **Clean and Build**
5) Maven will build the Clients Desktop application to its target directory and create a lib folder with all of the jar dependencies.
6) To deploy the standalone client, simply copy the sola-desktop.jar and lib folder to a folder location of your choice on each computer that will run the standalone client. Each computer will require the Java 7 JRE installed. You may also want to create a shortcut link to the sola-desktop.jar and place that link on the users desktop and/or their start menu to make it easier for them to start the application.

To unset the Runnable profile

7) Right click the Clients Desktop project and choose **Set Configuration > <default config>**
8) Revert the wsconfig.properites file so all URL references point to localhost again.

9.4 Guidelines for Production Releases

When createing a webstart or standalone release for production use, you should ensure the the on-line Help file is up to date and also update the version number on all SOLA projects to a final version number. E.g. if the development version number is 1.1-SNAPSHOT, update the version number to 1.1. Every SOLA project has its version number recorded in its POM file. To update version numbers, use the Find and Replace of Netbeans (e.g. find all `<version>1.1-SNAPSHOT</version>` and replace them with `<version>1.1</version>`). Note that some version references may be located in the dependencies section of the POM. You should also update the AboutFrom to indicate the new version of the application. Once the version numbers are updated, you will need to Clean and Build Main POM before creating the Release package. After creating the release package, update the version numbers a second time by incrementing the minor version number and adding –SNAPSHOT to indicate the build is a development build.

9.5 Guidelines for Test Releases

There may be occasions where you want to deploy a webstart build for testing purposes. In this case you still need to update version number for every SOLA project. The reason is the **webstart-jnlp-servlet** uses the version number of the jar to determine whether or not to serve a new copy of the jar to the client. If the version number is not changed, and the user already has a copy of the jar with the same version number, the servlet will not provide a

new copy of the jar even if the contents of the jar have changed. As above, use the Find and Replace of Netbeans to update the version number. The recommened change is to set the SNAPSHOT text to the current date using the YYYYMMDD format (e.g. find all `<version>1.1-SNAPSHOT</version>` and replace them with `<version>1.1-20120601</version>`).

Once the version numbers are updated, you will need to Clean and Build Main POM before creating the Test Release package. After creating the Test Release package, revert the version numbers to use the –SNAPSHOT suffix.

9.6 Hardening Glassfish

For a production deployment it is important to ensure that Glassfish is adequately hardened to reduce the potential avenues an attacker could use to compromise the system. This page references some basic hardening tasks that should be performed on Glassfish. Note that Glassfish is just one piece of the security puzzle. For a secure system, a Threat Assessment should be undertaken to identify the most likely areas of attack and the appropriate counter measures put in place. Areas of security risk will likely encompass the network, hardware and operating systems, backup procedures as well as non-technical aspects such as physical server access control.

Things you should do

1. Run Glassfish under a non root (admin) user account. If you have installed Glassfish as root, you can change the owner to a new account by stopping Glassfish and granting the new user account ownership rights to the Glassfish3 and all child folders. On Linux it will also be necessary to ensure all HTTP port listeners are configured for ports above 1024 (e.g. 8080 instead of 80 and 8181 instead of 443).
2. Change the admin password for the Glassfish Admin Console to ensure this is a strong password (min 12 characters with letters, numbers and symbols)
3. Secure the Administration Console by adding a password (current logon credentials are admin/<blank>)
   ```
   $ cd /payara5/bin
   $ sudo ./asadmin change-admin-password
   $ sudo enable-secure-admin
   $ sudo ./asadmin restart-domain domain1
   ```
4. Update Glassfish to use the Server JVM instead of the Client JVM with appropriate memory settings
5. Clear the product name setting
6. Obfuscate Glassfish Powered By details.
7. Use Password Aliasing to encrypt the database user and password used to connect to the SOLA Database
8. Update the Glassfish Keystores

References that explain the items above in more detail include;

Securing your Glassfish Hardening Guide
Java Key and Certificate Management Tool

9.6.1 Updating the Glassfish Keystores

Another very important element of hardening Glassfish and the SOLA application is to update the keystores used to secure the SOLA web service communications. Glassfish has

two password encrypted keystores (keystore.jks and cacerts.jks) to manage the digital certificates it uses for SSL, encryption, authentication and related security services. The source code for SOLA includes the default Glassfish keystores updated with the Development Default certificates used by WSIT and Metro. These keystores along with the passwords used to encrypt them are public information and should be treated as compromised. **They are not suitable for production deployments of SOLA**. The following sections describe how to update the keystores with unique certificates and change the passwords protecting the keystores. Note that care should be taken to ensure the updated keystores and passwords generated by this process do not become public information.

