Open Hospital 1.15.2 - Administrator’s Guide

Table of Contents

Open Hospital . . . 5
1 Introduction . . . 5
1.1 Main Features . . . 6
1.2 Hardware requirements . . . 6
1.3 Software requirements . . . 6
1.5 Download . . . 7
1.6 Unpack . . . 7
1.7 Documentation . . . 8
2 Installation . . . 8
2.1 OH - CLIENT mode . . . 8
2.1.1 Prerequisites - Software compatibility matrix . . . 9
2.1.2 Java Runtime Environment (JRE) . . . 13
2.1.3 Database Server . . . 14
2.1.3.1 MariaDB/MySQL - Server installation . . . 14
2.1.3.2 MariaDB/MySQL - Configuration . . . 15
2.1.3.3 MariaDB/MySQL - Networking . . . 16
2.1.4 OH - Database creation . . . 16
2.1.5 OH - CLIENT configuration . . . 18
2.1.6 OH - Startup and run . . . 20
2.2 OH - PORTABLE mode . . . 21
Linux . . . 22
Windows . . . 22
Menu and options . . . 22
Advanced options . . . 23
Script configuration . . . 24
OH directory path . . . 24
(Windows only) Enable interactive mode . . . 24
Config file generation . . . 24
Distribution type - CLIENT | PORTABLE | SERVER . . . 25
Interface and software language: . . . 25
Single / multi user . . . 25
Log level / debug mode . . . 25
Demo mode . . . 25
Enable system wide JAVA . . . 26

Database configuration . . . 26
OH configuration . . . 26
Default directory structure . . . 27
Documentation . . . 27
Other issues . . . 27
Linux . . . 27
Windows . . . 28
Windows - create startup shortcut . . . 28
Powershell configuration . . . 29
Windows - legacy mode . . . 29
Java . . . 29
MariaDB . . . 29
oh.sh / oh.ps1 - features and development . . . 30
2.3 OH - SERVER mode (portable) . . . 30
2.4 Backup & Restore . . . 30
2.4.1 Backup of database (alternative with MySQL Workbench®) . . . 30
2.4.2 Restore of database (alternative with MySQL Workbench®) . . . 32
2.4.3 Backup of OH code and settings . . . 32
2.5 Folders and directory structure . . . 33
2.6 Appearance . . . 34
3 Configuration . . . 34
3.1 settings.properties . . . 35
3.1.1 LANGUAGE . . . 37
3.1.2 SINGLEUSER . . . 38
3.1.3 AUTOMATICLOT . . . 39
3.1.4 LOTWITHCOST . . . 40
3.1.5 PATIENTSHEET . . . 40
3.1.6 OPDCHART . . . 40
3.1.7 ADMCHART . . . 40
3.1.8 DISCHART . . . 41
3.1.9 PATIENTBILL . . . 41
3.1.10 BILLSREPORT . . . 41
3.1.11 BILLSREPORTMONTH . . . 42
3.1.12 PHARMACEUTICALORDER . . . 42
3.1.13 PHARMACEUTICALSTOCK . . . 43
3.1.14 PATIENTEXTENDED . . . 43
3.1.15 OPDEXTENDED . . . 43
3.1.16 MATERNITYRESTARTINJUNE . . . 44
3.1.17 LABEXTENDED . . . 44
3.1.18 LABMULTIPLEINSERT . . . 44
3.1.19 INTERNALPHARMACIES . . . 45

3.1.20 MERGEFUNCTION . . . 45
3.1.21 INTERNALVIEWER . . . 45
3.1.22 DOC_DIR . . . 46
3.1.23 SMSENABLED . . . 47
3.1.23.1 GSM Configuration . . . 47
3.1.23.2 SetupGSM utility . . . 47
3.1.24 MAINMENUALWAYSONTOP . . . 48
3.1.25 RECEIPTPRINTER . . . 48
3.1.26 VIDEOMODULEENABLED . . . 48
3.1.27 PATIENTVACCINEEXTENDED . . . 49
3.1.28 ENHANCEDSEARCH . . . 49
3.1.29 XMPPMODULEENABLED . . . 49
3.1.30 DICOMMODULEENABLED . . . 50
3.1.31 DICOMTHUMBNAILS . . . 50
3.1.32 DEBUG . . . 50
3.1.33 ALLOWMULTIPLEOPENEDBILL . . . 51
3.1.34 OPENEDBILLSREPORT . . . 52
3.1.35 ALLOWPRINTOPENEDBILL . . . 52
3.1.36 USERSLISTLOGIN . . . 52
3.1.37 STRONGPASSWORD . . . 53
3.1.38 STRONGLENGTH . . . 53
3.1.39 PASSWORDTRIES . . . 54
3.1.40 PASSWORDLOCKTIME . . . 54
3.1.40 PASSWORDIDLE . . . 54
3.1.41 PATIENTPHOTOSTORAGE . . . 55
3.1.42 SESSIONTIMEOUT . . . 55
3.1.43 EXAMINATIONCHART . . . 55
3.1.44 VISITSHEET . . . 56
3.1.45 TELEMETRYENABLED . . . 56
3.1.46 PARAMSURL . . . 57
3.2 database.properties . . . 57
3.3 dicom.properties . . . 58
3.4 examination.properties . . . 59
3.5 log4j2-spring.properties . . . 61
3.6 sms.properties . . . 63
3.6.1 skebby-gateway-service . . . 64
3.6.2 textbelt-gateway-service . . . 64
3.7 telemetry.properties . . . 65
3.8 txtPrinter.properties . . . 66
3.9 xmpp.properties . . . 68
3.9.1 OpenFire Settings . . . 68

3.10 default_credentials.properties . . . 69
3.11 Bundles . . . 70
3.11.1 New Translations . . . 70
4 Reports . . . 71
5 Installing Open Hospital 1.15.0 in Eclipse EE . . . 72
5.1 Run the Project . . . 72
6 Update Open Hospital . . . 76
6.1 Update Database . . . 76
6.2 Update Client . . . 77
6.3 Update Portable . . . 77
7 Support . . . 78
8 License . . . 79



Informatici Senza Frontiere Onlus, 2026
Administrator's Guide, © 2026 by Informatici Senza Frontiere Onlus
Policies is made available under a Creative Commons Attribution-ShareAlike 4.0 International License: https://creativecommons.org/licenses/by-sa/4.0//.

Open Hospital

1 Introduction

OH - Open Hospital (https://www.open-hospital.org/) is a free and open-source Electronic Health Record (EHR) software application. Open Hospital is deployed as a desktop application that can be used in a standalone, single-user mode (PORTABLE mode) or in a client/server network configuration (CLIENT mode), where multiple clients and users connect to the same database server.

Open Hospital is developed in Java and it is based on open-source tools and libraries; it runs on any computer, requires low resources, and is designed to work without an internet connection.

Open Hospital is the first of a set of software applications that ISF[1] has developed to support the information management and the activities of hospitals and health centers in the simplest manner possible, by providing tools for the administrative operations (like registering patients, managing laboratory analysis and pharmaceutical stocks) and to produce detailed statistics and reports. It was first deployed in 2006 at the St. Luke Hospital in Angal (Uganda) and it is now used in dozens of different locations around the world.

1.1 Main Features

OH - Open Hospital features include:

- Pharmacy management
- Laboratory management
- OPD management
- Patient admission and discharge management
- Pregnancy management
- Malnutrition control management
- Vaccines database
- Patient billing support
- Therapy management
- Appointment scheduling
- Internal communication
- Statistics and printing

For a detailed description of these features please refer to the *Open Hospital User's Guide*.

1.2 Hardware requirements

Minimum hardware requirements for running OH:

- PC / Notebook with dual-core CPU
- 2 Gb RAM (4 Gb recommended)
- 4 Gb free hard disk space (8 Gb recommended)
- Network adapter (for client/server configuration)

These specifications are for the OH application only; database server specifications may vary accordingly to the different use cases.

1.3 Software requirements

Minimum Operating System version required for running OH:

- Windows 7 (Windows 10/11 recommended)
- Ubuntu 18.04 (Ubuntu 22.04 recommended)

32bit (i686) and 64bit (x86_64) architectures are supported: 64bit architecture is recommended.

1.5 Download

Open Hospital official releases can be downloaded from GitHub: https://github.com/informatici/openhospital/releases/latest

More information can be found on the Open Hospital website: https://www.open-hospital.org/download

1.6 Unpack



Do not double-click on the **oh.bat** startup script without unzipping the package first!

A compressed archive can be opened on most operating systems by just double-clicking on it. In this way, the archive is just opened, but not uncompressed.

In order to correctly run OH **you need to unzip (uncompress) the downloaded package** in a folder of your choice.

To uncompress the package: **Right-click on the package → Extract all/here**

Alternative method: uncompress package from the command line

- on Linux:

```
tar zxvf OpenHospital-v[VERSION]-linux_[ARCH]-[portable].tgz # portable version
unzip x OpenHospital-v[VERSION]-linux_multiarch.zip # multiarch client version
```

- on Windows:

```
unzip OpenHospital-v[VERSION]-windows_[ARCH]-[portable].zip # portable version
unzip OpenHospital-v[VERSION]-multiarch-client.zip # multiarch client version
```

After uncompressing the package, browse to the extracted directory (example given for version 1.15.0):

- on Linux:

```
cd /home/OH/OpenHospital-v1.15.0
```

- on Windows:

```
cd C:\Users\OH\OpenHospital-v1.15.0
```



You need read/write permission on the selected folder.



Do not run OH by double-clicking on the **oh.bat** startup script without unzipping the package first!

1.7 Documentation

In the following chapters, all the information needed to install, configure, deploy, run and maintain an Open Hospital installation is presented, including procedures on how to enable and disable features as well as manage users and groups in a multi-user environment. More information can be found on the Open Hospital website https://www.open-hospital.org/documentation.



The information needed to use the OH software is not included in this manual; please refer to the *Open Hospital User's Guide*.

2 Installation

Open Hospital is a client application that can be used in three different ways:

- 2.1 OH - CLIENT mode: multi-computer setup, with a central database (requires some IT knowledge)
- 2.2 OH - PORTABLE mode: single computer setup, with the database on the same computer (click and run)
- 2.3 OH - SERVER mode (portable): multi-computer setup, with the central database created by the Open Hospital script (click and run)



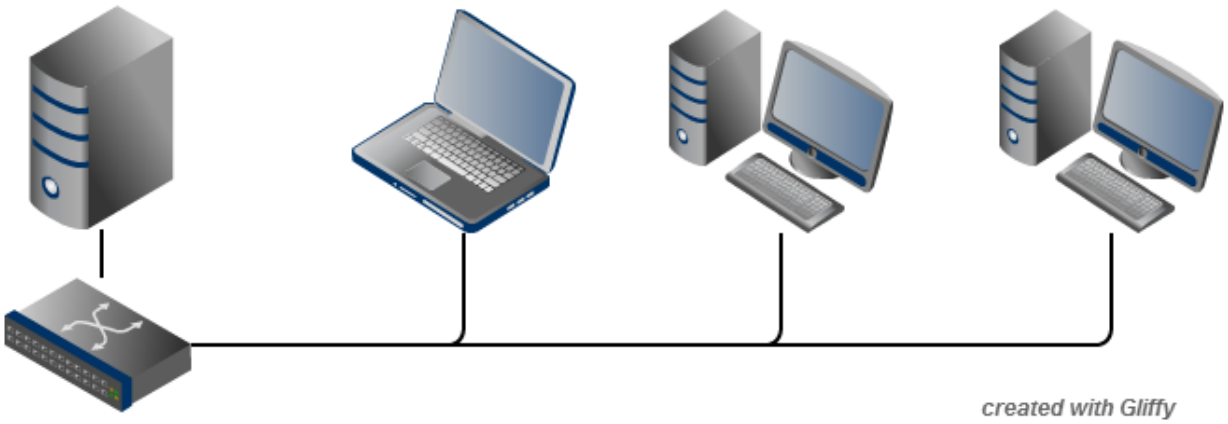
PORTABLE mode can be also used to test the application.

2.1 OH - CLIENT mode

In this mode, the application will need a Server with a DBMS[2] to store the data.

The free and open-source MariaDB[3] is the preferred database server.

The typical client/server, networked (LAN) configuration is shown in the following figure:



The Administrator/Installer tasks are hereby listed:

- Server
  - Download the Open Hospital package for the selected Operating System (see Prerequisites)
  - Unpack the downloaded package on the server machine by uncompressing the archive in a folder of your choice
  - Install and configure MariaDB/MySQL on the server/computer that will act as the database server
  - Create the database through the execution of a script found in the *sql/* folder
  - Test and run the application using the 'localhost' setting
- Clients
  - Unpack the downloaded package on a client machine in a folder of your choice
  - Configure and run the application to point to the database server using the hostname or the IP address
  - Copy the newly configured package on every client

2.1.1 Prerequisites - Software compatibility matrix

Before starting, check the following software compatibility matrix to verify the external software versions needed to run Open Hospital and the current available extended features for every version, architecture, and operating system.

Open Hospital compatibility matrix								
OH version	**MariaDB** / MySQL (community version)	**JAVA** JRE version (*)	O.S.	O.S. arch	JAVA/MariaDB arch	Special Features		
						Imaging - DICOM (*)	SMS (gsm)	Webcam

Open Hospital compatibility matrix								
1.15.2	**10.6.28** / n.a.	**zulu17.68.203**	**Ubuntu 22.10**	**x64**	**64bit**	✅	n/t	✅
			Windows 10/11 (**)	**x64**	**64bit**	✅	✅	✅
	10.5.28	zulu17.68.203	*Linux32*	*i686*	*32bit*	n/t	n/t	n/t
	10.6.5	zulu17.68.203	*Windows 7/10 (**)*	*i386*	*32bit*	✅	✅	✅
1.15.1	**10.6.23** / n.a.	**zulu17.60.17**	**Ubuntu 22.10**	**x64**	**64bit**	✅	n/t	✅
			Windows 10/11 (**)	**x64**	**64bit**	✅	✅	✅
	10.5.27	zulu17.60.17	*Linux32*	*i686*	*32bit*	n/t	n/t	n/t
	10.6.23	zulu17.60.17	*Windows 7/10 (**)*	*i386*	*32bit*	✅	✅	✅
1.15.0	**10.6.23** / n.a.	**zulu17.60.17**	**Ubuntu 22.10**	**x64**	**64bit**	✅	n/t	✅
			Windows 10/11 (**)	**x64**	**64bit**	✅	✅	✅
	10.5.27	zulu17.60.17	*Linux32*	*i686*	*32bit*	n/t	n/t	n/t
	10.6.23	zulu17.60.17	*Windows 7/10 (**)*	*i386*	*32bit*	✅	✅	✅
1.14.2	**10.6.14** / n.a.	**zulu17.46.19**	**Ubuntu 22.10**	**x64**	**64bit**	✅	n/t	✅
			Windows 10/11 (**)	**x64**	**64bit**	✅	✅	✅
	10.5.21	zulu17.46.19	*Linux32*	*i686*	*32bit*	n/t	n/t	n/t
	10.6.5	zulu17.46.19	*Windows 7/10 (**)*	*i386*	*32bit*	✅	✅	✅

Open Hospital compatibility matrix								
1.14.1	**10.6.14** / n.a.	**zulu17.46.19**	**Ubuntu 22.10**	**x64**	**64bit**	✅	n/t	✅
			Windows 10/11 (**)	**x64**	**64bit**	✅	✅	✅
	10.5.21	zulu17.46.19	*Linux32*	*i686*	*32bit*	n/t	n/t	n/t
	10.6.5	zulu17.46.19	*Windows 7/10 (**)*	*i386*	*32bit*	✅	✅	✅
1.14.0	**10.6.14** / n.a.	**zulu17.46.19**	**Ubuntu 22.10**	**x64**	**64bit**	✅	n/t	✅
			Windows 10/11 (**)	**x64**	**64bit**	✅	✅	✅
	10.5.21	zulu17.46.19	*Linux32*	*i686*	*32bit*	n/t	n/t	n/t
	10.6.5	zulu17.46.19	*Windows 7/10 (**)*	*i386*	*32bit*	✅	✅	✅
1.13.0	**10.6.14** / n.a.	**zulu11.64.19**	**Ubuntu 22.10**	**x64**	**64bit**	✅	✅	✅
			Windows 10/11 (**)	**x64**	**64bit**	✅	✅	✅
	10.5.21	zulu11.64.19	*Linux32*	*i686*	*32bit*	✅	✅	n/t
	10.6.5	zulu11.64.19	*Windows 7/10 (**)*	*i386*	*32bit*	✅	✅	n/t
1.12.1	**10.6.12** / n.a.	**zulu11.62.17**	**Ubuntu 22.10**	**x64**	**64bit**	✅	✅	✅
			Windows 10/11 (**)	**x64**	**64bit**	✅	✅	✅
	10.5.19	zulu11.62.17	*Linux32*	*i686*	*32bit*	✅	✅	n/t
	10.6.5	zulu11.62.17	*Windows 7/10 (**)*	*i386*	*32bit*	✅	✅	n/t

Open Hospital compatibility matrix								
1.12.0	**10.6.11** / n.a.	**zulu11.62.17**	**Ubuntu 22.10**	**x64**	**64bit**	✅	✅	✅
			Windows 10/11 (**)	**x64**	**64bit**	✅	✅	✅
	10.5.18	zulu11.62.17	*Linux32*	*i686*	*32bit*	✅	✅	n/t
	10.6.5	zulu11.62.17	*Windows 7/10 (**)*	*i386*	*32bit*	✅	✅	n/t
1.11.5/4/3	**10.2.42** / n.a.	**zulu8.60.0.21**	Ubuntu 21.10	x64	64bit	✅	✅	✅
			Windows 7/10/11 (**)	x64	64bit	✅	✅	✅
			Windows 7/10/11 (**)	x64	32bit	✅	✅	✅
			Windows 7/10 (**)	*i686*	*32bit*	✅	✅	✅
1.11.2/1	**10.2.41**/5.7.35	**zulu8.58.0.13**	Ubuntu 21.10	x64	64bit	✅	✅	✅
			Windows 7/10/11 (**)	x64	64bit	✅	✅	✅
			Windows 7/10/11 (**)	x64	32bit	✅	✅	✅
			Windows 7/10 (**)	i386	32bit	✅	✅	✅
1.11.0	**10.2.40**/5.7.35	**zulu8.56.0.21**	Ubuntu 21.04	x64	64bit	✅	✅	✅
			Ubuntu 18.04	i386	32bit	✅	n/t	n/t
			Windows 7/10 (**)	x64	64bit	✅	✅	✅
			Windows 7/10 (**)	x64	32bit	✅	✅	✅
			Windows 7/10 (**)	i686	32bit	✅	✅	✅

Open Hospital compatibility matrix								
1.10.0	5.1.x	1.6	Ubuntu 20.04	x64	64bit	✅	n/t	✅
			Ubuntu 18.04	i386	32bit	✅	✅	n/t
			Windows 7/10	x64	64bit	✅	✅	✅
			Windows 7/10	x64	32bit	✅	✅	✅
			Windows 7/10	i386	32bit	✅	✅	✅
1.9.1	5.0.x	1.6	Ubuntu 20.04	x64	64bit	✅	n/t	✅
			Ubuntu 18.04	i386	32bit	✅	n/t	n/t
			Windows 7/10	x64	64bit	✅	✅	✅
			Windows 7/10	x64	32bit	✅	✅	✅
			Windows 7/10	i386	32bit	✅	✅	✅

NOTE	**32-bit environments are EOL (End Of Life)**
(*)	Before OH v.1.12.0, Imaging/DICOM functionality may not work
(**)	Windows Powershell 5.1 is required
✅	*working*
❌	*not working*
n/t	*not tested*
n/a	*not applicable*
last updated	*2023.02.21*

2.1.2 Java Runtime Environment (JRE)

Depending on the operating system, there are different ways to install the Java Runtime Environment (JRE). The **oh.sh** / **oh.bat** scripts can be used to download and install the latest JRE available; it is also possible to install it manually, following the specific instructions for the operating system being used.



