

HDS Explorer manual

version 0.9.8

Paulo Filimone, Philimone's Group

1. Introduction

HDS-Explorer stands for Health and Demographics Surveillance Data Explorer;

A mobile health and demographics surveillance data collection system that facilitates efficient, accurate data collection for program or research implementation in low-resource settings.

It allows users to collect data for demographic surveillance programs or research studies and store that data on the HDS-Explorer Server data repository app.

The mobile application provides powerful validation tools to ensure data accuracy and completeness, and it includes features like household registration, visit registration, member enumeration, marital relationship registration, migration registration, death registration, pregnancy registration, birth registration, and more.

It also allows users to create and manage ODK (Open Data Kit) forms, pre-populate data from external datasets, group forms together, and create follow-up lists for regions, households and members.

HDS-Explorer is an open-source software developed by Paulo Filimone, Philimone's Group, Mozambique.

2. System Setup

This section contains the instructions on how to set-up the HDS-Explorer server also will cover the usage of the web and mobile application.

System Requirements

For the server application:

- > Linux based distribution (eg. Ubuntu 18.04, Ubuntu 22.04)
- > Minimum Java JDK 8+
- > Java servlet container 3.0 (eg. Tomcat 7+)
- > JDBC supported database system (eg. MySQL, Postgres, etc)

For the mobile application:

- > Minimum Android 5.0+
- > Minimum ODK Collect 1.4.4+

For the current manual we will be using a Ubuntu Linux 22.04 server as the operating system (you must ensure that you install an SSH Server into Ubuntu), you can also use a Windows based system as long as you install Java and a Java server container on it.

See below the list of apps we will be using for setting up the server:

- > Ubuntu Linux 22.04
- > Java OpenJDK 8
- > Apache Tomcat 9
- > MySQL Server 8
- > HDS-Explorer server v1.1.2 build 404

Installing OpenJDK on Ubuntu Linux

Using the terminal type the commands below

Install OpenJDK 8+

```
sudo apt install openjdk-8-jdk
```

Set environment variables for JAVA_HOME

```
sudo nano /etc/environment  
JAVA_HOME="/usr/lib/jvm/java-8-openjdk-amd64"
```

Reload the environment variables - to allow access the java path variable

```
source /etc/environment
```

Show the java current path by using the environment variable

```
echo $JAVA_HOME
```

Installing Apache Tomcat 9 on Ubuntu Linux

Using the terminal type the commands below

```
cd /opt  
sudo wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.108/bin/apache-tomcat-9.0.108.tar.gz
```

Extract the tomcat downloaded compressed (tarball) file into /opt directory

```
sudo tar xzvf apache-tomcat-9.0.108.tar.gz
```

Rename the installation directory (/opt/apache-tomcat-9.0.108) to /opt/tomcat

```
sudo mv /opt/apache-tomcat-9.0.108 /opt/tomcat
```

Create Tomcat User and Group in Linux distribution

```
sudo useradd -r -s /sbin/nologin tomcat
```

Change ownership and permissions of Tomcat installation directory

```
sudo chown -R tomcat: /opt/tomcat/  
sudo chmod -R 755 /opt/tomcat/ or sudo chmod -R u+rw,g+rx,o+rx /opt/tomcat/
```

Edit the tomcat-users.xml file to add an admin user

```
sudo nano /opt/tomcat/conf/tomcat-users.xml
```

In the body of the tag <tomcat-users> put the text below

```
<role rolename="admin-gui"/>  
<role rolename="manager-gui"/>  
<role rolename="manager-script"/>  
<user username="admin" password="data" roles="manager-gui,admin-gui,manager-script"/>
```

For a manager to be accessible from any host/IP, you need to change the manager context.xml

```
sudo nano /opt/tomcat/webapps/manager/META-INF/context.xml
```

Inside context.xml comment the Valve section tag

```
<!--  
    <Valve className="org.apache.catalina.valves.RemoteAddrValve"  
        allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" />  
-->
```

Start and shutdown commands

```
sudo sh /opt/tomcat/bin/shutdown.sh  
sudo sh /opt/tomcat/bin/startup.sh
```

After starting up, to test Tomcat go-to your web-browser and type,
http://ip_address:8080 (8080 is the default port)

Installing MySQL Server 8 on Ubuntu Linux

Using the terminal type the commands below

`sudo apt update`

Install MySQL Server 8 (it will prompt you to set the password)

`sudo apt install mysql-server`

Improve MySQL Installation Security

This program enables you to improve the security of your MySQL installation in the following ways:

- ◆ You can set a password for root accounts.
- ◆ You can remove root accounts that are accessible from outside the local host.
- ◆ You can remove anonymous-user accounts.
- ◆ You can remove the test database (which by default can be accessed by all users, even anonymous users), and privileges that permit anyone to access databases with names that start with test_.