Modifying the SOLA Source and Build

The first step is to update the SOLA source code to make it easier to hide private details such as the updated keystores and the keystore passwords.

NOTE: *The changes described in this section were made to the SOLA configuration in 2013. They are listed here for reference only. If you are creating new WIST digital certificates, proceed to the Change the Glassfish Master Password section below.*

- Add ***.jks** to the .../code/services/boundary/.gitignore file to ensure Git does not push the updated keystores to a public repository. There is no need to remove the existing keystore.jks and cacerts.jks - they are already compromised.
- Add the following resource filtering section to the pom.xml of the Boundary Web Services project

```
<resources>
<!-- Only filter the WSIT XML files. This will ensure the binary jks
 file is not corrupted by the filtering process. -->
   <resource>
      <!-- Sets filtering to true on the WSIT XML files -->
      <directory>src/main/java/META-INF</directory>
      <filtering>true</filtering>
      <includes>
         <include>**/*.xml</include>
      </includes>
   </resource>
   <resource>
      <!-- Sets filtering to false on all other files -->
      <directory>src/main/java/META-INF</directory>
      <filtering>false</filtering>
      <excludes>
         <exclude>**/*.xml</exclude>
      </excludes>
   </resource>
</resources>
```

- Add or update the properties section in the pom.xml of the Boundary Web Services project with the following. These properties relate to the default keystores but they can be overridden using the Maven settings.xml file.

```
<properties>
   <wsit.keystore.storepass>changeit</wsit.keystore.storepass>
   <wsit.keystore.peeralias>xws-security-
server</wsit.keystore.peeralias>
   <wsit.keystore.location>keystore.jks</wsit.keystore.location>
</properties>
```

- Replace the sc:Keystore element in each WSIT configuration XML file in the META-INF source code package of the Boundary Web Services project to read as follows. Note the FileStreaming file does not require an update.

```
<sc:KeyStore wspp:visibility="private" location="${wsit.keystore.location}"
type="JKS" storepass="${wsit.keystore.storepass}"
alias="${wsit.keystore.peeralias}"/>
```

- Add the following resource filtering section to the pom.xml of the Boundary Web Services Clients project

```
   <resources>
   <!-- Only filter the WSIT XML files. This will ensure the binary jks
    file is not corrupted by the filtering process. -->
      <resource>
         <!-- Sets filtering to true on the WSIT XML files -->
         <directory>src/main/resources</directory>
         <filtering>true</filtering>
         <includes>
            <include>**/*.xml</include>
         </includes>
      </resource>
      <resource>
         <!-- Sets filtering to false on all other files -->
         <directory>src/main/java/META-INF</directory>
         <filtering>false</filtering>
         <excludes>
            <exclude>**/*.xml</exclude>
         </excludes>
      </resource>
   </resources>
```

- Add or update the properties section in the pom.xml of the Boundary Web Service Clients project with the following.

```
   <properties>
      <wsit.truststore.storepass>changeit</wsit.truststore.storepass>
      <wsit.truststore.peeralias>xws-security-
server</wsit.truststore.peeralias>
      <wsit.truststore.location>cacerts.jks</wsit.truststore.location>
   </properties>
```

- Replace the sc:TrustStore element in each WSIT configuration XML file in the META-INF resources package of the Boundary Web Service Clients project to read as follows.

```
  <sc:TrustStore wspp:visibility="private" type="JKS"
storepass="${wsit.truststore.storepass}"
peeralias="${wsit.truststore.peeralias}"
location="${wsit.truststore.location}"/>
```

9.6.1.1 Change the Glassfish Master Password

The Glassfish Master Password is used to decrypt the Glassfish keystores allowing Glassfish to access the digital certificates they contain. By default, this password is *changeit* and you should do exactly that. Be aware that if you complete this process on your development instance of Glassfish so that you can create and test the build package for

deployment, you will also need to synchronize the Glassfish Master Password on the Glassfish production instance to match the password used to generate the updated keystores.