See OH Software Compatibility Matrix to identify the correct Java version.

2.1.3 Database Server

Depending on the operating system, there are several software tools to install and manage a MariaDB/MySQL database server. In most cases just a standard installation package of the software is needed, paying attention to configuration options during the installation process.



See OH Software Compatibility Matrix to identify the correct MariaDB / MySQL version.



MariaDB is the preferred alternative for the DBMS.

2.1.3.1 MariaDB/MySQL - Server installation

Please refer to the latest MariaDB/MySQL Server online documentation for download[4] and installation[5] information while using this document.

on a Linux machine: during the installation, a “root” password must be defined; it is very important to choose it carefully and to keep it safe.

on a Windows machine: launch the installation and pay attention to the following steps:

- Custom Setup
- Install all components except Development Components
- Select Modify password for database user 'root', choose the root password, and keep it safe
- Select Enable access from remote machines for 'root' user
- Select UTF8 as the default charset for Best Support For Multilingualism
- Select Install as service
- Select Enable networking and take note of the selected Port Number



MariaDB installer does not include the bin directory in the Windows PATH environment variable; it must be added manually to the Windows system variable PATH.

To check if the server is active and running, open a terminal window (command prompt) and type the following command:

 # mysql –u root -p

Enter password:

The terminal should reply with the MySQL client command line prompt:

MariaDB>

The prompt means that a successful connection to the database server instance has been

established.



If the terminal does not reply as above most probably during installation the **Include Bin Directory in Windows PATH** option was not selected and therefore Windows can not find the mysql command. If this is the case add the path manually by searching "system variables" in the Control Panel or append the full path to the "mysql" command (e.g., "C:\Program Files\MySQL\MySQL Server 5.0\bin\mysql" [with the quotes]). Another option is to uninstall and reinstall MySQL with the **Include** option selected.

2.1.3.2 MariaDB/MySQL - Configuration

Please use the following configuration values for MariaDB/MySQL server in *my.cnf* config file. Values can be adjusted and adapted to the hardware resources available. Config file can be *my.ini* or *my.cnf*; default for Windows MariaDB is *my.ini* located at "C:\Program Files\MariaDB-[VERSION]\data\"

```
#
# Configuration to be inserted below last row of [mysqld] section
#
sql_mode=STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,N
O_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION
max_allowed_packet = 4M # must match the value used on clients for DICOM_SIZE
skip-external-locking
key_buffer_size = 16M
thread_cache_size = 64
lower_case_table_names = 1
explicit_defaults_for_timestamp = OFF
table_open_cache = 64
tmp_table_size = 16M
read_buffer_size = 256K
read_rnd_buffer_size = 512K
join_buffer_size = 2M
sort_buffer_size = 2M
myisam_sort_buffer_size = 8M

[mysqldump]
quick
max_allowed_packet = 16M

[mysql]
no-auto-rehash

[isamchk]
key_buffer = 16M
sort_buffer_size = 16M
read_buffer = 2M
write_buffer = 2M
```

```
[myisamchk]
key_buffer = 16M
sort_buffer_size = 16M
read_buffer = 2M
write_buffer = 2M
```

2.1.3.3 MariaDB/MySQL - Networking

If deployed in a client-server, networked configuration, the database server must be configured to listen on the network interface connected to the local LAN; this can be achieved by editing the MariaDB/MySQL's config file, *my.cnf*, and setting the "bind-address" parameter. For example, if you want the database server to listen on all the available network interfaces / IP addresses:

Change:

```
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address = 127.0.0.1
```

To:

```
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address = 0.0.0.0
```

In this way, the database server listens on any IP address (0.0.0.0) configured on the server and can serve the connecting clients.

It is also possible to specify a specific IP address, e.g:

```
bind-address = 192.168.1.100
```

2.1.4 OH - Database creation

To create the Open Hospital database, a MySQL client and a set of SQL script files found under the *sql/* folder are needed.

On a Microsoft Windows® system, MySQL Workbench® can be used as a client to connect to the database server and perform the needed tasks, while on Linux based systems the MySQL Administrator and MySQL Query Browser tools can be used.

The suggested approach is to use the command-line MariaDB/MySQL client which uses the same commands and syntax for every platform.

Open a terminal and navigate to the *sql/* folder (e.g.):

```
C:\WINDOWS> cd C:\Users\OH\OpenHospital-v1.15.0\sql

C:\Users\OH\OpenHospital-v1.15.0\sql
```

and run the following command to connect to the database server (e.g.):

```
C:\Users\OH\OpenHospital-v1.15.0\sql> mysql -u root -p [-h hostname/ip address]
Enter password: *****
```

using the 'root' password chosen during the installation process. The terminal should reply with the MariaDB client command line prompt:

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 13
Server version: 10.6.11-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

we are now ready to create the database user (isf) with a password, the main OH database (oh) and grant the necessary permissions with the following SQL commands:

```
MariaDB> CREATE DATABASE oh CHARACTER SET utf8;
     CREATE USER 'isf'@'localhost' IDENTIFIED BY 'isf123';
     CREATE USER 'isf'@'%' IDENTIFIED BY 'isf123';
     GRANT ALL PRIVILEGES ON oh.* TO 'isf'@'localhost';
     GRANT ALL PRIVILEGES ON oh.* TO 'isf'@'%';
MariaDB> FLUSH PRIVILEGES;
```



"isf123" is used here as the default password. Change it for production setup!

To check if the database has been correctly created, enter the following command:

```
MariaDB> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| oh                 |
| performance_schema |
```

```
+--------------------+
```

The next step is to select the created database and execute the main SQL database creation script with the command:

```
MariaDB> use oh; source create_all_en.sql
```



The data can be installed in different languages by using the related "create_all_xx.sql" file or with demo data using the "create_all_demo.sql" file (English only).



If the error message "ERROR 1148 (42000): The used command is not allowed with this MySQL version" appears the "local infile" command must be enabled on the client by logging in again and specifying the "--local-infile=1" parameter after the "-p"; thus "mysql -u root -p --local-infile=1".

The SQL script creates the OH database structure and populates it with default data.

The ‘isf’ user, different from the ‘root’ one, is the user that the Open Hospital software uses to connect to the database and its password must be changed immediately with the command:

```
MariaDB> SET PASSWORD FOR 'isf'@'%' = PASSWORD('new-password-here');

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0
```

And then:

```
MariaDB> FLUSH PRIVILEGES;

Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB> quit;
```



Remember to set the updated database password also in the Open Hospital configuration files and startup scripts **oh.sh** / **oh.ps1**.

2.1.5 OH - CLIENT configuration

To set up the database connection for Open Hospital, edit and configure the database settings in the **oh.sh** / **oh.ps1** startup scripts.

Database settings are:

- the **IP address** of the MariaDB/MySQL database server hosting the Open Hospital database. Replace “localhost” with the server IP address (e.g., 192.168.1.100)
- the **TCP Port Number** of the database server - default is “3306"
- the **Database name** of the OH installation - default is "oh"
- the **Database user** of the OH database - default is "isf"
- the **Database password** associated with the user

Please set them according to the values specified in the installation phase:

in **oh.sh** / **oh.ps1** scripts

```
$script:MYSQL_SERVER="127.0.0.1"
$script:MYSQL_PORT=3306
$script:MYSQL_ROOT_PW="tmp2021oh111"
$script:DATABASE_NAME="oh"
$script:DATABASE_USER="isf"
$script:DATABASE_PASSWORD="isf123"
```



"isf123" is used here as the default password. Change it for production setup!

After you can generate the config files: on Linux execute **oh.sh -g**, on Windows launch **oh.bat** and choose the option **g**.

Alternative method: you can first generate the config files with the **-g** and after directly modify the *database.properties* and *log4j2-spring.properties* configuration files, located in the *oh/rsc/* resource folder, without generating them every time.

Please set them according to the values specified in the installation phase:

in *database.properties* config file

```
jdbc.url=jdbc:mysql://127.0.0.1:3306/oh
jdbc.username=isf
jdbc.password=isf123
```

in *log4j2-spring.properties* config file

```
appender.jdbc.connectionSource.connectionString =
mysql://127.0.0.1:3306/oh?autoReconnect=true
appender.jdbc.connectionSource.userName = isf
appender.jdbc.connectionSource.password = isf123
```



These settings are required for every Open Hospital installation/client.

More information on these files can be found in the 3. Configuration section.

2.1.6 OH - Startup and run

The Open Hospital software is now ready to be run by executing the proper launcher command:

- on a Linux machine: **oh.sh -C** (CLIENT mode)
- on a Windows machine: **oh.bat** (Select CLIENT mode from the interactive menu)

If everything is configured correctly, the Open Hospital main splash screen is presented:



And then the main Menu:



It is possible to create a shortcut (link) to the executable script on the desktop, in the Programs

Menu or wherever is useful by using the provided icon “oh.ico” that is found in the package.



On Linux, it might be necessary to set the correct permissions to the **oh.sh** script:

```
chmod 755 ./oh.sh
```



Remember to review your setup in the Open Hospital configuration files and startup scripts **oh.sh** / **oh.ps1**.

2.2 OH - PORTABLE mode

Open Hospital is a Java software program and it is portable by definition. The **PORTABLE** mode described in this chapter refers to the possibility of using OH as a self-contained package that includes Java, MariaDB/MySQL Server and all the data, without requiring any software installation. The package can be used on a single computer and used everywhere just by copying a folder - even onto an USB stick - carrying together all the saved data.

Pre-configured "Portable" packages are available for Windows and Linux; see the Download section for more details.

Since the JRE and MariaDB/MySQL are not platform-independent and are provided in different versions and architectures, there exists:

- **Open Hospital - portable for Linux** platforms (embedding Java JRE and MariaDB for Linux, 32 or 64bit)
- **Open Hospital - portable for Windows** platforms (embedding Java JRE and MariaDB for Windows, 32 or 64bit)

Once the required version is downloaded, the application can be started by launching the script inside the package:

- **oh.sh** in the Linux version
- **oh.bat** in the Windows version

No other configuration is needed. Additional options can be shown by executing:

- on Linux: **oh.sh -h**
- on Windows: **oh.bat** and choose among the available options. See **oh.bat -h** for legacy mode help.



To use Open Hospital in PORTABLE mode for Linux from a USB key it is required that the file permissions in the archive remain unchanged once extracted, otherwise the launcher cannot launch or use the application in an appropriate manner.



Important: the PORTABLE mode is meant to try or test the software or to be used in a facility (like a dispensary) where only a single computer is available. If the facility is large and more clients/operators need to work on the same data, the full installation (client/server) in a networked architecture is recommended.

Linux

- Start OH by running **./oh.sh**
- To show available options, run **./oh.sh -h**

Windows

- Double click **oh.bat** and choose among available options.

Note: The **oh.bat** launches the **oh.ps1** startup file automatically. The script presents the interactive menu that can be used to setup and choose how to run Open Hospital.

→ To manually run **oh.ps1** (powershell script):

- right-click on **oh.ps1** → Properties → General → Security
- select "Unblock"
- right click on **oh.ps1** and select "Run with Powershell"
- if asked for permission to execute the script select "Allow"
- choose among available options

It might be necessary to set the correct permissions / exclusions also in the Windows Security Center, to allow OH to communicate on the MySQL / MariaDB local TCP port.

→ To run oh.ps1 directly from command line:

```
powershell.exe -ExecutionPolicy Bypass -File ./oh.ps1 [options]
```

→ To run oh.ps1 with command line options (example):

```
./oh.ps1 -lang it -mode PORTABLE -loglevel DEBUG -interactive off -generate_config on
```

Menu and options

```
 ------------------------------------------------------------------------
|                                                                        |
|                   Open Hospital - v1.15.x                              |
|                                                                        |
 ------------------------------------------------------------------------
| arch: x86_64 | lang: en | mode: PORTABLE | Demo: off | log level: INFO |
```

```
 ------------------------------------------------------------------------
| Expert mode: off | API server: off | GUI: on | UI: off |
 ------------------------------------------------------------------------

 Usage: oh.sh/oh.ps1 -[OPTION]

   -C    set OH in CLIENT mode
   -P    set OH in PORTABLE mode
   -S    set OH in SERVER mode (portable)
   -l    set language -> [ ar de en es fr it pt sq ]
   -E    toggle EXPERT MODE - show advanced options
   -h    show help
   -q    quit

   -------------------------------
    EXPERT MODE - advanced options

   -A  toggle API server - EXPERIMENTAL | -d  toggle log level INFO/DEBUG
   -D  initialize OH with Demo data     | -X  clean/reset OH installation
   -e  export/save OH database          | -r  restore OH database
   -m  configure database connection manually   | -i  initialize/install OH databas
   -s  save OH configuration            | -t  test database connection (CLIENT mode
only)
   -G  setup GSM
   -U  enable UI web interface          | -u  create Desktop shortcut
   -v  show configuration               | -V  check for latest available OH version

 Please select an option or press enter to start OH:
```

- **C** set Open Hospital to start in CLIENT mode, usually when an external database server is used (Client / Server configuration)
- **P** set Open Hospital to start in PORTABLE mode, where data is saved locally
- **S** set Open Hospital to start in SERVER mode: the local portable instance of MariaDB is launched to act as a portable database server
- **l** set local language: en|fr|it|es|pt|ar
- **E** toggle EXPERT MODE: used to show advanced options and features. Use at your own risk!
- **h** show help
- **q** quit (windows only)

Advanced options

- **A** toggle API server: activate and start openhospital api jetty server - EXPERIMENTAL
- **e** export / save / dump the Open Hospital database in sql format
- **r** restore Open Hospital database from backup or external sql file: user will be prompted for input sql file
- **d** toggle log level between INFo and DEBUG - useful to execute OH in debug mode in order to log

errors or bugs with more extended informations to log file

- **G** setup GSM modem to enable sms interaction
- **D** initialize OH database with Demo data - loads a demo database in order to test the software
- **i** initialize / install OH database
- **m** configure OH database connection settings manually
- **s** save / write / generate OH configuration files (oh/rsc/*.properties) and exit
- **t** test database connection to the configured database server (Client mode only)
- **u** create Desktop shortcut with current params (Windows / Linux)
- **U** enable and start OH UI - Web interface
- **v** show Open Hospital external software version and configuration
- **V** check online for latest OH available version
- **X** clean/reset OH installation by deleting all data and configuration files → **use with caution** ←

Script configuration

Some advanced options can be configured manually by editing the scripts (oh.sh and oh.ps1 - do not modify oh.bat unless legacymode is used) and setting the specific script variables. This might also be useful to set different combinations of options (language, debug level, ...) for specific needs.