`sudo mysql_secure_installation`

If you can't set a new password, login as mysql root

`sudo mysql`

`alter user 'root'@'localhost' identified with mysql_native_password by 'mynewpassword';`

Showing mysql version

`mysqladmin -u root -p version`

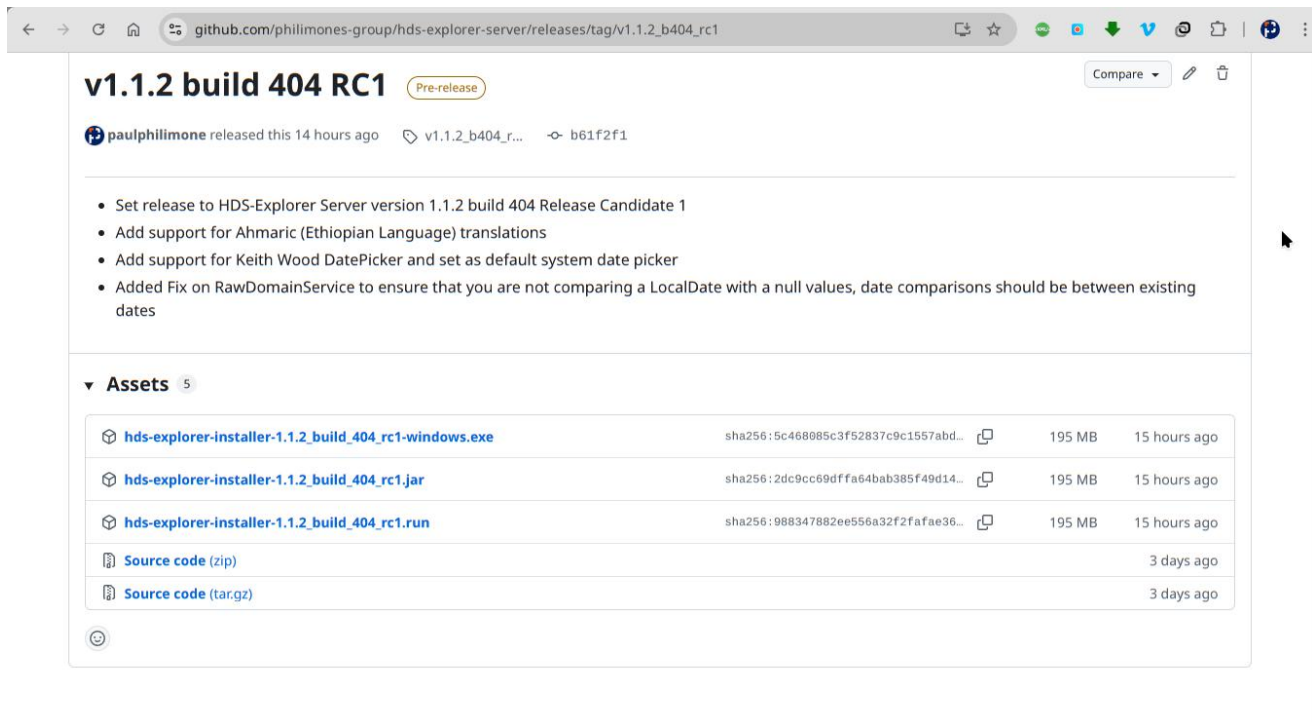
Configuring and setting up HDS-Explorer on Tomcat Server

HDS-Explorer its an open-source software and the code can be found on GitHub repositories, also the latest binaries files used for deploying the app on Tomcat and Android can the found on the repositories below:

Server App:	https://github.com/philimones-group/hds-explorer-server/releases
Mobile App:	https://github.com/philimones-group/hds-explorer-tablet/releases

Download the HDS-Explorer server setup wizard to help configuring the server,
The setup wizard wont actually install anything but will costumize the WAR file with database access and app parameters to become the default and then this WAR file will be deployed on a Tomcat Server.