To update the Glassfish Master Password, stop your instance of Glassfish and use the change-master-password subcommand of asadmin from a command prompt. e.g.

```
<Glassfish Install Dir>\bin\asadmin change-master-password --
savemasterpassword=true --domaindir <SOLA Glassfish Domain Dir>
```

You may need to run the change-master-password command as an Administrator. You should also explicitly set the location of the domain (domaindir) to ensure the change-master-password updates the passwords on the default Glassfish keystores.

9.6.1.2 Configure your Maven settings.xml

settings.xml is used by Maven to provide override property values and User Settings (located in the users .m2 folder) are often used to limit exposure of passwords and other secure information. Add or update the settings.xml file in your .m2 folder (will be in your Windows user profile or you Linux home directory) to be similar to the following. Replace YourMasterPassword with the Glassfish Master Password set in the following section.

```
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
http://maven.apache.org/xsd/settings-1.0.0.xsd">
   <profiles>
      <profile>
         <id>prod</id>
         <activation>
            <activeByDefault>false</activeByDefault>
         </activation>
         <properties>
           <wsit.truststore.storepass><!--YourMasterPassword--
></wsit.truststore.storepass>
           <wsit.truststore.peeralias>sola-wsit-
cert</wsit.truststore.peeralias>
           <wsit.truststore.location>sola-
truststore.jks</wsit.truststore.location>
           <wsit.keystore.storepass><!--YourMasterPassword--
></wsit.keystore.storepass>
           <wsit.keystore.peeralias>sola-wsit-
cert</wsit.keystore.peeralias>
           <wsit.keystore.location>sola-
keystore.jks</wsit.keystore.location>
        </properties>
     </profile>
  </profiles>
</settings>
```

9.6.1.3 Generate new Certificates

To generate new certificates for Glassfish requires using the Java Keytool. The following Windows CMD script can be used to update the default Glassfish keystores and generate two new keystores suitable for SOLA (sola-keystore.jks and sola-truststore.jks). Simply update the **keytool** and **glassfish_config_dir** variables to match your locations. If the script