OH directory path

```
############### OH general configuration - change at your own risk :-) ###############
#
# -> OH_PATH is the directory where Open Hospital files are located
# OH_PATH="c:\Users\OH\OpenHospital\oh-1.15"
```

(Windows only) Enable interactive mode

```
# Interactive mode
# set INTERACTIVE_MODE to "off" to launch oh.ps1 without calling the user
# interaction menu (script_menu). Useful if automatic startup of OH is needed.
# In order to use this mode, setup all the OH configuration variables in the script
# or pass arguments via command line.
$script:INTERACTIVE_MODE="on"
```

Config file generation

It is possibile to set the WRITE_CONFIG_FILES option to "on" to regenerate the OH configuration files at startup (this is also possibile by selecting the **g** script option). The default is set to off, so the configuration files are not regenerated and overwritten at every startup. This is useful for production environment where the configuration is fixed.

```
# set WRITE_CONFIG_FILES=on "on" to force generation / overwriting of configuration
files:
# data/conf/my.cnf and oh/rsc/*.properties files will be regenerated from the original
.dist files
# with the settings defined in this script.
#
# Default is set to "off": configuration files will not be generated or overwritten if
already present.
#
WRITE_CONFIG_FILES="off" # linux
$script:WRITE_CONFIG_FILES="off" # windows
```

Distribution type - CLIENT | PORTABLE | SERVER

```
# set OH mode to PORTABLE | CLIENT | SERVER - default set to PORTABLE
#OH_MODE=PORTABLE
#$script:OH_MODE="PORTABLE" # windows
```

Interface and software language:

```
# Language setting - default set to en
OH_LANGUAGE_LIST=("ar" "de" "en" "es" "fr" "it" "pt" "sq")
OH_LANGUAGE_LIST_INFO=("Arabic" "German" "English" "Spanish" "French" "Italian"
"Portuguese" "Albanian")
$script:OH_LANGUAGE="en" # ar de en es fr it pt sq # windows
```

Single / multi user

```
# single / multi user - set "yes" for single user configuration
#OH_SINGLE_USER="no"
$script:OH_SINGLE_USER="no"
```

Log level / debug mode

```
# set log level to INFO | DEBUG - default set to INFO
LOG_LEVEL=INFO # linux
$script:LOG_LEVEL="INFO" # windows
```

Demo mode

```
# set DEMO_DATA to on to enable demo database loading - default set to off
# -> Warning -> __requires deletion of all portable data__
DEMO_DATA=off # linux
```

```
$script:DEMO_DATA="off" # windows
```

Enable system wide JAVA

```
# set JAVA_BIN
# Uncomment this if you want to use system wide JAVA
#JAVA_BIN=`which java` # linux
#$script:JAVA_BIN="C:\Program Files\JAVA\bin\java.exe" # windows
```

Database configuration

If a database server hostname/address is specified (other then localhost), OH can be started in CLIENT mode and used in a client/server / LAN environment.

```
###################### Database configuration ########################
DATABASE_SERVER=localhost
DATABASE_PORT=3306
DATABASE_ROOT_PW="xxxxxxxxxx"
DATABASE_NAME=oh
DATABASE_USER=isf
DATABASE_PASSWORD="xxxxx"
```

OH configuration

```
########################  OH configuration  ##########################
# path and directories
OH_DIR="oh"
OH_DOC_DIR="doc"
CONF_DIR="data/conf"
DATA_DIR="data/db"
PHOTO_DIR="data/photo"
BACKUP_DIR="data/dump"
LOG_DIR="data/log"
SQL_DIR="sql"
SQL_EXTRA_DIR="sql/extra"
TMP_DIR="tmp"

# imaging / dicom
DICOM_MAX_SIZE="4M"
DICOM_STORAGE="FileSystemDicomManager" # SqlDicomManager
DICOM_DIR="data/dicom_storage"

# logging
LOG_FILE="startup.log"
OH_LOG_FILE="openhospital.log"
API_LOG_FILE="api.log"
```

```
# SQL creation files
#DB_CREATE_SQL="create_all_en.sql" # default to create_all_en.sql
DB_DEMO="create_all_demo.sql"
```

Default directory structure

The scripts takes care of creating all the needed data directories and configuration files. Everything is also parametric and user adjustable in the scripts with variables (or via command line options). The default folder structure is now clean, simple and **common to all distros:**

```
/oh -> Open Hospital distribution
/sql -> containing the SQL creation scripts
/data/conf -> configuration files for database (MariaDB / MySQL)
```

Created at runtime:

```
/tmp
data/db
data/dicom_storage
data/dump
data/log
data/photo
```

External software package downloaded at first run:

```
Mariadb 10.x.x server
Java JRE, Zulu or OpenJDK distribution
```

Documentation

Administrator and User manuals are available in the **doc** folder. Online versions of the manuals can be found on the [Open Hospital website](https://www.open-hospital.org/documentation)

Other issues

If you experience problems in starting up the script, avoid long folder path and path with special characters / spaces in it.

Linux

- If you get one of these errors:

```
Error on creating OH Database error while loading shared libraries: libncurses.so.5.
```

```
Error: MySQL root password not set! Exiting
```

You have to install the ncurses librares, on Ubuntu:

```
sudo apt-get install libncurses5
```

- If you get this error:

```
Error Initializing MySQL database on port 3306 error while loading shared libraries:
libaio.so.1.
```

You have to install the libaio libraries, on Ubuntu:

```
sudo apt-get install libaio1
```

- If you select languages en-fr-it, a ICD10 patologies subset is loaded at startup, languages es-pt don’t.

Windows

Windows - create startup shortcut

Suggested method is to use the "u" script option to create a Windows Desktop shortcut for OH. Follow these instruction to create it manually:

Method 1 (with launch parameters configured in oh.ps1) - Rigth click on Desktop - New Shortcut - Browse to OH folder location and select oh.bat - Assign a name to the shortcut - Right click on the shortcut and select Properties - Change icon - Specify a different file - Browse to OH folder location and select oh.ico - Apply

Method 2 (with launch parameters stored on execution command) - Rigth click on Desktop - New Shortcut - Browse to OH folder location and select oh.ps1 - Assign a name to the shortcut - Right click on the shortcut and select Properties - Change icon - Specify a different file - Browse to OH folder location and select oh.ico - Modify Target with

```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -ExecutionPolicy Bypass
-File ./oh.ps1
```

- Apply Option parameters can be added at the end of Target string separated by spaces, example:

```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -ExecutionPolicy Bypass
-File ./oh.ps1 -loglevel DEBUG
```

Powershell configuration

Powershell minimum version 5.1 is required to run oh.ps1. To install Powershell 5.1 go to https://www.microsoft.com/en-us/download/details.aspx?id=54616

If you get this error:

```
+ CategoryInfo : NotSpecified: (:) [], PSSecurityException
+ FullyQualifiedErrorId : RuntimeException or UnauthorizedAccess
```

- Start Windows PowerShell with the "Run as Administrator" option. Only members of the Administrators group on the computer can change the execution policy. Enable running unsigned scripts by entering:

```
set-executionpolicy remotesigned
```

- You might also be required to enable access on Windows Firewall to oh.ps1 and/or to the TCP port used for the local database (PORTABLE mode).

Windows - legacy mode

It's also possible to start Open Hospital with the legacy batch file (old oh.bat behaviour): - open cmd.exe, browse to the OH installation directory and run **.\oh.bat -legacymode** - to see available options in legacymode, run **.\oh.bat -h**

(*) If you are using oh.bat in legacy mode, you might have to download and unzip java ad mysql manually.

Java

In order to download and unzip Java:

- Visit https://cdn.azul.com/zulu/bin/
- download the latest **JRE** for your architecture:

x64 - 64bit: https://cdn.azul.com/zulu/bin/zulu17.68.203-ca-fx-jre17.0.20.1-win_x64.zip

x86 - 32bit: https://cdn.azul.com/zulu/bin/zulu17.68.203-ca-fx-jre17.0.20.1-win_i686.zip

- unzip the downloaded file into the base directory where OpenHospital has been placed.

MariaDB

In order to download and unzip MariaDB:

- Visit https://mariadb.org/download/
- Select the Operating System: **Windows**
- Select package type: **ZIP file**

- Select CPU (architecture) **32 / 64**
- Download the zip file:

x86 - 32bit: https://archive.mariadb.org/mariadb-10.6.5/win32-packages/mariadb-10.6.5-win32.zip

x64 - 64bit: https://archive.mariadb.org/mariadb-10.6.28/winx64-packages/mariadb-10.6.28-winx64.zip

- unzip the downloaded file into the base directory where OpenHospital has been placed.

oh.sh / oh.ps1 - features and development

Check online documentation.

last updated: 2026.09.01

2.3 OH - SERVER mode (portable)

Open Hospital can be launched in SERVER (portable) mode. SERVER mode launches a local instance of the database server and listens for connections from other clients. This mode is useful to test/simulate a client/server configuration or networked environment; it is not suggested for production setup. To setup a complete client / server configuration, refer to chapter 2.1.3 - Database Server

2.4 Backup & Restore

Backup of the Open Hospital database can be performed in several ways.



The easiest way is to use the provided scripts: on Linux execute **oh.sh -e**, on Windows launch **oh.bat** and choose the option **e**.

Restoring the Open Hospital database can also be performed in several ways.



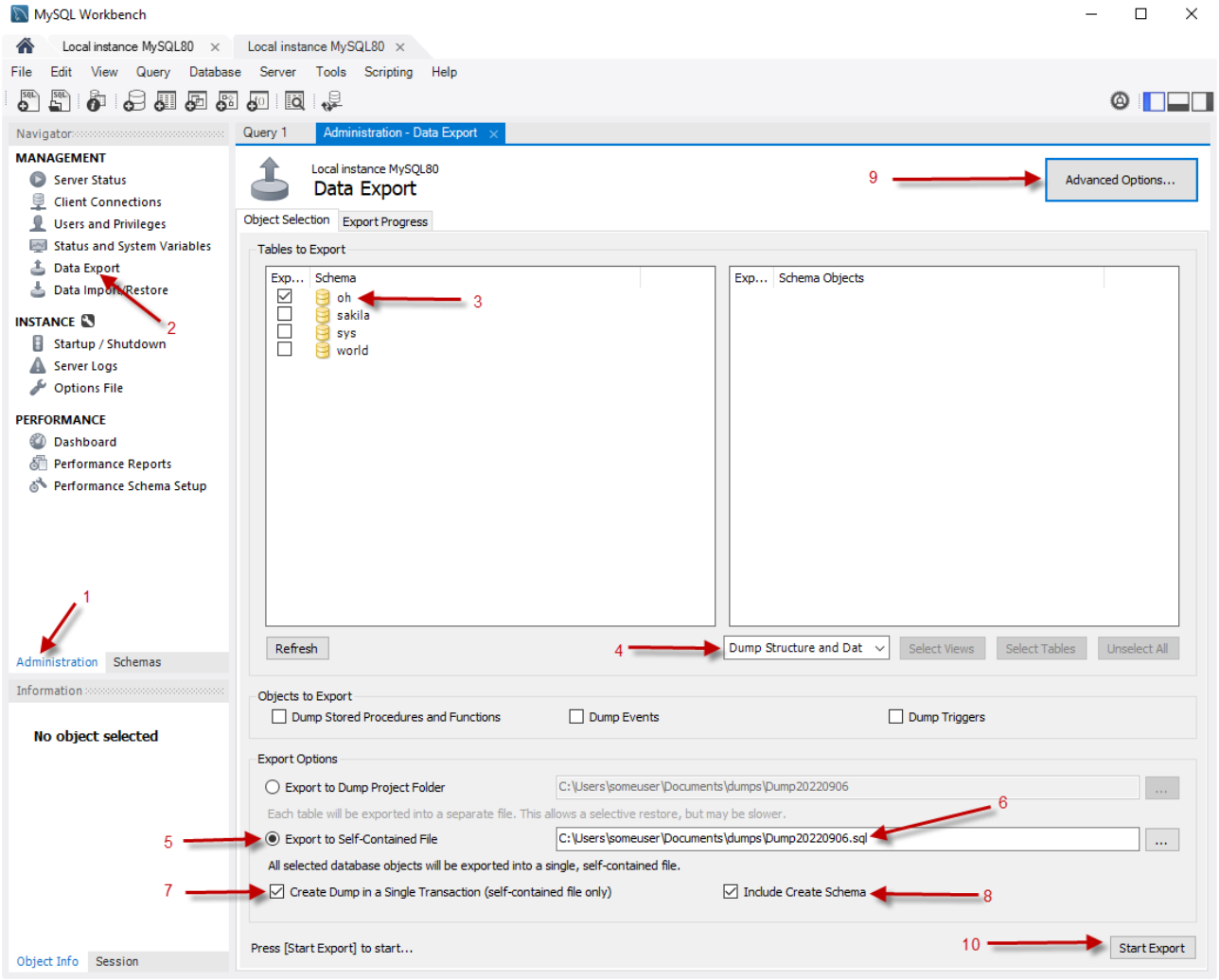
The easiest way is to use the provided scripts: on Linux execute **oh.sh -e**, on Windows launch **oh.bat** and choose the option **e**.

2.4.1 Backup of database (alternative with MySQL Workbench®)

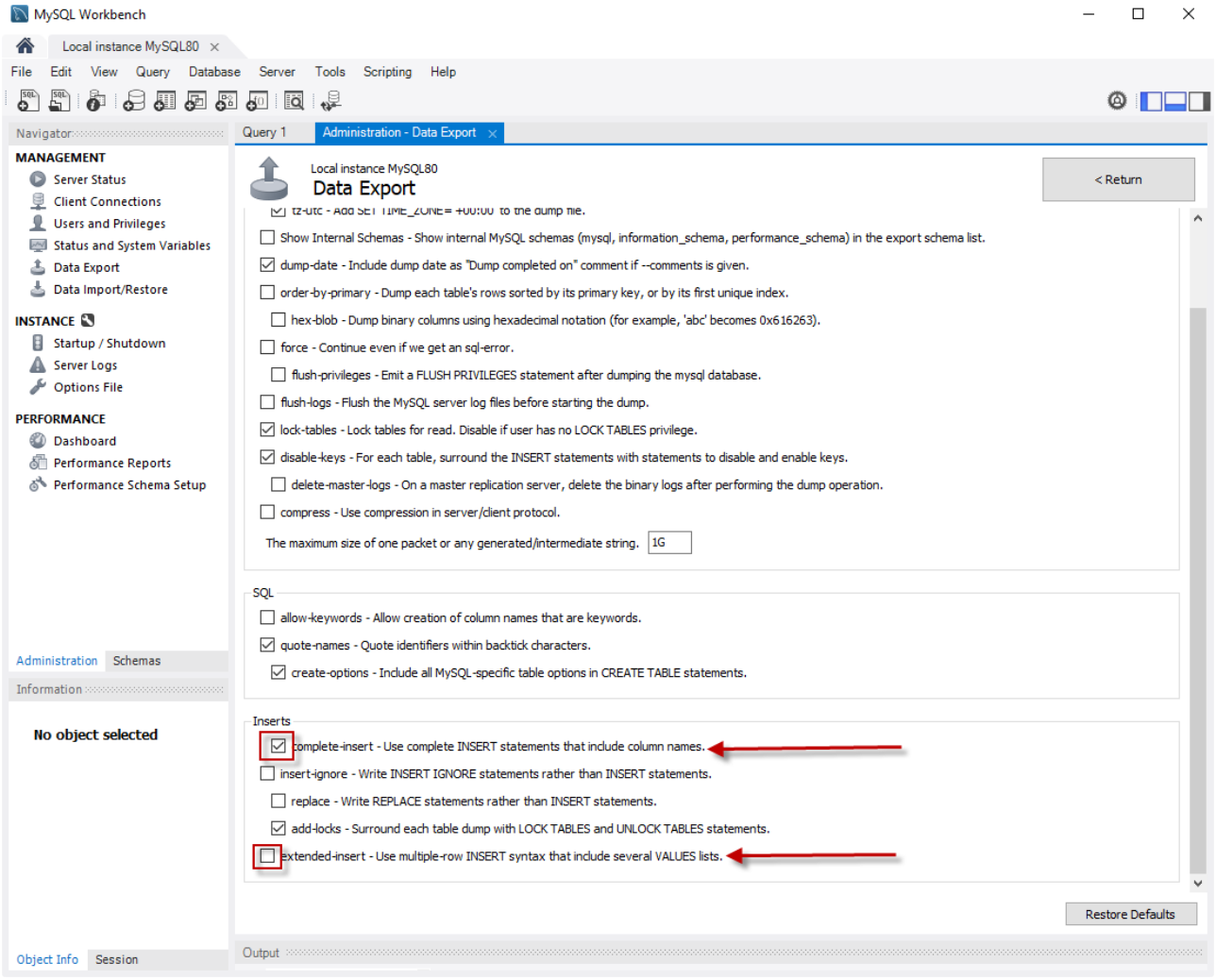
There are several ways to back up and restore **data** with MySQL. The recommended method is to use a common MySQL client such as MySQL Workbench®:

1. Click on the **Administration** tab on the left side of the Navigator Panel
2. Click on **Data Export** option
3. Select the **oh** schema
4. Select the required option from the dropdown below the tables list as required (the suggested option is **Dump Structure and Data**)

5. Click on **Export to Self-Contained File**
6. Adjust output file location as needed
7. Select **Create Dump in a Single Transaction** checkbox
8. Select **Include Create Schema** checkbox
9. Click on the **Advanced Options** button and scroll down to the **Inserts** panel (see the second image below):
    a. Select **complete-insert** checkbox
    b. Uncheck **extended-insert** checkbox
    c. Return to the main panel
10. Select the **Start Export** button



Advanced Options image:



2.4.2 Restore of database (alternative with MySQL Workbench®)

To restore database data with MySQL Workbench®:

1. Under **Server Administration** on the Home window select the server instance onto which the data will be restored (Create **New Server Instance** if doing it the first time).
2. Click on **Manage Import/Export**
3. Click on **Data Import/Restore** on the left side of the screen.
4. Select **Import from Self-Contained File** radio button (right side of the screen)
5. Select the path and the name of the restore sql file.
6. Click the **Start Import** button at the right bottom corner of the window.