1. Using a Internet Browser navigate to the link:
<https://github.com/philimones-group/hds-explorer-server/releases>



Download options:

For all the options you must have installed a Java JDK minimum version 8+

1. Linux executable installer: ***hds-explorer-installer-1.1.2_build_404_rc1.run***
While under a Desktop Linux Distribution double click this file to open the setup wizard
2. Windows executable installer: ***hds-explorer-installer-1.1.2_build_404_rc1-windows.exe***
While under a Windows OS double click this file to open the setup wizard
3. Java executable jar file: ***hds-explorer-installer-1.1.2_build_404_rc1.jar***
On a terminal execute:
`java -jar hds-explorer-installer-1.1.2_build_404_rc1.jar`

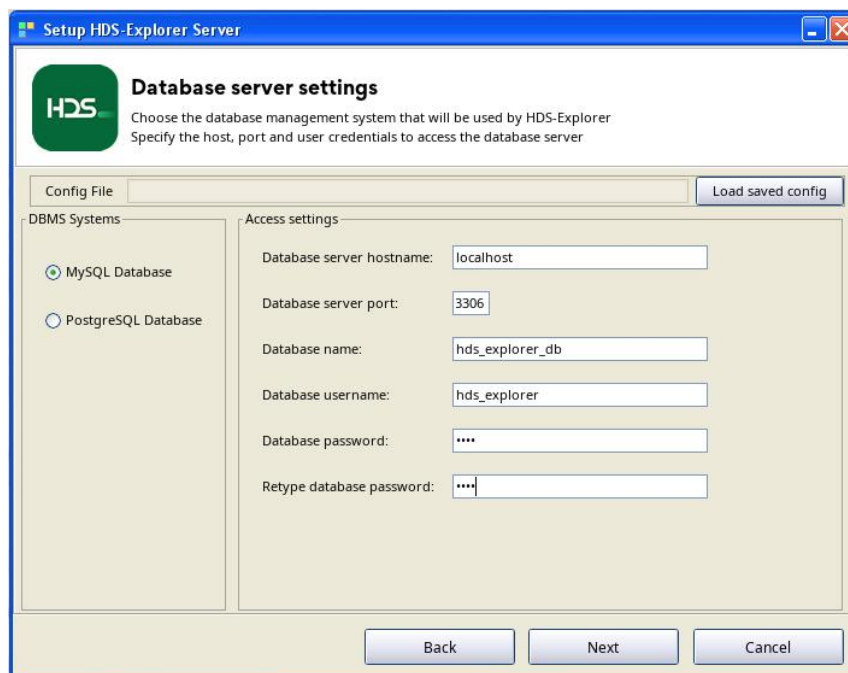
This setup wizard will help you customize the app parameters and generate a WAR file to be deployed on Tomcat server, each release of HDS-Explorer Server will be rolling out a new version of the installer containing a specific version of the app.

Follow the steps below:

1. After executing the file a window will be displayed with the overview of the app and setup wizard



2. Click “Next” to go to “Database server settings” panel



On the database server settings panel you are able to customize the way you access the HDS-Explorer database, goes from defining the database server system that you will be using to the database name and user credentials.

Select according to your HDSS Database management system, by default the system supports MySQL and PostgreSQL, but this can be extended to any database management system with a JDBC Driver and the access can be customized in the review panel text settings editor.

- ◆ **Config file**, contains the filename of the configuration file loaded from previous setup wizards
- ◆ **Load saved config**, when clicked allows the selection of a configuration file, that contains all the parameters that need to be filled on this setup wizard, this function is used to save the user the time of filling out parameters that already were filled before, in this way you can save a setup wizard configuration file and reuse in future versions.