fails with an error indicating the password for the keystores is incorrect, make sure you correctly set the domaindir in the change-master-password command above. E.g. If the domain config dir is C:\Work\source\glassfish\sola\config\, the domain dir will be C:\Work\source\glassfish

```
@echo off
REM 27 FEB 2013
REM Windows CMD script that generates new self-signed certificates using
the
REM Java 7 Keytool and configures the default keystores in Glassfish so
REM the WSIT certificate can be used for SOLA Web Services authentication.
REM
REM This script should be used to generate a new WSIT certificate for
REM production deployments of SOLA. The keystores and certificates
REM packaged with the generic SOLA are publicly available and should
REM be treated as compromised. THEY ARE NOT SUITABLE FOR PRODUCTION
REM DEPLOYMENTS OF SOLA. Care should be taken to ensure the keystores
REM and passwords generated for a production deployment do not become
REM public information.
REM
REM This script relies on the Glassfish Master Password. The default master
REM password for Glassfish is changeit. You should change this password
REM on the production instance of Glassfish using the Glassfish asadmin
REM change-master-password sub-command.
REM
REM When running this script on a developer instance of Glassfish,
REM ensure the master password matches the Production master password
REM otherwise the keystores created will not be accessible by the
REM Production Glassfish instance.

REM Location of the Java Keytool. Note that the Java 7(+) Keytool must be
REM used otherwise the certificate generated will not be suitable for use
REM by Metro/WSIT. Update as required.
set keytool="C:\Program Files\Java\jre7\bin\keytool"
REM Location of the SOLA domain config folder. Update as required.
set glassfish_config_dir=C:\Work\source\glassfish\sola\config\
set master_password=?

set /p master_password= Enter your Glassfish Master Password
[%master_password%] :

echo
echo Updating the Glassfish keystores. Please wait...
echo

REM Remove the default s1as and glassfish-instance certificates from the
REM Glassfish keystore.jks and the cacerts.jks keystore. As with the
default
REM SOLA WSIT certs, these certs are public information and should be
REM considered compromised for any production deployment.
%keytool% -delete -alias s1as -keystore %glassfish_config_dir%keystore.jks
-storepass %master_password%
%keytool% -delete -alias glassfish-instance -keystore
%glassfish_config_dir%keystore.jks -storepass %master_password%
%keytool% -delete -alias sola-wsit-cert -keystore
%glassfish_config_dir%keystore.jks -storepass %master_password%
%keytool% -delete -alias s1as -keystore %glassfish_config_dir%cacerts.jks -
storepass %master_password%
%keytool% -delete -alias glassfish-instance -keystore
%glassfish_config_dir%cacerts.jks -storepass %master_password%
```

```
REM Generate three new certificates in the default Glassfish keystore. Two
REM that replace the s1as and glassfish-instance certificates and a new one
REM to be used by SOLA WSIT authentication. Note that only one X509v3
REM certificate is required for Username Authentication with Symmetric Key
-
REM the default WSIT authentication mechanism used by SOLA. If you are
using
REM a different authentication mechanism such as Mutual Authentication, you
REM may need to generate more certificates.
REM
REM Notes:
REM 1) The certificates generated are valid for 10 years from the day of
REM    generation
REM 2) The keypass and storepass must match the Glassfish Master Password
REM    otherwise Glassfish will not be able to load the certificates from
REM    the keystore
REM
%keytool% -genkeypair -keysize 2048 -alias sola-wsit-cert -keyalg RSA -
validity 3650 -dname "CN=sola-wsit-cert, O=SOLA" -keypass %master_password%
-storepass %master_password% -keystore %glassfish_config_dir%keystore.jks
%keytool% -genkeypair -keysize 2048 -alias s1as -keyalg RSA -validity 3650
-dname "CN=s1as, O=SOLA" -keypass %master_password% -storepass
%master_password% -keystore %glassfish_config_dir%keystore.jks
%keytool% -genkeypair -keysize 2048 -alias glassfish-instance -keyalg RSA -
validity 3650 -dname "CN=glassfish-instance, O=SOLA" -keypass
%master_password% -storepass %master_password% -keystore
%glassfish_config_dir%keystore.jks


REM Export the three new cerificates so they can be loaded into other
REM keystores.
%keytool% -exportcert -alias sola-wsit-cert -keystore
%glassfish_config_dir%keystore.jks -storepass %master_password% -rfc -file
%glassfish_config_dir%sola-wsit-cert.cer
%keytool% -exportcert -alias s1as -keystore
%glassfish_config_dir%keystore.jks -storepass %master_password% -rfc -file
%glassfish_config_dir%s1as.cer
%keytool% -exportcert -alias glassfish-instance -keystore
%glassfish_config_dir%keystore.jks -storepass %master_password% -rfc -file
%glassfish_config_dir%glassfish-instance.cer


REM Tidy up any sola keystores created from a previous run of this script.
DEL /Q  %glassfish_config_dir%sola-truststore.jks
DEL /Q  %glassfish_config_dir%sola-keystore.jks


REM Import the new certificates into the default Glassfish cacerts.jks
REM keystore and create a new keystore (sola-truststore.jks) to hold the
REM public key for sola-wsit-cert
echo
echo Enter Y to confirm the import of the certificates into the keystores
echo
%keytool% -importcert -alias sola-wsit-cert -keystore
%glassfish_config_dir%sola-truststore.jks -storepass %master_password% -
file %glassfish_config_dir%sola-wsit-cert.cer
%keytool% -importcert -alias s1as -keystore
%glassfish_config_dir%cacerts.jks -storepass %master_password%  -file
%glassfish_config_dir%s1as.cer
```

```
%keytool% -importcert -alias glassfish-instance -keystore
%glassfish_config_dir%cacerts.jks -storepass %master_password% -file
%glassfish_config_dir%glassfish-instance.cer


REM Delete the certificate files - they are no longer required.
DEL /Q %glassfish_config_dir%sola-wsit-cert.cer
DEL /Q %glassfish_config_dir%s1as.cer
DEL /Q %glassfish_config_dir%glassfish-instance.cer


REM Create a copy of the Glassfish keystore.jks and remove any certs that
REM are not required. The sola-keystore.jks will be used by the SOLA Web
REM Services and does not need the additional certificate information.
COPY /Y /B %glassfish_config_dir%keystore.jks %glassfish_config_dir%sola-
keystore.jks
%keytool% -delete -alias s1as -keystore %glassfish_config_dir%sola-
keystore.jks -storepass %master_password%
%keytool% -delete -alias glassfish-instance -keystore
%glassfish_config_dir%sola-keystore.jks -storepass %master_password%
%keytool% -delete -alias wssip -keystore %glassfish_config_dir%sola-
keystore.jks -storepass %master_password%
%keytool% -delete -alias xws-security-client -keystore
%glassfish_config_dir%sola-keystore.jks -storepass %master_password%
%keytool% -delete -alias xws-security-server -keystore
%glassfish_config_dir%sola-keystore.jks -storepass %master_password%


echo
echo All Done!
echo
pause
```