2.4.3 Backup of OH code and settings

Backup of the Open Hospital **code and settings** can be performed by saving the **"oh"** folder (configuration files are found under the **"oh/rsc"** folder).

Backup of the client **data files**, useful for "PORTABLE" mode, can be performed by saving/copying the **"data"** folder (including all subfolders) found under the main OH installation path

2.5 Folders and directory structure

Open Hospital has a folder tree hierarchy which is shown here, where <***version***> is replaced with the current Open Hospital version, <***operating_system***> with windows or linux and <***arch***> with the OS architecture (32 or 64bit):

- **OpenHospital-<*version*>-<operating_system>_<arch>-<client|portable>**/
- **bundle** – language files
- **doc** – Open Hospital documentation
- **lib** – Java libraries needed to run the software
- **oh** – Open Hospital application
- **oh/bin** – Open Hospital binaries (compiled software)
- **oh/rpt_base** – JasperReports® reports used by Open Hospital
- **oh/rpt_base/<lang>** – Language-specific translation files for reports in oh/rpt_base
- **oh/rpt_base/PDF** – Reports defined in oh/rpt_base generated by Open Hospital in PDF format
- **oh/rpt_stat** – JasperReports® reports used by Open Hospital in section Statistics
- **oh/rpt_stat/<lang>** – Language-specific translation files for reports in oh/rpt_stat
- **oh/rpt_stat/PDF** – Reports defined in oh/rpt_stat generated by Open Hospital in PDF format
- **oh/rpt_extra** – JasperReports® reports defined by the user
- **oh/rpt_extra/<lang>** – Language-specific translation files for reports in oh/rpt_extra
- **oh/rpt_extra/PDF** – Reports defined in oh/rpt_extra generated in PDF format
- **oh/rsc** – **resources** of Open Hospital, as configuration files and other related files
- **oh/rsc/icons** – icons used in Open Hospital for windows and buttons
- **oh/rsc/images** – images used in Open Hospital
- **data** – Open Hospital local data and configuration
- **data/conf** – Database server configuration files
- **data/db** – Database files (PORTABLE mode)
- **data/logs** – Log files
- **data/dicom** – Image files
- **data/dump** – Database dumps/backups
- **data/photo** – Patients photos
- **sql** – Database creation scripts
- **sql/extra** – SQL scripts for custom / experimental / temporary features
- **tmp** – Temporary files

These folders may be organized differently depending on the version of the software or the architecture of the operating system.

Depending on the chosen operating system and architecture, the needed external applications and libraries (like MySQL/MariaDB and Java JRE) will be present also:

- *zulu11.<version>-<operating_system><arch>/* - Java Virtual Machine
- *mariadb-<version>-<arch>/ or _mysql-<version>-<arch>-<build>/* - MySQL Server or MariaDB Server

2.6 Appearance

Open Hospital's main menu appearance can be customized by adding a logo of choice. To test the feature, rename the image file *logo_hospital_example.png* found in the *oh/rsc/images* folder to *logo_hospital.png*:



The custom logo has to be a 100x100 pixel png image file.

3 Configuration

Configuration of the local Open Hospital instance (CLIENT and PORTABLE mode) is set in the configuration files called "properties files". The following properties files are found in the **oh/rsc** subfolder:

- **database.properties** – Open Hospital database connection
- **dicom.properties** – Imaging viewer module options
- **examination.properties** – contains the settings for the 'examination' module
- **log4j2-spring.properties** – Logging system and paths

- **settings.properties** – Open Hospital options and settings
- **sms.properties** – SMS Manager module
- **telemetry.properties** – Telemetry module settings
- **txtPrinter.properties** – Text printing system
- **xmpp.properties** – Xmpp Server configuration
- **default_credentials.properties** – contains the default credentials

Furthermore, additional configurations files (not meant to be modified by the user) are:

- **resolutions.xml** – stores locally the information about webcams
- **version.properties** – contains the current Open Hospital version

The properties files are related only to the local instance of Open Hospital; different instances may use specific OH configurations with custom properties files. The properties files can be generated automatically from the packaged *.dist* files by editing the configuration options available in the startup scripts **oh.sh** / **oh.ps1** and launching OH. The properties files can also be modified manually using a text editor.



Any change to these files requires an application restart to apply the modified settings.



The properties files can be automatically generated/overwritten at any Open Hospital startup, by setting the WRITE_CONFIG_FILES option to "on" in the **oh.sh** / **oh.ps1** scripts.

The following chapters will describe these properties files and their features in detail.

3.1 settings.properties

Open Hospital general configuration is set in the *settings.properties* file. The default configuration is available in the *settings.properties.dist* file:

```
##########################################
# This file contains Open Hospital settings
##########################################
# external settings
MODE=OH_MODE
DEMODATA=off

# experimental settings
APISERVER=off

# internal settings, modules, directories
LANGUAGE=OH_LANGUAGE
SINGLEUSER=YES_OR_NO
DEBUG=no
```

```
DOC_DIR=OH_DOC_DIR
PATIENTPHOTOSTORAGE=PHOTO_DIR
INTERNALVIEWER=yes
SMSENABLED=no
VIDEOMODULEENABLED=yes
XMPPMODULEENABLED=no
PARAMSURL=https://conf.open-hospital.org/oh-conf.json

# application settings
ENHANCEDSEARCH=no
INTERNALPHARMACIES=yes
LABEXTENDED=yes
LABMULTIPLEINSERT=yes
MATERNITYRESTARTINJUNE=no
MERGEFUNCTION=yes
OPDEXTENDED=yes
PATIENTEXTENDED=yes
PATIENTVACCINEEXTENDED=yes

# GUI settings
MAINMENUALWAYSONTOP=no

# accounting
ALLOWMULTIPLEOPENEDBILL=yes
ALLOWPRINTOPENEDBILL=yes
BILLSREPORT=BillsReport
BILLSREPORTMONTHLY=BillsReportMonthly
BILLSREPORTPENDING=BillsReportPending
PATIENTBILL=PatientBill
PATIENTBILLGROUPED=PatientBillGrouped
PATIENTBILLSTATEMENT=PatientBillStatement
RECEIPTPRINTER=yes

# pharmacy
AUTOMATICLOT_IN=no
AUTOMATICLOT_OUT=no
AUTOMATICLOTWARD_TOWARD=no
LOTWITHCOST=yes
PHARMACEUTICALORDER=PharmaceuticalOrder
PHARMACEUTICALSTOCK=PharmaceuticalStock_ver4
PHARMACEUTICALSTOCKLOT=PharmaceuticalStock_ver5
PHARMACEUTICALAMC=PharmaceuticalAMC

# dicom / imaging settings
DICOMMODULEENABLED=yes
DICOMTHUMBNAILS=yes

# reports
ADMCHART=patient_adm_chart
DISCHART=patient_dis_chart
EXAMINATIONCHART=patient_examination
```

```
OPDCHART=patient_opd_chart
PATIENTSHEET=patient_clinical_sheet_ver3
VISITSHEET=WardVisits

# security
SESSIONTIMEOUT=5
STRONGPASSWORD=yes
STRONGLENGTH=6
USERSLISTLOGIN=no
PASSWORDTRIES=5
PASSWORDLOCKTIME=60
PASSWORDIDLE=365

# telemetry
TELEMETRYENABLED=yes
```

Every line is composed of a KEY and a value:

KEY=value

Values can be:

- **Boolean**: yes | no or true | false
- **String**: usually a filename or a country code (ISO 3166-1)

Use the provided startup scripts in order to automatically generate the *settings.properties* file from the corresponding *.dist* file; after generation, optionally adjust the KEY/value pair as explained in the following sections.

3.1.1 LANGUAGE

The following table shows the allowed values for the OH_LANGUAGE variable:

Key	Default Value	Allowed Values
LANGUAGE	en	ar, de, en, es, fr, it, pt, sw, zh_CN

Open Hospital is available in nine different languages, identified by the international country code:

- ar – Arabic
- de - German
- en – English
- es – Spanish
- fr – French
- it – Italian
- pt – Portuguese
- sw – Swahili

- zh_CN - simplified Chinese

To change the language used in the application edit *settings.properties* and change the value of this key. If an unknown value is set, the local computer language is applied.



Report translation files are stored in language subfolders under `rpt_base`, `rpt_stat`, and `rpt_extra` (e.g. `rpt_base/es/`). Installations upgrading from a previous version should migrate their translation files from the legacy naming convention (e.g. `PatientBill_es.properties`) to the new subfolder structure. A temporary fallback for the legacy structure is available but will be removed in a future version.



An application restart is required to apply the modified setting.



The language (OH_LANGUAGE) can be automatically set by using the provided **oh.sh** / **oh.ps1** scripts.

3.1.2 SINGLEUSER

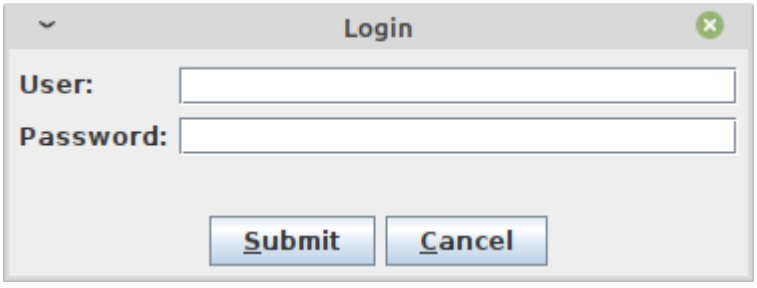
The following table shows the default value and the allowed ones:

Key	Default Value	Allowed Values
SINGLEUSER	yes	yes, no

Open Hospital is designed to support multi-user usage. This means that it is possible to define different users, arrange them by group and set different rights in application usage; for instance, a user “receptionist” may be able to register a new patient but not to prescribe a therapy.

Open Hospital has a predefined user called “admin” that has access rights to all the features of the application. When SINGLEUSER is set to YES (default) the program starts with this user and all the features are available.

If SINGLEUSER is set to NO the following login window appears when the program is started:





See 3.1.36 USERSLISTLOGIN in this document to change the login mode.

The default password for the “admin” user is “admin”. After typing this password in the white field and pressing **Submit**, the main ***Menu*** with the functions are enabled.

Once logged in as the “admin” user defining additional users and/or groups is possible (see the Users & Groups chapter in the User’s Guide), or just continue with a simple form of data protection.



The setting is client-side specific, this means that the login can be disabled on a specific client, but logons will still be required on other clients, with the same defined user and group.



Data security must never be left solely to the application but it must include proper network architecture and a rigid configuration of the clients.

3.1.3 AUTOMATICLOT

The following table shows the default value and the alternatives for lot management:

Key	Default Value	Valid Values
AUTOMATICLOT_IN	no	yes, no
AUTOMATICLOT_OUT	no	yes, no
AUTOMATICLOTWARD_TOWARD	no	yes, no

Open Hospital allows for automatic management of lots in the main pharmacy. This means that it is possible to work in the pharmacy without taking care of lot definitions in loading the store and just specify preparation and expiring date for each charging movement (see 4.2.2.2 Insert stock charging movement in the User's Guide).

For discharging movement, the lot is automatically calculated according to the nearest expiring date of lots (FEFO - First Expiring First Out). If the quantity to discharge is larger than the quantity of the first selected lot, more than one discharging operation can be generated.

AUTOMATIC LOTS MANAGEMENT is split into two different settings for charging and discharging the main pharmacy and managing lots also in the wards. In this way, more control is given to the user that may want to provide all the lots' details in *charging operations* but ask the application to automatically *discharge* the nearest expiring ones (FEFO). Similarly, in wards (see 4.3 Pharmaceuticals Stock Ward in the User's Guide) the user may decide to manage lots manually or to ask the application to do it automatically.

With **AUTOMATICLOT_IN** = **YES** the application generates a lot number automatically and only asks for an expiration date (mandatory). By default, AUTOMATICLOT_IN is set to NO. It is possible to change the value at any time.

With **AUTOMATICLOT_OUT** = **YES** the application discharges automatically the nearest expiring lot when required. By default, AUTOMATICLOT_OUT is set to NO. It is possible to change the value at any time.

With **AUTOMATICLOTWARD_TOWARD** = **YES**, the application discharges from a ward (to other wards) the nearest expiring lot automatically when required. By default, AUTOMATICLOTWARD_TOWARD is set to NO. It is possible to change the value at any time. For discharges to patients (drug giving) the lot is **never** asked.



An application restart is required to apply the modified settings.

3.1.4 LOTWITHCOST

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
LOTWITHCOST	yes	yes, no

Open Hospital allows for managing the cost of medicals in the main pharmacy (see 4.2.2.2 Insert stock charging movement in the User's Guide).



An application restart is required to apply the modified setting.

3.1.5 PATIENTSHEET

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
PATIENTSHEET	patient_clinical_sheet	any kind of .jasper file name

Open Hospital can produce a report about the clinical history of a patient (see 8.9 Clinical Sheet in the User's Guide).

By default, PATIENTSHEET is set to patient_clinical_sheet, that is, the filename of the related report to use for the Clinical Sheet functionality. It is possible to use a different report by installing it in the report folder (see Reports) and by changing this parameter.



An application restart is required to apply the modified setting.

3.1.6 OPDCHART

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
OPDCHART	patient_opd_chart	any kind of .jasper file name

Open Hospital can produce a report about the OPD chart of a patient (see 8.9 Clinical Sheet in the User's Guide).

By default, OPDCHART is set to patient_opd_chart which is the filename of the related report to use for the Clinical Sheet functionality **OPD Chart**. It is possible to use a different report by installing it in the report folder (see Reports) and by changing this parameter.



An application restart is required to apply the modified setting.

3.1.7 ADMCHART

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
ADMCHART	patient_adm_chart	any kind of .jasper file name

Open Hospital can produce a report about the Admission of a patient (see 8.9 Clinical Sheet in the User's Guide).

By default, ADMCHART is set to patient_adm_chart, that is, the filename of the related report to use for the Clinical Sheet functionality **Admission Chart**. It is possible to use a different report by installing it in the report folder (see Reports) and by changing this parameter.



An application restart is required to apply the modified setting.

3.1.8 DISCHART

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
DISCHART	patient_dis_chart	any kind of .jasper file name

Open Hospital can produce a report about the Discharge of a patient (see 8.9 Clinical Sheet in the User's Guide).

By default, DISCHART is set to patient_dis_chart which is the filename of the related report to use for the Clinical Sheet functionality **Discharge Chart**. It is possible to use a different report by installing it in the report folder (see Reports) and by changing this parameter.



An application restart is required to apply the modified setting.

3.1.9 PATIENTBILL

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
PATIENTBILL	PatientBill	any kind of .jasper file name

Open Hospital can manage patient bills and produce an A4 format breakdown with his/her items and payments (see 6.2 Functions of Accounting in the User's Guide).

By default, PATIENTBILL is set to PatientBill, that is, the filename of the related report to use for the bill printing functionality. It is possible to use a different report by installing it in the report folder (see Reports) and by changing this parameter.



An application restart is required to apply the modified setting.

3.1.10 BILLSREPORT

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
BILLSREPORT	BillsReport	any kind of .jasper file name

Open Hospital can manage patient bills and produce a report about all bills paid (or not paid) within a span of time (see **8.2 Functions of Accounting** in the *User's Guide*).

By default, BILLSREPORT is set to BillsReport, that is, the filename of the related report to use for the account printing functionality. It is possible to use a different report by installing it in the report folder (see Reports) and by changing this parameter.



An application restart is required to apply the modified setting.

3.1.11 BILLSREPORTMONTH

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
BILLSREPORTMONTH	BillsReportMonth	any kind of .jasper file name

Open Hospital can manage patient bills and produce a report about all bills paid (or not paid) monthly (see 6.2 Functions of Accounting in the User's Guide).

By default, BILLSREPORTMONTH is set to BillsReportMonth, that is, the filename of the related report to use for the account printing functionality. It is possible to use a different report by installing it in the report folder (see Reports) and by changing this parameter.



An application restart is required to apply the modified setting.

3.1.12 PHARMACEUTICALORDER

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
PHARMACEUTICALORDER	PharmaceuticalOrder	any kind of .jasper file name

Open Hospital can produce a report for the Pharmacy Stock critical levels which helps to identify which pharmaceuticals are running low and which ones need to be ordered (see 4.1 Pharmaceuticals in the User's Guide).

By default, PHARMACEUTICALORDER is set to PharmaceuticalOrder, which is the filename of the related report to use for the pharmacy printing functionality. It is possible to use a different report by installing it in the report folder (see Reports) and by changing this parameter.



An application restart is required to apply the modified setting.

3.1.13 PHARMACEUTICALSTOCK

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
PHARMACEUTICALSTOCK	PharmaceuticalStock	any kind of .jasper file name

Open Hospital can produce a report for the Pharmacy Stock Status which identifies which pharmaceuticals are currently available (see **6.1 Pharmaceuticals** in the *User's Guide*).

By default, PHARMACEUTICALSTOCK is set to PharmaceuticalStock which is the filename of the related report to use for the pharmacy printing functionality. It is possible to use a different report by installing it in the report folder (see Reports) and by changing this parameter.



An application restart is required to apply the modified setting.

3.1.14 PATIENTEXTENDED

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
PATIENTEXTENDED	yes	yes, no

Open Hospital allows for the registration of a patient with extended or more detailed information (see 8.4 Insert a new Patient Extended in the User's Guide).