3. After filling the form, click “Next” to go to “System settings” panel

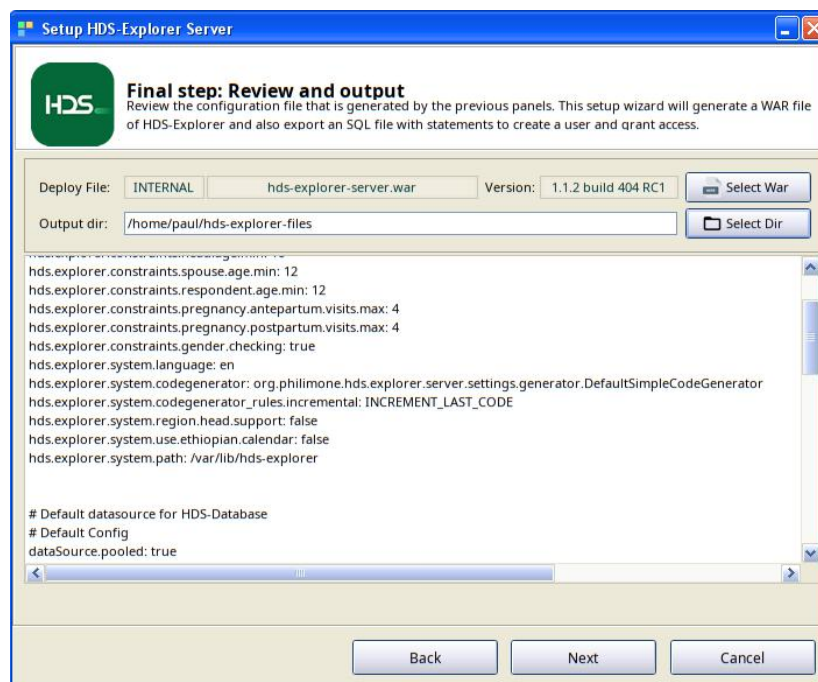
In this panel we can set the default parameters that will be used for more details check below:

- **System resources path** its the folder that contains attachments (csv datasets, follow-up lists), exported datasets used for synchronization and the application error logs
- **System Language** is the language that will be used internally on the system to generate reports and logs, currently the app supports English, French, Portuguese and Ahmaric (Ethiopian translation and Calendar Date Input is available from version **v1.1.2 build 404**).
- **System Calendar** defines the calendar system used for entering and displaying dates in both the mobile app and the web application. All dates are stored and processed internally using the **Gregorian calendar**, regardless of the selected display format.
- **System code generator** represents the regions, households, members code scheme generators, that are essentially the visible ID's that will be granted to each region, household and member.
The system its opened to implement a variety of code schemes that can be easily programmed to be supported by system.

Up to the date of creation of this manual the system only supports the code schemes described below:

1. **Default code generator**, that uses for:
 - a) **User/Fieldworker**: the code has 3 characters taken from the name of user, eg. *PFI* from “Paulo Filimone”
 - b) **Region**: the code has 3 characters taken from the name of the region, eg. *TXU* from “Txumene”
 - c) **Household**: the code contains Region code + User code + 001 (ordinal number), eg. *TXUPFI001*
 - d) **Member**: the code contains the Household code + 001 (ordinal number), eg. *TXUPFI001002*
2. **Default simple code generator**, that uses for:
 - a) **User/Fieldworker**: the code has 3 characters taken from the name of user, eg. *PFI* from “Paulo Filimone”
 - b) **Region**: the code has 3 characters taken from the name of the region, eg. *TXU* from “Txumene”
 - c) **Household**: the code contains Region code + 000001 (ordinal number), eg. *TXU000001*
 - d) **Member**: the code contains the Household code + 001 (ordinal number), eg. *TXU000001002*
3. **Compound Based code generator**, usually used for sites that locate households within compounds:
 - a) **User/Fieldworker**: the code has 3 characters taken from the name of user, eg. *PFI* from “Paulo Filimone”
 - b) **Region**: the code has 3 characters taken from the name of the region, eg. *TXU* from “Txumene”
 - c) **Lowest Region (Compound)**: its the last Region level before locating the Household, the code contains the previous Region code + 000001 (ordinal number), eg. *TXU000001*
 - d) **Household**: the code contains the Lowest Region code + 001 (ordinal number), eg. *TXU000001001*
 - e) **Member**: the code contains the Household code + 001 (ordinal number), eg. *TXU000001001001*
4. **Kimpese HDSS Code Scheme Generator - DRC**, used specially in Kimpese HDSS and meets the criteria of this particular HDSS, eg. (Household no: A04A00101).
(Any group or HDSS site can request guidance on how to have they own HDSS Code Scheme)