9.6.2 Build and Deploy with the Updated Keystores

You will now have 4 keystores in the Glassfish config directory - keystore.jks, cacerts.jks, sola-keystore.jks and sola-truststore.jks

- Copy the sola-keystore.jks to the Boundary Web Services META-INF source code package
- Copy the sola-truststore.jks to the Boundary Web Service Clients META-INFO resource package
- Build the Main POM using the prod configuration setup in settings.xml. You may need to include some extra build parameters for this configuration such as skip tests and Maven java memory settings. See the default build configuration for details.
- Deploy and test the deployment packages as per usual.

When you are ready to deploy the updated build to the Production Glassfish instance

- Stop the production instance
- Update the Glassfish Master Password on the production instance to match the password used to encrypt the keystores
- Copy the updated keystore.jks and cacerts.jks to the Domain config directory on the Production Glassfish instance. Be careful to ensure these keystores are not exposed more than is absolutely necessary. Also be sure to use the correct config directory - the keystore.jks and cacerts.jks should replace files that already exist. If they do not exist its because you're in the Glassfish config directory instead of the Domain config directory.

- Deploy the new SOLA Services EAR and Web Start WAR

If you want to return to using the development keystores, revert the Glassfish Master password on your developer glassfish instance to *changeit* and rebuild SOLA using the default build configuration. Be aware that once the Production Glassfish instance has been updated, you will need to use the prod build configuration whenever you create a build for production. You also need to ensure the sola-keystore.jks and sola-truststore.jks are in the appropriate directories as they will be excluded from source control.

10. Troubleshooting

10.1 SOLA Client Software Logging

This section describes the two logs available with the SOLA client software for the SOLA Desktop applications; the SOLA Application Log and Java Runtime Environment (JRE) logging. It also describes how to access information from those logs.

10.1.1 SOLA Application Log

The SOLA Web Start application (eg. SOLA Registry Desktop Client software) creates an application log file on your local computer to capture exceptions and various status messages. You can find those log files at the following directory location **<User Home Directory>/sola/logs**.