By default, PATIENTEXTENDED is set to yes. It is possible to set it to no to reduce the amount of data collected and so reduce the workload for the staff involved in data entry.



An application restart is required to apply the modified setting.

3.1.15 OPDEXTENDED

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
OPDEXTENDED	yes	yes, no

Open Hospital can link every OPD visit to a registered patient to generate a comprehensive clinical history. For each visit, the user will be asked to select a registered patient or to register a new one, so each visit is attached to the patient's history (see **5.3 OPD Extended** and **10.9 Clinical Sheet** in the *User's Guide*).

By default, OPDEXTENDED is set to yes. It is possible to set it to no to reduce the amount of data to be collected (only age and sex, no patient registration) and so reduce the workload for the staff involved in data entry.



An application restart is required to apply the modified setting.

3.1.16 MATERNITYRESTARTINJUNE

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
MATERNITYRESTARTINJUNE	no	yes, no

Open Hospital allows changing the way the admissions in the Maternity ward are counted within the year; in Open Hospital the first admission of the year for every ward has progressive number 1 (one) and it increments itself automatically up to the end of the year; in some facilities, this is partially true and the progressive numbering starts from June only for the Maternity ward. This option controls which behavior is followed.

By default, MATERNITYRESTARTINJUNE is set to no.



An application restart is required to apply the modified setting.

3.1.17 LABEXTENDED

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
LABEXTENDED	yes	yes, no

Open Hospital can force every laboratory exam to be linked to a registered patient, to have a comprehensive clinical history. For each exam, the user will be asked to select a registered patient so from that moment the exam will be attached to the patient's history (see 5.3.2 New Laboratory Exam in the User's Guide).

By default, LABEXTENDED is set to yes. Anyway, is possible to set it to no to reduce the amount of data to be collected (only name, age, sex, no patient registration) and so reduce the workload for the staff involved in data entry.



An application restart is required to apply the modified setting.

3.1.18 LABMULTIPLEINSERT

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
LABMULTIPLEINSERT	yes	yes, no

Open Hospital can insert multiple laboratory tests per patient at one time to avoid the repetitive operation of selecting a patient or writing his/her name; note that the LABEXTENDED option must also be set to yes (see 5.3.3 Laboratory Multiple Insert in the User's Guide).

By default, LABMULTIPLEINSERT is set to yes, but if LABEXTENDED is set to no it will be just ignored.



An application restart is required to apply the modified setting.

3.1.19 INTERNALPHARMACIES

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
INTERNALPHARMACIES	yes	yes, no

Open Hospital can register all dispensing to patients within a ward. Activating this option, the Pharmaceutical Stock Ward functionality will be available in the application.

By default, INTERNALPHARMACIES is set to yes.



An application restart is required to apply the modified setting.

3.1.20 MERGEFUNCTION

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
MERGEFUNCTION	no	yes, no

Open Hospital can merge two patients' histories into one. This is particularly useful in the case of double registration where different information was collected over time. Activating this option, the Merge functionality will be available in the Admission/Patient module (see Merge function in the User's Guide).

By default, MERGEFUNCTION is set to no.



An application restart is required to apply the modified setting.

3.1.21 INTERNALVIEWER

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
INTERNALVIEWER	yes	yes, any executable filename in the path or with an absolute path

Open Hospital can use a different PDF reader for generated reports. If this parameter is set to yes, the Jasper PDF Viewer is used (see Report Launcher in the User's Guide). If another PDF viewer is required, specify the executable's filename; the executable must be available in the desktop

environment.

Example in Windows:

```
INTERNALVIEWER=AcroRd32.exe

INTERNALVIEWER=C:\\Program Files (x86)\\Adobe\\Reader 10.0\\Reader\\AcroRd32.exe
```



Please note the double file and folder separator "\|".

Example in Linux:

```
INTERNALVIEWER=evince
```

By default, INTERNALVIEWER is set to yes.



An application restart is required to apply the modified setting.

3.1.22 DOC_DIR

The following table shows the allowed value for the DOC_DIR variable:

Key	Default Value	Valid Values
DOC_DIR	doc	any path

Open Hospital documentation is available online; PDF versions of the manuals are packaged with every OH release in a folder that must be accessible by the application; this folder might change depending on the application version, or eventually be customized by the administrator (e.g., subfolder on the Desktop)

Example in Windows:

```
DOC_DIR=doc
DOC_DIR=C:\\Users\\user\\OneDrive\\Desktop\\doc
```



Please note the double file and folder separator "\|".

Example in Linux:

```
DOC_DIR=doc
DOC_DIR=../doc
```

By default, DOC_DIR is set to 'doc'.



An application restart is required to apply the modified setting.



The DOC_DIR setting can be automatically generated/overwritten at any Open Hospital startup, by setting the WRITE_CONFIG_FILES option to "on" in the **oh.sh** / **oh.ps1** scripts.

3.1.23 SMSENABLED

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
SMSENABLED	no	yes, no

Open Hospital can indicate whether information can be sent to patients via SMS notifications. Enabling or disabling this option only affects the possibility to set, or not set, a therapy as notifiable (see 8.10.7 Notify & SMS in the User's Guide).

By default, SMSENABLED is set to no.

3.1.23.1 GSM Configuration

To set up a GSM device, the GSM mode has to be defined in the *sms.properties* file (see sms.properties).

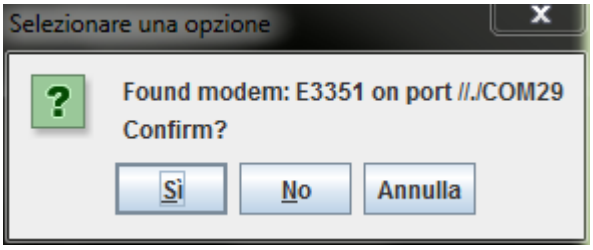
The default settings for a GSM modem should work with the majority of GSM devices (like phones, smartphones, USB modems, etc...). The exception is the PORT parameter which must be changed to match the SERIAL port address used by the device plugged into the system.

If the PORT param is correct but Open Hospital is not able to communicate with the device (try switching to DEBUG log level – or lower - during this setup – see log4j2-spring.properties) and try to use the SetupGSM utility as explained in the next chapter.

3.1.23.2 SetupGSM utility

To setup GSM communication it is possible to use the included SetupGSM utility and follow these instructions:

1. Plug the device into the system and make sure the system recognizes it, loads a proper driver for it and assigns a serial port (COM)
2. Launch the OH startup script and select the G option:
   - On Windows launch **oh.bat** → select **G** option
   - On Linux launch **oh.sh -G**
3. Once the utility has started, it will scan all plugged devices and will try to recognize the modem (or phone) within them
4. If the device is recognized as a modem, a confirmation message prompt is shown:



4. If the identified device is the correct one, just click “yes”, otherwise click “no”, and the scan will continue
5. Once “yes” is selected the *sms.properties* file is automatically modified by the utility by inserting the proper port address.

3.1.24 MAINMENUALWAYSONTOP

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
MAINMENUALWAYSONTOP	no	yes, no

Open Hospital can keep the main menu always on top so it cannot be overlapped or hidden by other windows.

By default, MAINMENUALWAYSONTOP is set to no.

An application restart is required to apply the modified setting.

3.1.25 RECEIPTPRINTER

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
RECEIPTPRINTER	no	yes, no

Open Hospital can manage text or ZPL printers with aim of printing simple receipts in the Accounting module (see 6.2.1.11 Print receipt function in User’s Guide). Once this option is activated, additional parameters are needed in the **txtPrinter.properties file** to adjust the output with the printer connected to the system (see txtPrinter.properties in this document).

By default, RECEIPTPRINTER is set to no.

The text printer must be set as the default printer.

An application restart is required to apply the modified setting.

3.1.26 VIDEOMODULEENABLED

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
VIDEOMODULEENABLED	no	yes, no

Open Hospital can drive any webcam with the aim of capturing patient images in the Admission/Patient module (see 8.4.3 Patient Photo function in User's Guide). Once this option is activated the related button, **New Photo** in the **New Patient** window is available to start and use the webcam.

By default, VIDEOMODULEENABLED is set to no.



An application restart is required to apply the modified setting.

3.1.27 PATIENTVACCINEEXTENDED

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
PATIENTVACCINEEXTENDED	yes	yes, no

Open Hospital can register vaccines given to patients (see 7 Vaccines in the User's Guide). This option allows toggling the patient's full name in the ***Patient Vaccine Browser*** window.

By default, PATIENTVACCINEEXTENDED is set to yes.



An application restart is required to apply the modified setting.

3.1.28 ENHANCEDSEARCH

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
ENHANCEDSEARCH	no	yes, no

Open Hospital can optimize memory usage when the number of registered patients becomes large (see 8.2.2 Search patient Enhanced in the User's Guide). With the enhanced search only, the patient matching a search criterion will be loaded in the memory, otherwise, all patients registered in the system will be loaded in the ***Patient Browser*** window.

By default, ENHANCEDSEARCH is set to no. It is possible to toggle this option anytime to reduce the amount of memory needed by the computer or the server.



An application restart is required to apply the modified setting.

3.1.29 XMPPMODULEENABLED

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
XMPPMODULEENABLED	no	yes, no

Open Hospital embeds a technology that allows users logged into the application to chat and share information related to the hospital activities (see 11 Communication in the User's Guide); the SINGLEUSER option must be set to "no". If this option is active, Open Hospital will look for an XMPP Server at startup time.

See the xmpp.properties chapter for instructions on how to setup and configure the XMPP Server communication.

By default, XMPPMODULEENABLED is set to no, but if SINGLEUSER is set to yes XMPPMODULEENABLED is ignored.



An application restart is required to apply the modified setting.

3.1.30 DICOMMODULEENABLED

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
DICOMMODULEENABLED	no	yes, no

Open Hospital includes a feature that allows attaching image files to a patient's clinical sheet. When this option is enabled, an **Imaging** button is shown in the patient Clinical sheet module.

By default, DICOMMODULEENABLED is set to no.



An application restart is required to apply the modified setting.

3.1.31 DICOMTHUMBNAILS

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
DICOMTHUMBNAILS	yes	yes, no

Open Hospital can enable or disable thumbnails in the Imaging Viewer.

By default, DICOMTHUMBNAILS is set to yes.



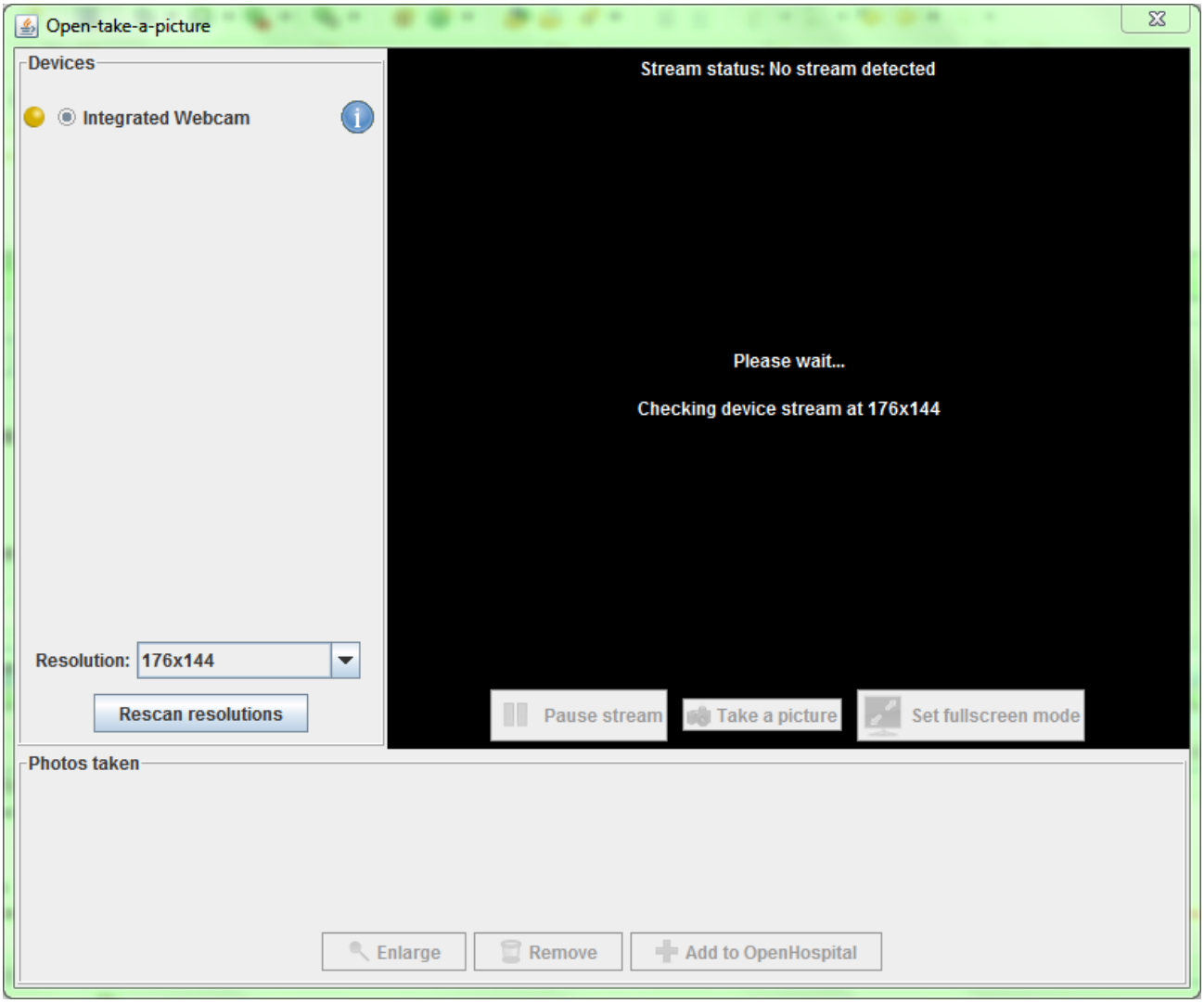
An application restart is required to apply the modified setting.

3.1.32 DEBUG

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
DEBUG	no	yes, no

Open Hospital can run in debug mode which is useful to understand what is happening behind the user graphical interface. Currently, this option affects only the video module, which means that in place of the normal **New Patient** window (see 8.4.3 Patient Photo function in User's Guide), the program opens the new special window shown below:



From this window it is possible to have more information about webcams connected to the system. It is possible to apply different resolutions in order to find the best setting or get information about a problem.

Once a photo is produced, Open Hospital remembers the settings and uses them in the future.

By default, DEBUG is set to no.



An application restart is required to apply the modified setting.

3.1.33 ALLOWMULTIPLEOPENEDBILL

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
ALLOWMULTIPLEOPENEDBILL	no	yes, no

Open Hospital allows a patient to have multiple open bills (invoices). If this option is enabled, when creating a new bill for a patient if that patient already has an open bill the user is asked to confirm the creation of another one.

By default, ALLOWMULTIPLEOPENEDBILL is set to no.



An application restart is required to apply the modified setting.

3.1.34 OPENEDBILLSREPORT

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
OPENEDBILLSREPORT	OH023_BillsReportMonth	OH023_BillsReportMonth

Open Hospital can print the list of open bills. The parameter contains the Jasper report file name to print the list of open bills.

By default, OPENEDBILLSREPORT is set to OH023_BillsReportMonth.



An application restart is required to apply the modified setting.

3.1.35 ALLOWPRINTOPENEDBILL

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
ALLOWPRINTOPENEDBILL	no	yes, no

Open Hospital can print the receipt of a single open invoice. When the parameter is enabled, in the Accounting module under the pending tab one can select an open bill and get the receipt by clicking the Receipt button in the buttons panel.

By default, ALLOWPRINTOPENEDBILL is set to no.



An application restart is required to apply the modified setting.

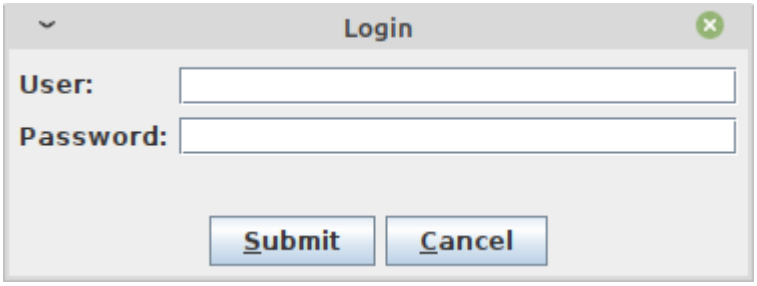
3.1.36 USERSLISTLOGIN

The following table shows the default value and the allowed ones:

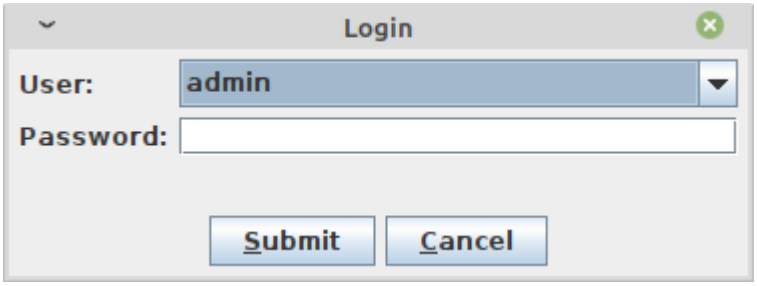
Key	Default Value	Allowed Values
USERSLISTLOGIN	no	yes, no

When Open Hospital is used in multi-user mode, i.e. SINGLEUSER = no, it is possible to choose between two different login modes:

(1) by typing the username directly in a textbox, if USERSLISTLOGIN is set to NO (default):



(2) by selecting the username from a list of users, if USERSLISTLOGIN is set to YES:



By default, USERSLISTLOGIN is set to no.