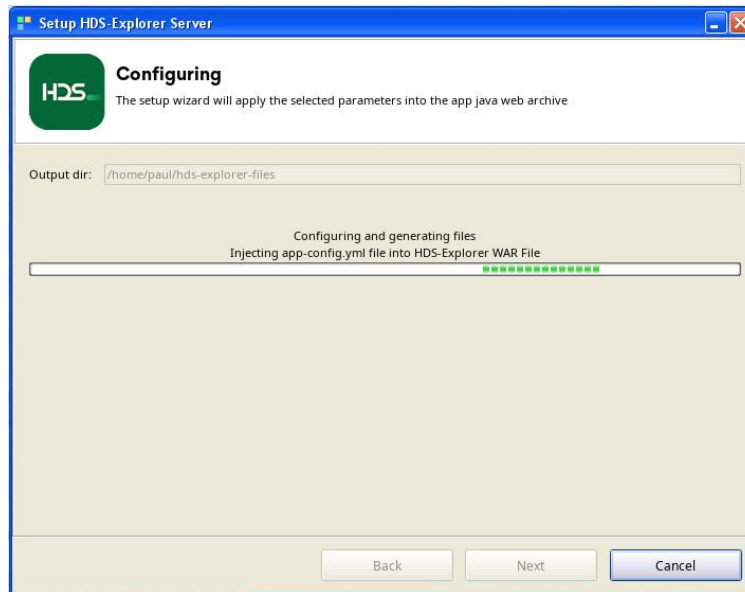
5. Navrongo HDSS Compound-Based Code Scheme Generator, used specially in Navrongo HDSS site, Ghana, its a Custom Compound Based code generator.
 - **Code Generator Incremental Rule:** represents the rule that controls the way new codes will be sequentially generated by the system, two options are available:
 1. **Fill Gaps:** the system will try first to fill any gap of code that wasnt used before it increments one unity on the highest generated code
 2. **Increment the last maximum code:** the system will increment one unity to the last highest generated code
 - **Minimum father age:** the minimum age to be a father in the app, its used as a constraint in the system
 - **Minimum mother age:** the minimum age to be a mother in the app, its used as a constraint in the system
 - **Minimum spouse age:** the minimum age to be a spouse (marital relationship registration)
 - **Minimum head of household age:** the minimum age to be a head of a household in the app
 - **Minimum Respondent Age:** the minimum age to be a respondent to perform a Visit
 - **Maximum Antepartum Visits:** the maximum number of antepartum to complete pregnancy surveillance
 - **Maximum Postpartum Visits:** the maximum number of postpartum to complete pregnancy surveillance
 - **Mail configuration:** contains the gmail credentials to enable the app to send email notifications
4. After filling the form, click “Next” to go to “Review and outputs” panel



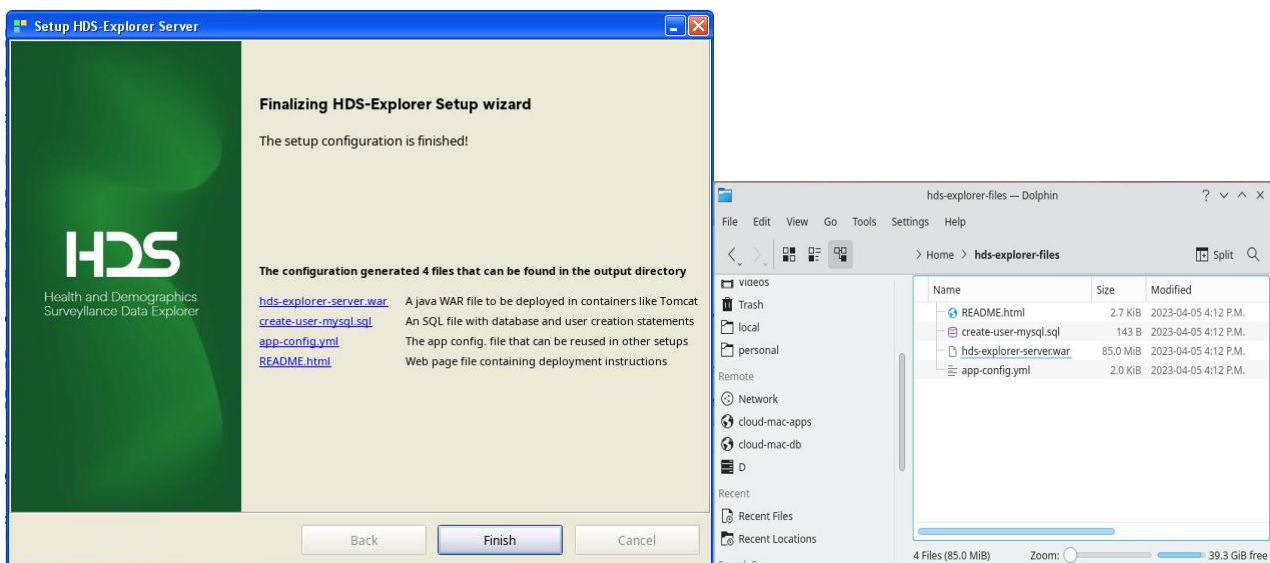
This panel shows the configuration file text generated by the parameters in filled in the previous panels so that you are able to review the configurations.