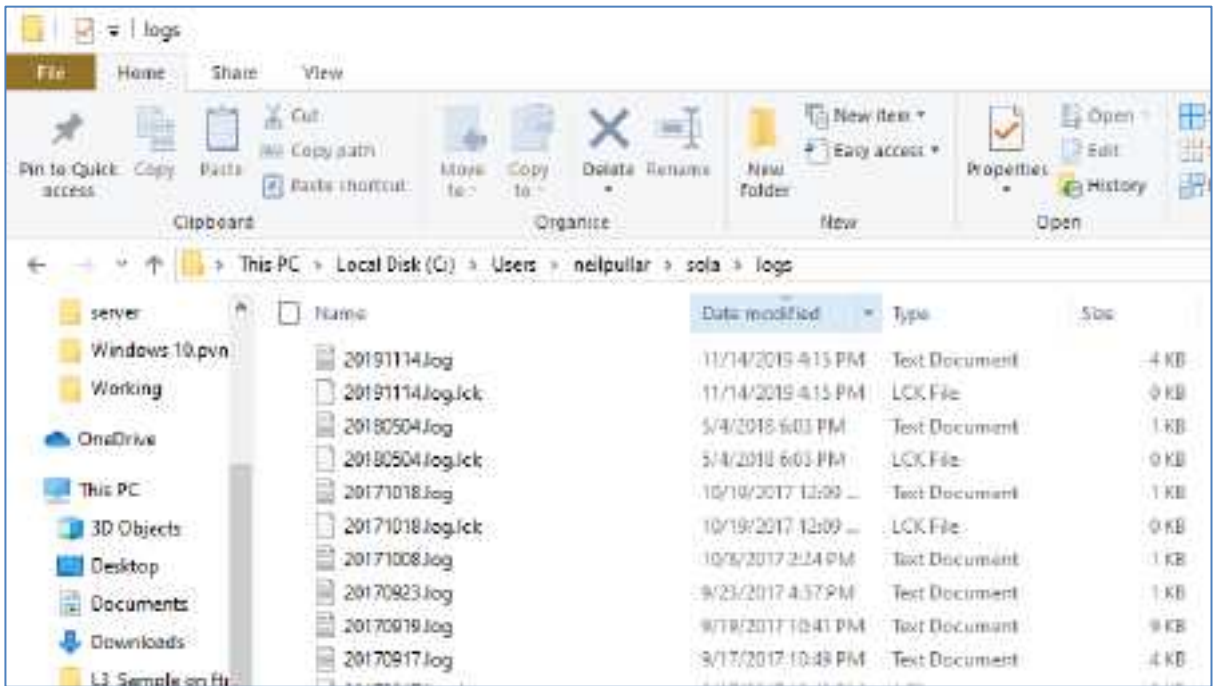


Figure 90 – Location of SOLA Application Log in Windows 10

The name of the log file will be DesktopApplication_<date>.log (for SOLA State Land Desktop). Simply open the appropriate log and copy the text from the log file into your email reply or issue ticket.

Note that your <User Home Directory> will depend on the operating system of your computer. For example, under Windows 10 this will be C:\Users\<Your Windows User Name>.

10.1.2 Java Runtime Environment (JRE) Logging

SOLA is a Java application and the Java Runtime Environment (JRE) also provides a detailed logging capability. The advantage of using JRE Logging is that every exception and/or status message is reported whereas the SOLA Application Logs only capture SOLA specific exceptions and messages and may not necessarily capture the true cause of an issue.

JRE Logging is **not** turned on by default. You need to use the Java Control Panel to turn on the Java Console which can then be used to turn on JRE Logging as required. To turn on the Java Console and JRE Logging

1. From the Control Panel, choose the **Java** icon.
2. On the Java Control Panel, click the Advanced tab and expand the **Java console** node to select the **Show console** option.
3. Click the OK button to close the Java Control Panel and save changes.