An application restart is required to apply the modified setting.

3.1.37 STRONGPASSWORD

The following table shows the default value and the allowed ones:

Key	Default Value	Allowed Values
STRONGPASSWORD	yes	yes, no

By default, the value of STRONGPASSWORD is set to yes indicating that the password must contain at least one alphabetic, numeric, and special character. The list of recognized special characters is: **_$&+,:;=\?@#|/'<>.^()%!-***



An application restart is required to apply the modified setting.

3.1.38 STRONGLENGTH

The following table shows the default value and the allowed ones:

Key	Default Value	Allowed Values
STRONGLENGTH	6	0 or a positive integer value

The value of STRONGLENGTH is the minimum length of a user's password. To disable the checking of the password's length set STRONGLENGTH to 0.



If STRONGPASSWORD is set to **yes**, then STRONGLENGTH should be set to a value greater than or equal to 3 as that is the minimum number of unique characters required.



An application restart is required to apply the modified setting.

3.1.39 PASSWORDTRIES

The following table shows the default value and the allowed ones:

Key	Default Value	Allowed Values
PASSWORDTRIES	5	0 or a positive integer value greater than or equal to 5

The value of PASSWORDTRIES is the maximum number of consecutive failed passwords when attempting to log in before the account is locked. To disable the checking of the consecutive failed attempts set PASSWORDTRIES to 0.



An application restart is required to apply the modified setting.

3.1.40 PASSWORDLOCKTIME

The following table shows the default value and the allowed ones:

Key	Default Value	Allowed Values
PASSWORDLOCKTIME	60	a positive integer value greater than or equal to 60; the unit of time is minutes

The value of PASSWORDLOCKTIME is the number of minutes that an account is locked after failing to specify the correct password for the number of times specified by the PASSWORDTRIES variable described above.



An application restart is required to apply the modified setting.

3.1.40 PASSWORDIDLE

The following table shows the default value and the allowed ones:

Key	Default Value	Allowed Values
PASSWORDIDLE	365	a positive integer value greater than or equal to 0; the unit of time is days

The value of PASSWORDIDLE specifies the maximum number of days that an account can be unused before being locked and requiring intervention by the administrator. To disable this check

set PASSWORDIDLE to 0.



An application restart is required to apply the modified setting.

3.1.41 PATIENTPHOTOSTORAGE

The following table shows the default value and the allowed ones:

Key	Default Value	Valid Values
PATIENTPHOTOSTORAGE	DB	DB, <*path_to_folder*>

Open Hospital can save the patient's profile picture in the database or on the file system.



The maximum size for a patient's profile picture is limited to 32,768 bytes. If a bigger image is provided, it will automatically be resized.

If the PATIENTPHOTOSTORAGE parameter is set to a path (that is, not "DB"), then Open Hospital looks for or saves the patient's profile picture in the specified path.

If the PATIENTPHOTOSTORAGE parameter is set to "DB", then Open Hospital looks for or saves the patient's profile picture in the Open Hospital database.

By default, PATIENTPHOTOSTORAGE is set to "DB".



An application restart is required to apply the modified setting.

3.1.42 SESSIONTIMEOUT

The following table shows the default value and the allowed ones:

Key	Default Value	Allowed Values
SESSIONTIMEOUT	5	positive integer value

The value SESSIONTIMEOUT represents the time in minutes of user inactivity beyond which the software application performs the automatic logout.



An application restart is required to apply the modified setting.

3.1.43 EXAMINATIONCHART

The following table shows the default value and the allowed ones:

Key	Default Value	Allowed Values
EXAMINATIONCHART	patient_examination	any kind of .jasper file name

Open Hospital can produce a report about patient_examinations. (see 8.5.2 Patient Examination in the User's Guide).

By default, EXAMINATIONCHART is set to patient_examination which is the filename of the related report to use for the **Patient Examination** functionality. It is possible to use a different report by installing it in the report folder (see Reports) and by changing this parameter.



An application restart is required to apply the modified setting.

3.1.44 VISITSHEET

The following table shows the default value and the allowed ones:

Key	Default Value	Allowed Values
VISITSHEET	WardVisits	any kind of .jasper file name

By default, VISITSHEET is set to WardVisits which is the filename of the related report to use for the **Visit Sheet** functionality. It is possible to use a different report by installing it in the report folder (see Reports) and by changing this parameter.



An application restart is required to apply the modified setting.

3.1.45 TELEMETRYENABLED

The following table shows the default value and the allowed ones:

Key	Default Value	Allowed Values
TELEMETRYENABLED	yes	yes, no

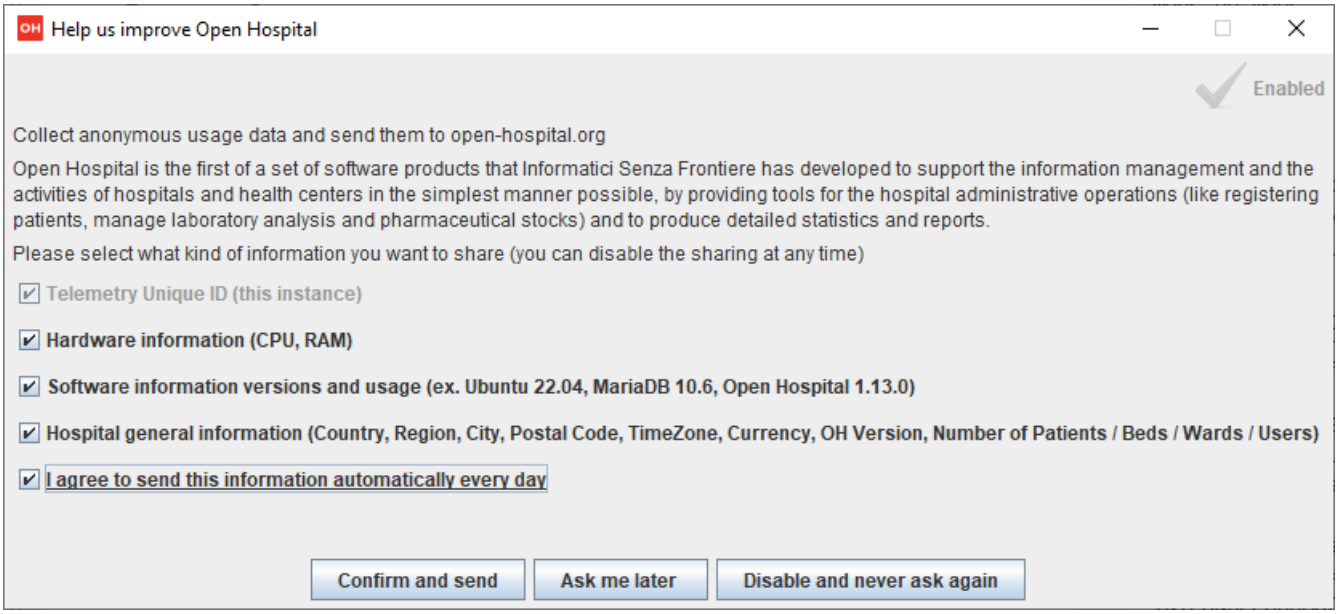
By default, the value of TELEMETRYENABLED is the set to `yes` indicating that the Telemetry module is enabled.

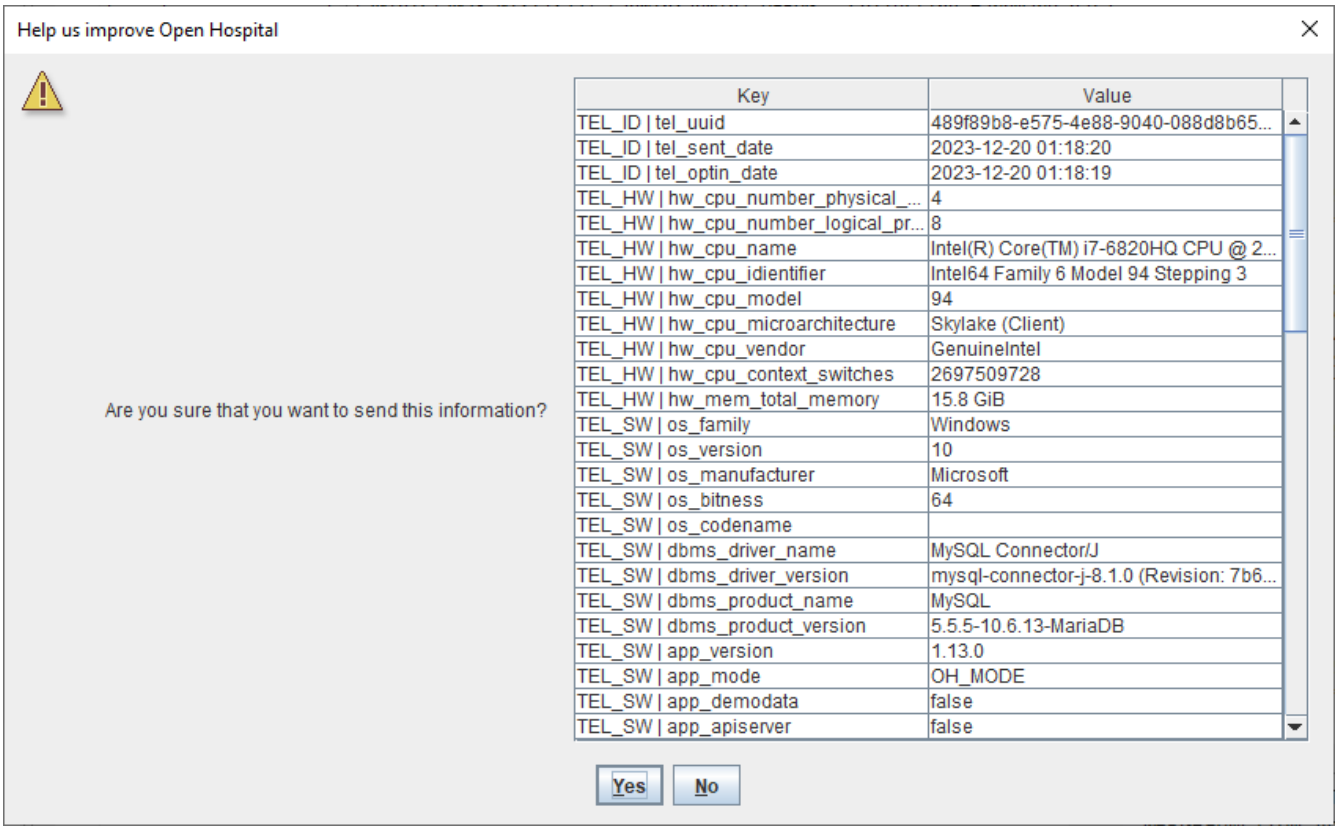
When the Telemetry module is enabled, users are presented with the option to participate in anonymous data collection on the software's usage during the initial application launch. This process is designed to gather information that contributes to the ongoing improvement of software quality.

On the first telemetry configuration, all data categories are selected by default. The mandatory identification category cannot be deselected; users can deselect any optional category before opting in. Preselected categories do not enable telemetry or send data: users must accept the telemetry agreement and explicitly confirm the data before it is sent. When the configuration is reopened, the saved category choices are restored, including categories previously deselected. Disabling telemetry preserves these choices for later review.

The collected data is transmitted to a server managed by ISF, solely for the purpose of optimizing the application and generating usage statistics. ISF ensures that this data is treated anonymously and with utmost respect for user privacy.

The Telemetry module can be disabled at any time through the Settings menu or via settings.properties.





3.1.46 PARAMSURL

This is the URL of the online configuration/parameters file.



This setting is not meant to be modified by the user.

3.2 database.properties

Database configuration is set in the *database.properties* file. Default database configuration is available in the *database.properties.dist* file:

```
jdbc.url=jdbc:mysql://DBSERVER:DBPORT/DBNAME
jdbc.username=DBUSER
jdbc.password=DBPASS
```

Use the provided startup scripts in order to automatically generate the *database.properties* file from the corresponding *.dist* file; after generation, optionally adjust the following parameters:

- **DBSERVER**: the IP address of the OH database server. IP address can be set to:
  127.0.0.1 – localhost / loopback network interface (local computer)
  192.168.0.100 – an example of a private IP address (LAN)
  217.147.110.117 – an example of a public IP address (WAN)
- **DBPORT**: the TCP port of the OH database server - default is "3306"
- **DBNAME**: the database name - default is "oh"
- **DBUSER**: the database user - default is "isf"
- **DBPASS**: the database user password



An application restart is required to apply the modified setting.



The same settings must be applied in *log4.properties*, see log4j2-spring.properties in this manual.



The *database.properties* file can be automatically generated/overwritten at any Open Hospital startup, by setting the WRITE_CONFIG_FILES option to "on" in the **oh.sh** / **oh.ps1** scripts.

3.3 dicom.properties

Imaging configuration is set in the *dicom.properties* file. A default imaging configuration is available in the *dicom.properties.dist* file:

```
#dicom.manager.impl=org.isf.dicom.manager.FileSystemDicomManager        # filesystem
storage
#dicom.manager.impl=org.isf.dicom.manager.SqlDicomManager               # database
storage
#dicom.max.size=1024B, 2048B, 1M, 16M, 256M, 512M, 1024M, 1G           # image size
examples
dicom.manager.impl=org.isf.dicom.manager.DICOM_STORAGE
dicom.storage.filesystem=OH_PATH_SUBSTITUTE/DICOM_DIR
dicom.max.size=DICOM_SIZE
```

Use the provided startup scripts in order to automatically generate the *dicom.properties* file from the corresponding *.dist* file; after generation, optionally adjust the following parameters:

- **DICOM_STORAGE**: This property accepts one of the following:

    - SqlDicomManager: Image files are stored in the database
    - FileSystemDicomManager: Image files are stored in the file system
- **OH_PATH_SUBSTITUTE/DICOM_DIR**: if the value of the first parameter is set to "FileSystemDicomManager", this property specifies the path where the image files are stored. It is possible to specify a local folder (*data/dicom_storage* is the default path under the OH installation folder) or to set a shared network folder, useful for LAN - client/server environment. The suggested configuration is to map the network/NFS folder under the *data/dicom_storage* path with a symbolic link. The shared folder needs read/write permissions for uploading images. Always use slash "/" in the path variable and not backslash "\".
- **DICOM_SIZE**: maximum allowed size for a dicom/jpg image (the MariaDB/MySQL server setting max_allowed_packet should be set with the same or a larger value). If not specified the default value of "4M" is used.

Example *dicom.properties* file:

```
dicom.manager.impl=org.isf.dicom.manager.FileSystemDicomManager
dicom.storage.filesystem=Z:/OH/Shared/Dicom_images
dicom.max.size=4M
```



The *dicom.properties* file can be automatically generated/overwritten at any **oh.sh** / **oh.ps1** startup, by setting the WRITE_CONFIG_FILES option to "on" in the **oh.sh** / **oh.ps1** scripts.