- ◆ **Deploy File description:** contains the state and version of HDS-Explorer that we are configuring
 - **INTERNAL**, indicates that the WAR file is inside the setup wizard executable file
 - **EXTERNAL**, indicates that the WAR file was selected by the user using the button “Select War”
- ◆ **Output directory:** its the destination folder to where the files will be created the files below:
 - Database creation and access SQL Statements File
 - A the customized HDS-Explorer War file to deploy on tomcat,
 - The app configuration file (that can be reloaded on future setup wizard to use the same settings)
 - An html file containing instructions on how to deploy and install the HDS-Explorer on your server
- ◆ **Select Dir button**, allows you to select the output directory folder
- ◆ **Select War button**, allows the user to select a different WAR file than the one provided by the installer itself, this function allows the setup wizard to configure different versions of HDS-Explorer War files

5. Click “Select dir” to select the output directory and after click “Next” to start the configuration



6. After configuration finishes four files will be created on the output directory



The generated files are:

- ◆ **hds-explorer-server.war**, a java WAR file to be deployed in server containers like Tomcat
- ◆ **create-user-mysql.sql**, an sql file with database and user creation statements (we used MySQL as the db)
- ◆ **app-config.yml**, the configuration file that can be used on other setup wizards by loading saved configs
- ◆ **README.html**, web page file containing the deployment instructions

7. After completing this setup wizard follow the instructions on the README.html file that look like below:

HDS-Explorer Server

platform Grails platform Groovy license Apache%202.0

HDS-Explorer Server application provides the data repository for a Health and Demographics Surveillance system and data management tool for data collection for program or research implementation

This application server can be deployed on a Tomcat server (or any servlet 2.5+ compatible web container) and backed by any Hibernate supported database (eg. MySQL or PostgreSQL database server, etc).

- HDS-Explorer website: <https://www.hds-explorer.org>

Deployment Instructions

After using the HDS-Explorer Setup wizard configuration, to complete the installation you need to:

Lets consider that we are in a Linux distribution and Tomcat Server that was manually installed in the path: /opt/tomcat.

- 1. Goto the output directory where the installer created the files**
`cd /home/paul/hds-explorer-files`
- 2. Copy the the files (hds-explorer-server.war and create-user-mysql.sql) to the Server where tomcat is installed (In Linux or WSL you can you SSH Copy commands to copy the files or an equivalent in Windows)**
Use the appropriate way for you to copy those files
- 3. Configure the MySQL Database Access (import the file *create-user-mysql.sql* to MySQL database)**
`mysql -u $MYSQL_USER -p < create-user-mysql.sql`
- 4. Create the hds-explorer resources directory**
`sudo mkdir /var/lib/hds-explorer`
- 5. Grant access to tomcat user to the folder /var/lib/hds-explorer**
`sudo chown -R tomcat:tomcat /var/lib/hds-explorer`
- 6. Copy the war file to \$TOMCAT/webapps**
`cp hds-explorer-server.war /opt/tomcat/webapps`
- 7. Start the webserver or we can copy the war file while the server is running and it will be deployed onfly.**
`/opt/tomcat/bin/startup.sh`

Deployment Instructions

After using the HDS-Explorer Setup wizard configuration, to complete the installation you need to:
Let's consider that we are in a Linux distribution and Tomcat Server that was manually installed in the path: /opt/tomcat.

1. Goto the output directory where the installer created the files
`cd /home/paul/hds-explorer-files`
2. Copy the files hds-explorer-server.war and create-user-mysql.sql to the server that contains the Tomcat
You can use a SSH Copy to do this process or even putty if you are using Windows
3. Configure the MySQL Database Access (import the file create-user-mysql.sql to MySQL database)
`mysql -u $MYSQL_USER -p < create-user-mysql.sql`
4. Create the hds-explorer resources directory
`sudo mkdir /var/lib/hds-explorer`
5. Grant access to tomcat user to the folder /var/lib/hds-explorer
`sudo chown -R tomcat:tomcat /var/lib/hds-explorer`
6. Copy the war file to \$TOMCAT/webapps
`cp hds-explorer-server.war /opt/tomcat/webapps`
7. Start the webserver or we can copy the war file while the server is running and it will be deployed onfly.
`/opt/tomcat/bin/startup.sh`

Now you need to open the server app through the Internet Browser:

Use <http://localhost:8080/hds-explorer-server> where localhost can be your server's IP Address or hostname.