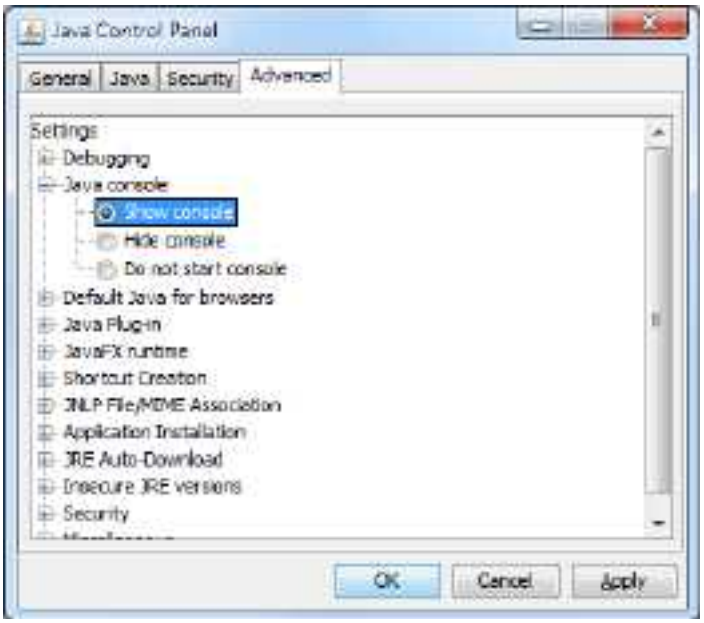


Figure 91 - Java Control Panel Advanced tab

4. Start the SOLA Web Start application by double clicking the appropriate desktop shortcut. You will notice the Java Console display along with the SOLA login page.
5. Click the Java Console so that it has focus and press **5**. The message **Trace level set to 5: all ...completed** should display in the console. Pressing 5 turns on logging to its maximum level.

Figure 92 - Java Console with logging turned on

6. As you use the SOLA application, messages will be output to the console. Repeat the actions that cause the issue to occur.
7. After successfully repeating the issue, click the **Copy** button on the Java Console and paste the contents of the JRE log into your email reply or issue ticket.

Be aware that JRE Logging will only be turned on for the duration of your current session. If you restart the SOLA Web Start application, JRE Logging will be turned off until you set the logging level in the Java Console.

10.2 How to describe a software issue

When a problem is encountered it is important to record the following:

1. Any error message details
2. What form, record and field were being accessed and what was the user action immediately before the error message displayed or software problem became evident
3. Was the error able to be repeated ?
4. What is the version of the software (note header on Home page)

The other source of information is the Payara (raw) log which is viewable from the Payara Admin console. Under the Common Tasks panel (left side of screen), click on **server** and then within the General Information Screen, click on the **View Raw Log** button. Scroll to the bottom (most recent log entries) and then scroll up until you find reference to the software incident. These log entries will often provide the reason for the software issue. A copy of an extract of the raw log file from before the incident when the system was operating OK until after the incident with all the log entries resulting from the incident should be made – highlight those entries, CTRL-C and paste into a text editor like Notepad++

You should send these items to your technical support. Backups of the SOLA database ideally before and after the incident should also be sent if feasible.

Should the technical support decide that the software incident is actually a software fault, they should log these details and their conclusions as an issue on Github against the appropriate SOLA Desktop application /code repository. These are:

- SOLA Registry – https://github.com/SOLA-FAO/code
- SOLA State Land - https://github.com/SOLA-SL-FAO/code
- SOLA Systematic - https://github.com/SOLA-SR-FAO/code

10.3 Payara Server Deployment Problems

If you experience a Payara Deployment Problems that are not solved by restarting the domain1 Payara domain (either through the Payara Admin console or by the command line .**asadmin stop-domain domain1** and then .**asadmin start-domain domain1** – run from the payara5/bin folder), then stop the domain1 Payara domain and delete the **generated** sub-folder (..payara5/glassfish/domains/domain1/**generated**). Then restart this Payara domain

10.4 Payara Server starts but the SOLA server does not appear run

When you type in http://localhost:8080/sola/admin into your web browser and you do not get the SOLA Web Admin login page then you should follow the following steps:

1. Open Payara Server Admin console (http://localhost:4848) and click on **Applications** in the left hand panel
2. Open the Applications, and click on the **Re-deploy** link all 3 applications

Figure 93 – Payara Server Admin Console – Redeploy Application

3. Under **Server** (near the top of the page) click on the **Restart** button

11. SOLA Desktop Database Data Dictionaries

Version: Release 1503a

Each of the 3 SOLA Desktop applications has a distinct database schema although they have many common elements including the schema structure.

The SOLA Desktop databases have been developed using the PostgreSQL open source database and is an implementation of an extended version of the Land Administration Domain Model (LADM). LADM is published as ISO 19152 by the International Organization for Standardization (ISO). In the design of the SOLA databases, it has been necessary to extend ISO 19152 because of the operational needs of land administration agencies to incorporate case management and other features into any system that supports the processing of client service requests (for land information, registration and cadastre change requests and others) and the maintaining and updating of the record of rights and restrictions, ownership and property boundaries.