3.4 examination.properties

The examination module configuration is set in the *examination.properties* file. A default configuration file for this module is available in the *examination.properties.dist* file:

```
# This file contains PatientExamination module settings
LIST_SIZE = 10
HEIGHT_MIN = 0
HEIGHT_MAX = 250
HEIGHT_INIT = 0
#HEIGHT_STEP = 1
#WEIGHT_UNIT = kg
WEIGHT_MIN = 0
WEIGHT_MAX = 200
WEIGHT_INIT = 0
WEIGHT_STEP = 0.1
#AP_UNIT = mmHg
AP_MIN_INIT = 80
AP_MAX_INIT = 120
#HR_UNIT = bpm
HR_MIN = 0
HR_MAX = 240
HR_INIT = 60
```

```
#TEMP_UNIT = *C
TEMP_INIT = 36
TEMP_MIN = 30
TEMP_MAX = 50
TEMP_STEP = 0.1
#SAT_UNIT = %
SAT_INIT = 98
SAT_MIN = 50
#SAT_MAX = 100
SAT_STEP = 0.1
HGT_MIN = 30
HGT_MAX = 600
HGT_INIT = 80
DIURESIS_MIN = 0
DIURESIS_MAX = 2500
DIURESIS_INIT = 100
RR_INIT = 20
RR_MIN = 0
RR_MAX = 100
```

- **LIST_SIZE**: the maximum number of examinations that can be viewed in the history. Must be less than or equal to 10
- **HEIGHT_MIN**: the height minimum value (in cm), used to initialize the height slider in the ***Patient Examination*** window
- **HEIGHT_MAX**: the height maximum value (in cm), used to initialize the height slider in the ***Patient Examination*** window
- **HEIGHT_INIT**: the height default value (in cm)
- **WEIGHT *MIN*: the weight minimum value (in Kg), used to initialize the Weight slider in the _Patient Examination** window
- **WEIGHT *MAX*: the weight maximum value (in Kg), used to initialize the Weight slider in the _Patient Examination** window
- **WEIGHT _INIT**: the weight default value (in Kg)
- **WEIGHT _STEP**: the step (in Kg) used when moving the weight slider
- **AP _MIN**: the Arterial pressure minimum value (in mmHg)
- **AP _MAX**: the Arterial pressure maximum value (in mmHg)
- **HR _MIN**: the Heart rate minimum value (in bmp)
- **HR _MAX**: the Heart rate maximum value (in bmp)
- **HR _INIT**: the Heart rate default value (in bmp)
- **TEMP _INIT**: the temperature default value (in °C)
- **TEMP _MIN**: the temperature minimum value (in °C)
- **TEMP _MAX**: the temperature maximum value (in °C)
- **TEMP _STEP**: the temperature step (in °C) used when moving the slider

- **SAT _INIT**: the saturation default value (%)
- **SAT _MIN**: the saturation minimum value (%)
- **SAT _STEP**: the saturation step (%) used when moving the slider
- **HGT_INIT**: = the Hemo Glucose Test default value (in mg/dl)
- **DIURESIS_INIT**: = the Daily Urine Volume default value (in ml)

3.5 log4j2-spring.properties

The logging configuration is set in the *log4j2-spring.properties* file. Default logging configuration is available in the *log4j2-spring.properties.dist* file:

```
status = INFO
rootLogger.level = INFO
rootLogger.appenderRef.console.ref = STDOUT
rootLogger.appenderRef.rolling.ref = RollingFile

# Direct log messages to STDOUT
appender.console.type = Console
appender.console.name = STDOUT
appender.console.layout.type = PatternLayout
appender.console.layout.pattern = [%d{dd/MMM/yyyy HH:mm:ss}]
[%X{OHUserGroup}:%X{OHUser}] %-p - %m%n

# File Appender (with classes), daily rotation
appender.rolling.type = RollingFile
appender.rolling.name = RollingFile
appender.rolling.fileName= LOG_DEST/openhospital.log
appender.rolling.filePattern= LOG_DEST/openhospital.log.%d{yyyy-MM-dd}
appender.rolling.layout.type = PatternLayout
appender.rolling.layout.pattern = [%d{dd/MMM/yyyy HH:mm:ss}]
[%X{OHUserGroup}:%X{OHUser}] %-p - %m (%l)%n
appender.rolling.policies.type = Policies
# To change log file every day
appender.rolling.policies.time.type = TimeBasedTriggeringPolicy
appender.rolling.policies.time.interval = 1
appender.rolling.policies.time.modulate = true


# DB Appender (table columns)
appender.jdbc.type = JDBC
appender.jdbc.name = jdbc
appender.jdbc.connectionSource.driverClassName = com.mysql.cj.jdbc.Driver
appender.jdbc.connectionSource.type = DriverManager
appender.jdbc.connectionSource.connectionString =
dbc:mysql://DBSERVER:DBPORT/DBNAME?autoReconnect=true
appender.jdbc.connectionSource.userName = DBUSER
appender.jdbc.connectionSource.password = DBPASS
appender.jdbc.tableName = logs
```

```
appender.jdbc.ignoreExceptions = false
appender.jdbc.columnConfigs[0].type = COLUMN
appender.jdbc.columnConfigs[0].name = LOG_TIME
appender.jdbc.columnConfigs[0].pattern = %d
appender.jdbc.columnConfigs[0].isUnicode = false
appender.jdbc.columnConfigs[1].type = COLUMN
appender.jdbc.columnConfigs[1].name = LOG_LEVEL
appender.jdbc.columnConfigs[1].pattern = %5p
appender.jdbc.columnConfigs[1].isUnicode = false
appender.jdbc.columnConfigs[2].type = COLUMN
appender.jdbc.columnConfigs[2].name = MESSAGE
appender.jdbc.columnConfigs[2].pattern = %mm%ex%n
appender.jdbc.columnConfigs[2].isUnicode = false


# Assigning appenders to Hibernate packages (DB loggers)
# - hibernate.SQL to DEBUG for SQL queries to be logged
# - hibernate.type to TRACE for queries parameters to be logged with "binding
parameter [?]"
##logger.hibernate-SQL.name=org.hibernate.SQL
##logger.hibernate-SQL.level=DEBUG
##
##logger.hibernate-type.name=org.hibernate.type
##logger.hibernate-type.level=TRACE
```

Use the provided OH startup scripts in order to automatically generate the *log4j2-spring.properties* file from the corresponding *.dist* file; after generation, optionally adjust the following parameters:

- **DBSERVER**: the IP address of the OH database server. IP address can be set to:
  127.0.0.1 – localhost / loopback network
  192.168.0.100 – an example of a private IP address (LAN)
  217.147.110.117 – an example of a public IP address (WAN)
- **DBPORT**: the TCP port of the OH database server - default is "3306"
- **DBNAME**: the database name - default is "oh"
- **DBUSER**: the database user - default is "isf"
- **DBPASS**: the database user password
- **LOG_LEVEL**: the OH application log level - can be set to INFO|DEBUG|TRACE



The same settings must be applied in the *database.properties* configuration file, see database.properties in this manual.

```
status = INFO
rootLogger.level = INFO
rootLogger.appenderRef.rolling.ref = RollingFile
```

To investigate bugs or issues, the log level can be increased to "TRACE" in order to log more detailed

information:

```
status = TRACE
rootLogger.level = TRACE
rootLogger.appenderRef.rolling.ref = RollingFile
```

Logging can be routed to the database (DB) by adding the respective **appender**:

```
status = INFO
rootLogger.level = INFO
rootLogger.appenderRef.console.ref = STDOUT
rootLogger.appenderRef.rolling.ref = RollingFile
rootLogger.appenderRef.db = JDBC
```

Logging can also be routed to standard output (the console window) with:

```
status = INFO
rootLogger.level = INFO
rootLogger.appenderRef.console.ref = STDOUT
rootLogger.appenderRef.rolling.ref = RollingFile
```

If the DB appender is specified, the configuration must be set to match the settings in the *database.properties* file (see database.properties).



The *log4j2-spring.properties* file can be automatically generated/overwritten at any Open Hospital startup, by setting the WRITE_CONFIG_FILES option to "on" in the **oh.sh** / **oh.ps1** scripts.



DEBUG mode can generate large log files and should be avoided in a production environment.



An application restart is required to apply the modified setting.

3.6 sms.properties

The SMS communication module configuration is set in the *sms.properties* file. A default configuration file for this module is available in the *sms.properties.dist* file:

```
#####################################################################
# Global configuration
#####################################################################
# use: gsm-gateway-service || skebby-gateway-service || textbelt-gateway-service
sms.gateway=textbelt-gateway-service
sms.gateway.thread.timeout=3000
sms.gateway.thread.loop=15
```

```
sms.gateway.thread.icc=+39

... other sections
```

- **sms.gateway**: the SMS sender mode. Can take three values:
  - **gsm-gateway-service**: requires a GSM modem connected to a COM port. The administrator should configure the SMSGateway/GSM.properties file to set the suitable PORT.
  - **skebby-gateway-service**: requires a Skebby provider account.
  - **textbelt-gateway-service**: requires a Textbelt provider account.
- **TIMEOUT**: the timeout for the HTTP request. Required when sms.gateway is set to skebby-gateway-service or textbelt-gateway-service.
- **LOOP**: the delay used by the sender to fetch new SMS operations. The value is in seconds.
- **ICC**: the International Country Code that needs to be added to the phone numbers (if missing).

3.6.1 skebby-gateway-service

The default skebby-gateway-service section is:

```
#####################################################################
# Skebby configuration
#####################################################################
skebby-gateway-service.username=
skebby-gateway-service.password=
skebby-gateway-service.ribbon.base-url=https://api.skebby.it:443
# USER_KEY and ACCESS_TOKEN avoids the login call every time we need to send sms
skebby-gateway-service.accessToken=
skebby-gateway-service.userKey=
```

This file defines the parameters for sending SMS using the Skebby HTTP API.

- **username**: the username for authenticating to the Skebby server.
- **password**: the password for authenticating to the Skebby server.
- **ribbon.base-url**: the URL of the Skebby HTTP API.
- **accessToken**: the token generated by the Skebby provider upon user request (it replaces username)
- **userKey**: the key generated by the Skebby provider upon user request (it replaces password)

3.6.2 textbelt-gateway-service

The default textbelt-gateway-service section is:

```
#####################################################################
# Textbelt configuration
```

```
#################################################################
# enables/disables server testing mode (so that textbelt will do fake actions)
textbelt-gateway-service.enable-testing-mode=false
# use: textbelt (in order to send 1 free sms per day) or your api key (if you
purchased sms)
textbelt-gateway-service.key=textbelt
textbelt-gateway-service.ribbon.base-url=https://textbelt.com:443
```

This file defines the parameters for sending SMS using the Skebby HTTP API.

- **enable-testing-mode**: if set to "true", textbelt will do fake actions upon user/application requests.
- **key**: the API key (if using a purchased sms) from Textbelt provider (one can use "textbelt" to send 1 free sms per day)
- **ribbon.base-url**: The URL of the Textbelt HTTP API.

3.7 telemetry.properties

The telemetry configuration is set in the *telemetry.properties* file. A default configuration file for this module is available in the *telemetry.properties.dist* file:

```
# number of seconds to check if it should send a message
telemetry.daemon.thread.loop.seconds=14400

# enabled remote geo ip lookup service
# allowed values:
# - geoiplookup-remote-service
# - ipapi-remote-service
telemetry.enabled.geo.ip.lookup.service=geoiplookup-remote-service
# endpoints remote services
geoiplookup-remote-service.ribbon.base-url=https://json.geoiplookup.io
ipapi-remote-service.ribbon.base-url=http://ip-api.com/json
```

This file is used only if the TELEMETRYENABLED flag is true (see TELEMETRYENABLED in this document).

The file has the following configurable parameters:

- **telemetry.daemon.thread.loop.seconds**: number of seconds the Telemetry module should try again to send the telemetry data; if the telemetry data have been sent in the same day it will not send again.
- **telemetry.enabled.geo.ip.lookup.service**: the GeoIP service to use among the ones available in the comment: geoiplookup-remote-service or ipapi-remote-service; more services may be added in the future.
- **<service>.ribbon.base-url**: the URL for each available GeoIP service



An application restart is required to apply the modified setting.

3.8 txtPrinter.properties

The text printer configuration is set in the *txtPrinter.properties* file. A default configuration file for this module is available in the *txtPrinter.properties.dist* file:

```
# This file contains text printing information
# MODE = TXT, PDF or ZPL

# TXT_CHAR_HEIGHT: height of a text row used by JRTextExporter to convert the
# JRXML layout into a text grid. It should match (or be proportional to)
# the height of the fields in the jrxml report (e.g. 18 in our templates).

# TXT_CHAR_WIDTH: width of a text column used by JRTextExporter. It depends on
# the printable width of the report and determines how many characters fit
# on one line (typically around 42□48 columns for 80mm receipts).

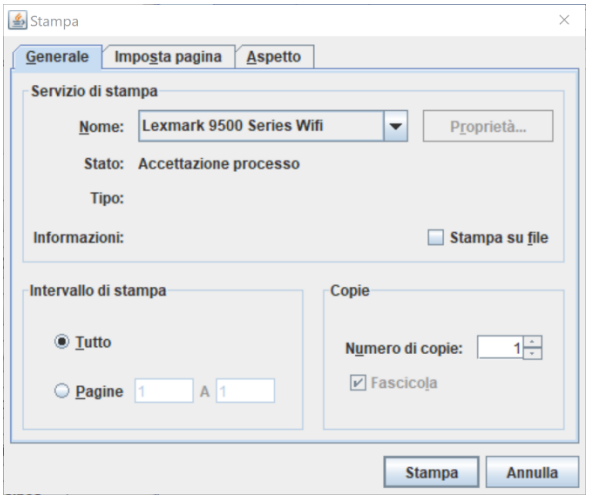
# TXT_FEED_LINES: number of blank lines sent before finishing the print
# TXT_CUTTER: enable/disable the ESC/POS cut command

USE_DEFAULT_PRINTER=yes
PRINT_AS_PAID=no
PRINT_WITHOUT_ASK=no
MODE=PDF
TXT_CHAR_HEIGHT=18
TXT_CHAR_WIDTH=12
TXT_FEED_LINES=3
TXT_CUTTER=yes
ZPL_FONT_TYPE=0
ZPL_ROW_HEIGHT=25
```

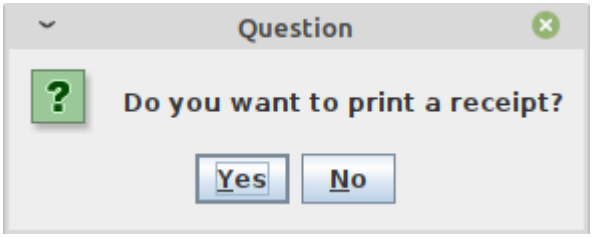
This file is used only if the RECEIPTPRINTER flag is enabled (see RECEIPTPRINTER in this document).

The file has the following configurable parameters:

- **USE_DEFAULT_PRINTER**: If it is set to *yes*, Open Hospital will use the default printer, otherwise it will show the system printing dialog allowing a change to the printer at every print attempt:



- **PRINT_AS_PAID**: This changes the behavior when a Bill is set to PAID in the Accounting module (see 6.2.1.11 Print receipt function in User's Guide).
  - If it is set to *yes*, Open Hospital will try to print a receipt just after a Bill is set to PAID
    - A confirmation window will appear (see below), otherwise, it must be done after, by pressing the **Edit Bill** button on the already closed ("C") bill



- **PRINT_WITHOUT_ASK**: if set to yes Open Hospital will try to print the receipt without the confirmation window
- **MODE**: the MODE can take three values:
  - **TXT**: the bill is printed as pure text using a monospaced text layout suitable for thermal POS printers
  - **PDF**: the bill will be printed as a PDF
  - **ZPL**: the text printer connected to the system works with ZPL language (an EPL evolution for Zebra Label Printer)
- **TXT_CHAR_HEIGHT**: controls the text row height used by the JasperReports TXT exporter.
  This value must be consistent with the JRXML layout to obtain proper line spacing and alignment.
- **TXT_CHAR_WIDTH**: controls the text column width used by the JasperReports TXT exporter.
  This value affects the number of printable characters per line and should be adjusted according to the paper width and font size.
- **TXT_FEED_LINES**: number of empty lines appended at the end of the receipt before closing the print job.
  Useful to improve paper feeding on thermal POS printers.
- **TXT_CUTTER**: if set to *yes*, Open Hospital sends the ESC/POS cutting command after printing.
  If set to *no*, the printer will not attempt automatic cutting.

Useful for printers without an automatic cutter.

- **ZPL_FONT_TYPE**: a 0 (zero) value stands for a standard character; the value can be changed to “A”, “B”, “C”, etc. according to the device datasheet, to obtain a better look
- **ZPL_FONT_SIZE**: an integer value to adjust the character size to obtain a better look



The best way to configure TXT and ZPL parameters is to adjust them incrementally and print test receipts until the best alignment and readability are achieved for the target printer.



An application restart is required to apply the modified setting.

3.9 xmpp.properties

The XMPP module configuration is set in the *xmpp.properties* file. A default configuration file for this module is available in the *xmpp.properties.dist* file:

```
# This file contains Xmpp Server information
DOMAIN=127.0.0.1
PORT=5222
```

This file is used only if the XMPPMODULEENABLED flag is enabled (see XMPPMODULEENABLED in this document).

To use the Communication module (see 11 Communication in the User’s Guide), an XMPP Server must be installed and configured separately on a server/computer in the local network. The XMPP server can also be installed on the same machine as the OH database server.

The most common XMPP Server is the free and open-source project Ignite Openfire® (https://www.igniterealtime.org/projects/openfire/) available for Linux, Windows, and Mac.

Once the XMPP Server is installed, active and running set the two parameters in the *xmpp.properties* file as follows:

- **DOMAIN** – the IP address of the XMPP server (it could be the same as the database server)
- **PORT** – 5222 or another if set differently



An application restart is required to apply the modified setting.

3.9.1 OpenFire Settings

The XMPP module needs some knowledge about the XMPP protocol and how an XMPP server works.

Every new user login into Open Hospital creates an OpenFire user with the same username and password (even if passwords are saved differently in OH and OpenFire DB). For this reason, the OH "admin" user may conflict with the OpenFire user, not allowing the log in as admin and to chat.



If the admin user needs to chat with other users consider creating another "superuser" in OH under the "admin" group.

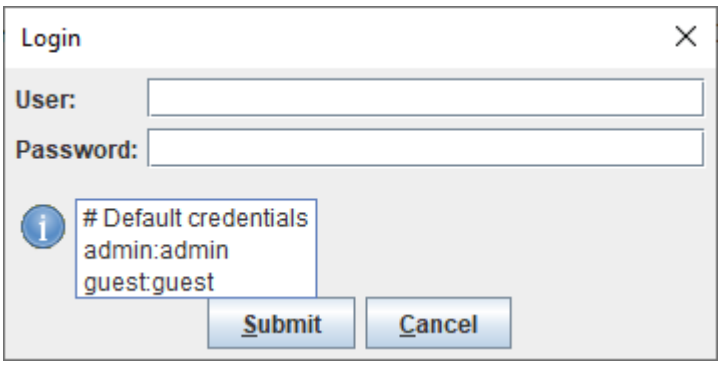
To allow users to see each other and communicate via XMPP protocol, they must belong to the same OpenFire "group" and have the other users in their own "*roster*" (find more comprehensive information about "group" and "roster" in the online documentation).

Please follow these settings:

1. Install the OpenFire server (it is better if from the zip file) on a writable path
2. Set in both OpenFire and in the *xmpp.properties* file an IP address rather than the FQDN (e.g., 127.0.0.1)
3. Set OpenFire to use Embedded DB
4. Create a group "OH" in the OpenFire server and enable the "Contact List (Roster) Sharing"
5. Add new users automatically created from OH to the "OH" group
6. Consideration should be given to using the OpenFire plugin to automatically add new users to a group (e.g., registration plugin https://www.igniterealtime.org/projects/openfire/plugins/registration.jar)

3.10 default_credentials.properties

Open Hospital is packaged with a login "hint" visible only if the *default_credentials.properties* file is found.