The SOLA Data Dictionaries are a series of linked html pages. There is an index for each of the SOLA database and then html pages organized according to the the SOLA Desktop database schemas. To navigate the data dictionaries, follow the hyperlinks available on each page. Each schema section includes a list of all tables within the schema along with table descriptions, column list, constraints and relationship details. Database views and functions are also described.

Tags are used to summarise common table features. The tags used in this dictionary are

- Business Rules - The table provides configuration or implementation details for business rules.
- Change History - All changes to the table are tracked in a duplicate _historic table for audit and recovery purposes.
- FLOSS SOLA Extension - The table or column is an extension to the LADM standard.
- LADM Reference Object - The table is an implementation of a object from the LADM standard.
- Map Configuration - The table provides configuration details for the Map Viewer.
- Not Used - The table has been included in the Community Server database schema as it represents an object from the LADM, however it is not used by the Community Server application.
- Reference Table - The table contains code values used in other Community Server tables.
- System Configuration - The table provides Community Server system configuration details.
- User Admin - The table is part of the user management system in Community Server.

12. gitslave routine to download all 10 SOLA repositories

This routine assumes the setup is on a Windows workstation.

1. Create an Account on GitHub (first time use only)
   Go to www.github.com and click on the **Signup and Pricing** tab and then click on **Create a Free Account** button. Enter user name, email and password (this is NOT your github passphrase) and click on the **Create an Account** button.
2. Setup **git** (first time use only)
   - Download Git for Windows from https://git-scm.com/download/win
   - Install by **Run as Administrator** downloaded installation file changing default editor to Notepad++ and selecting the "Use Git from Git Bash only" option
   - Perform all of these steps to setup up SSH keys and your details
     - from https://help.github.com/articles/generating-ssh-keys
     i. Run Git Bash **cd ~/.ssh** ("No such file or directory" should display)
     ii. Use the email registered with www.github.com
     ssh-keygen -t rsa -C your_email@youremail.com
     iii. Enter file in which to save the key
     (/c/Users/*yourWindowsUserName*.ssh/id_rsa): ***[Press enter]***
     iv. Enter passphrase (empty for no passphrase): ***[Type a passphrase]***
     v. Enter same passphrase again: ***[Type passphrase again]***
     vi. Run the following code to copy the key to your clipboard.
     clip < ~/.ssh/id_rsa.pub
     vii. Log onto www.github.com logon and click on the icon for your Account Settings
     viii. Click **"SSH Keys"** in the left sidebar
     ix. Click "**Add SSH key**" button
     x. Paste your key into the "**Key**" field
     xi. Click "**Add key**"
     xii. Confirm the action by entering your GitHub password
     xiii. Back to GitBash type in **ssh -T git@github.com**
     xiv. Verify that the fingerprint matches the one generated and copied into your githube ssh Keys and type "**yes**"
     xv. If that username is correct, you've successfully set up your SSH key. Do not worry about the shell access comment.
3. Setup **gitslave** (first time use only)
   - Download gitslave from https://sourceforge.net/projects/gitslave/files/
   - UnTar the download file and copy resulting folder to C:\ (eg C:\gitslave-2.0.2)
   - Add C:\gitslave-2.0.2 to the Environment variable **Path**

   Right click **This PC – Properties – Advance System Settings – Environment**
4. Complete the following **git and gitslave** processes:
   - Create the following directory structure on C:\drive
     - C:\SOLAsource\OpenTenure
   - Run Git Bash
     - enter your github passphrase when prompted
     - **cd C:/solaSource/OpenTenure** to navigate to the super repository (code) root folder. If this folder does not exist create this folder.
     - **git clone git://github.com/OpenTenure/code** to create the main repository for Open Tenure
     - **cd code**
     - ./**gits –v populate** if you lose your connection during this action repeat the gits populate command until you are successful for all 10 git repositories.