Mouse over the "i" icon to see the hint.

To not show the hint just delete *default_credentials.properties* file.

The original values are found in *default_credentials.properties.dist* and *default_demo_credentials.properties.dist* files.

```
# Default credentials
admin:admin
guest:guest
```

```
# Default demo credentials
admin:admin
guest:guest
laboratorist:laboratorist
doctor:doctor
```



It is always recommended to change the **admin** user password in a production environment.

3.11 Bundles

Bundles are the language (or translation) files provided with Open Hospital. As described in the LANGUAGE chapter, Open Hospital comes with different available languages in the folder **bundles/** within the package file:

language_XY.properties

where XY is an international country code.

These property files (text format) contain multiple **key** = **value** pairs containing localized text for the language specified.

All files must be encoded in **UTF-8** to accept any language-specific characters (e.g. è ì ò à ñ ú ù ¡ ¿ ...).

3.11.1 New Translations

To create a new translation, simply copy the English file, rename it with the new country code in place of the “XY” in the filename, and start translating with a simple text editor. A text editor can also be used to modify existing translations that are incorrect.

For instance, if to create an Arabic translation, copy a new bundle from the English one by copying *language_en.properties* to a **new file** *language_ar.properties*.

Then edit the Arabic bundle file and set the LANGUAGE parameter to **ar** to start testing the translation.

4 Reports

Reports in Open Hospital are produced with JasperReports® technology. JasperReports gives the possibility to design a report in an visual / WYSIWYG editor, connect it to a datasource (DB), test it, modify it, and then compile it to use it directly from the OpenHospital application.

Open Hospital reports are placed into three different folders (see Folders):

- **rpt_base/** → contains all the hardcoded reports used by the application;
- **rpt_stat/** → contains the *OHxxx* basic reports available in the Statistics menu;
- **rpt_extra/** → contains additional reports defined by the user.

Each report consists of two files:

- a .**jrxml** file: JasperReport XML file, that can be modified with Jaspersoft® Studio (see below)
- a .**jasper** file: Jasper file, which is the compiled version of the jrxml report, suitable to be run in Open Hospital

Reports that support translation also include:

- a .**properties** file: default translation file, placed alongside the `.jasper` file. This file must be written in **English**, as it serves as the fallback for all languages that do not have a specific translation. It must contain the **jTitle** property that defines the name of the report visualized in the Open Hospital application, otherwise the application will not show it (this can be useful for subreports).
- language-specific .**properties** files: placed in a `<lang>/` subfolder (e.g., `it/Laboratory.properties` for Italian), with the same filename as the default. If no language-specific file is found for the current language, Open Hospital falls back to the JRXML default automatically.



Translations in languages other than English are managed via Transifex, which will automatically attempt to provide translations for new files and changes.

In order to create / modify a report, it is possibile to use a free application, TIBCO Jaspersoft® Studio version 6.14.0 or later. To define a new report it is possible to start from scratch or modify an existing Open Hospital report:

- open Jaspersoft® Studio
- open the original jrxml file
- modify the report as desired
- save the .jrxml file with a new name
- re-compile the .jrxml file in jasper format (generate the .jasper file)
- place the .jasper file in the report folder used by the Open Hospital application (rpt_extra/)
- Open Hospital will show the custom reports in the Statistics menu.

5 Installing Open Hospital 1.15.0 in Eclipse EE

The OH source code is available on GitHub at the following links:

- openhospital-core
- openhospital-gui
- openhospital-doc

Please use EGit (Eclipse Plugin for Git) to clone the code into the Eclipse instance.

Assuming Eclipse EE, MariaDB/MySQL and the Java Virtual Machine 11 or higher (Java8 or greater) are already installed on the computer, this chapter will focus on cloning version 1.15.0 into the Eclipse J2EE (or Eclipse EE) environment.

Please follow these EGit User Guide - Working with remote Repositories.

5.1 Run the Project

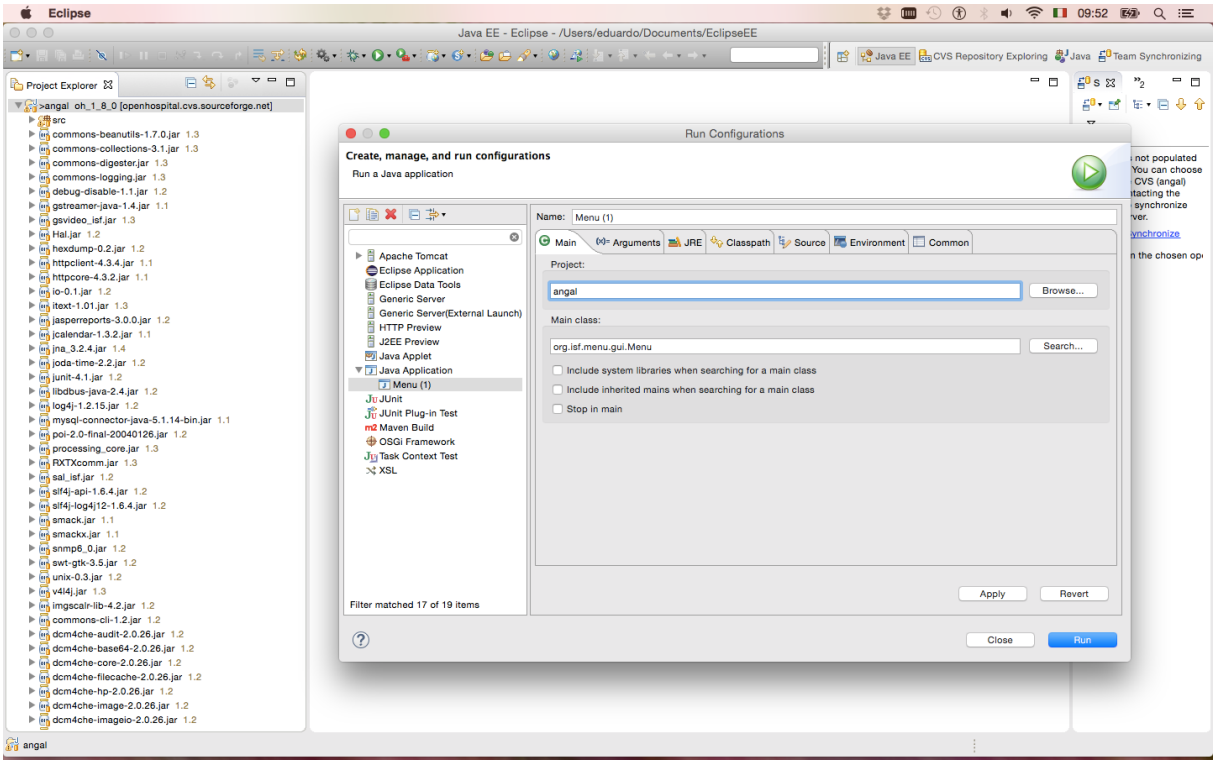


Before running the project, the database must be created as explained in the chapter MySQL Server and Open Hospital Database.

To run the application, look for class *openhospital-gui/src/main/java/org/isf/menu/gui/Menu.java*, then right-click on it and choose *Run As → Java Application*. If everything is OK, the splash screen will appear. If not, look in the Eclipse console for any error or warning messages.



Open Hospital's advanced features require "native" libraries generated to match the computer's operating system. Right-click on the project's name, then select *Run As → Run Configurations*:

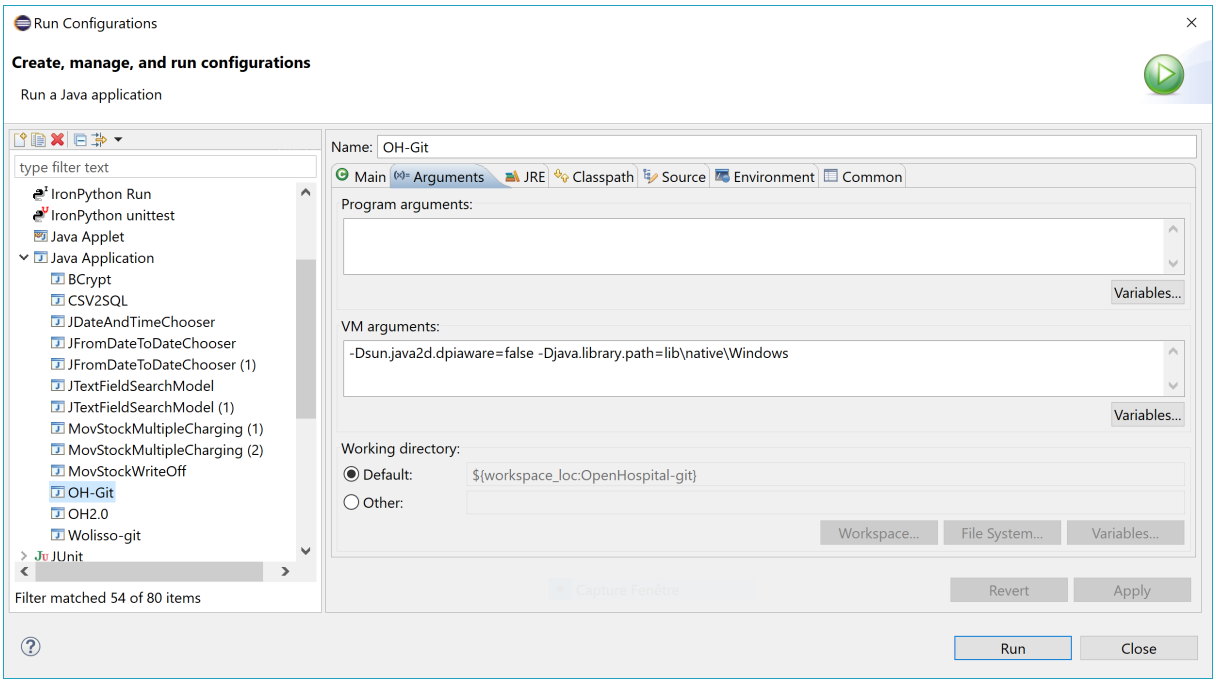


A “Run Configurations” window opens. Choose Java Application → <application’s name> - Menu (1) on this example. Select the “Main” tab, write “angal” on the “Project” field and check if the “Main class” is org.isf.menu.gui.Menu.

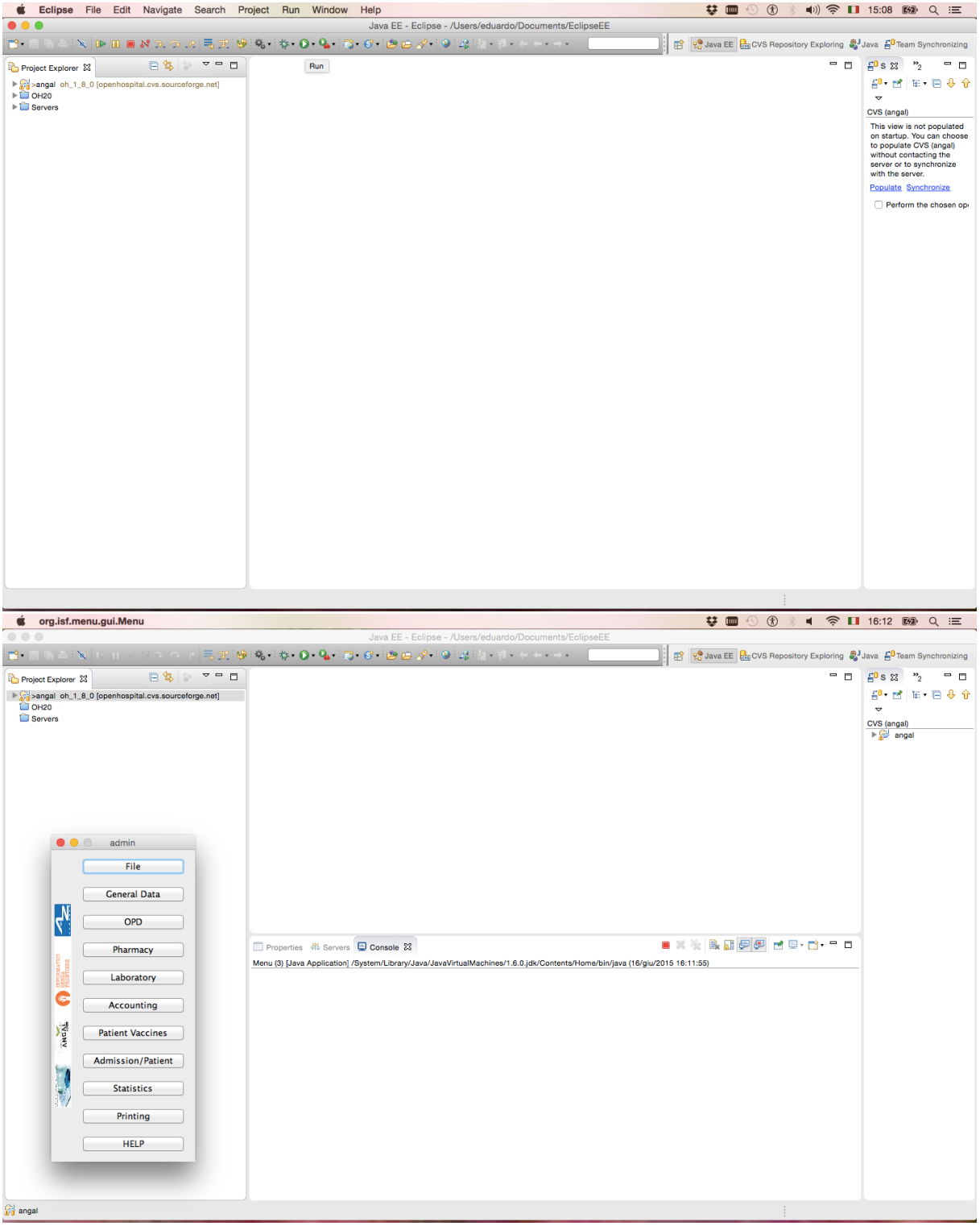
Select the “Arguments” tab and fill in the “VM arguments” path typing (on a single line):

```
-Dsun.java2d.dpiaware=false -Djava.library.path=lib\native\<OS name>
```

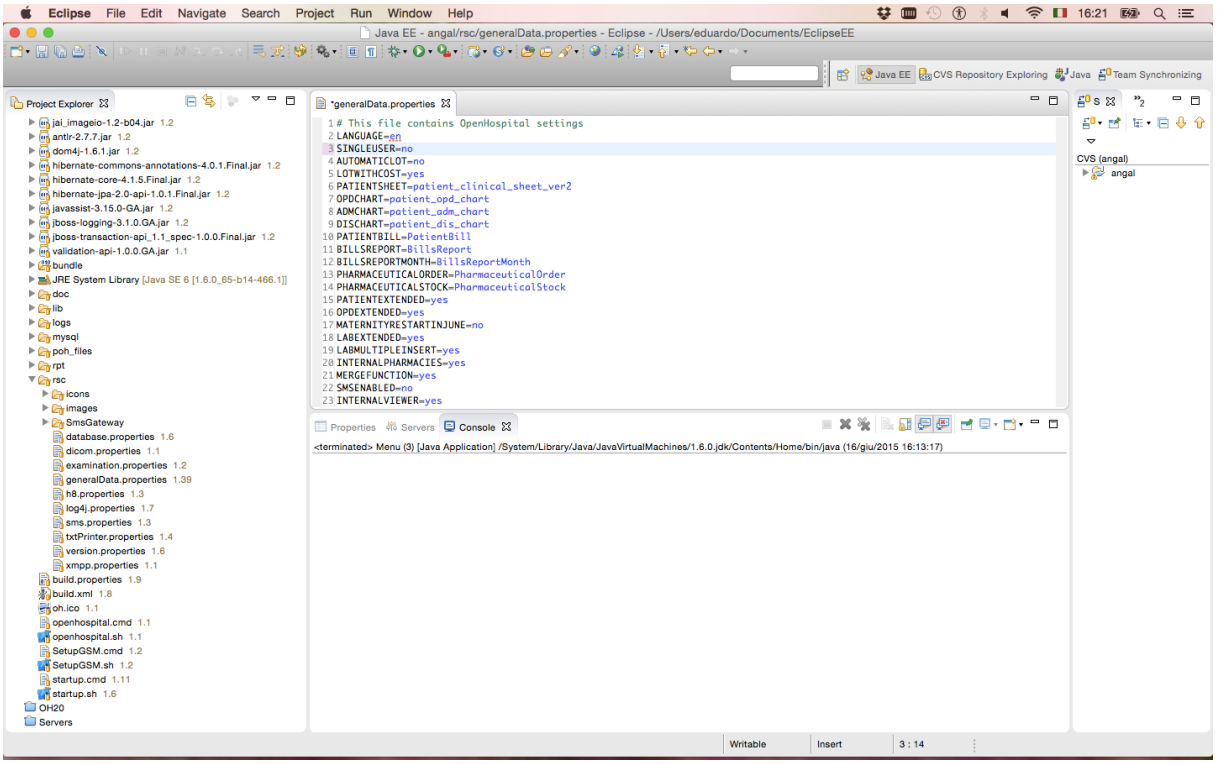
<OS name> can be “Mac_OS_X”, “Solaris”, “Linux”, “Win64” (for 64bit Windows) or “Windows” (for 32bit Windows) according to the folder’s schema found in the related folder “native”. Click on “Run” to close the window and run the application.



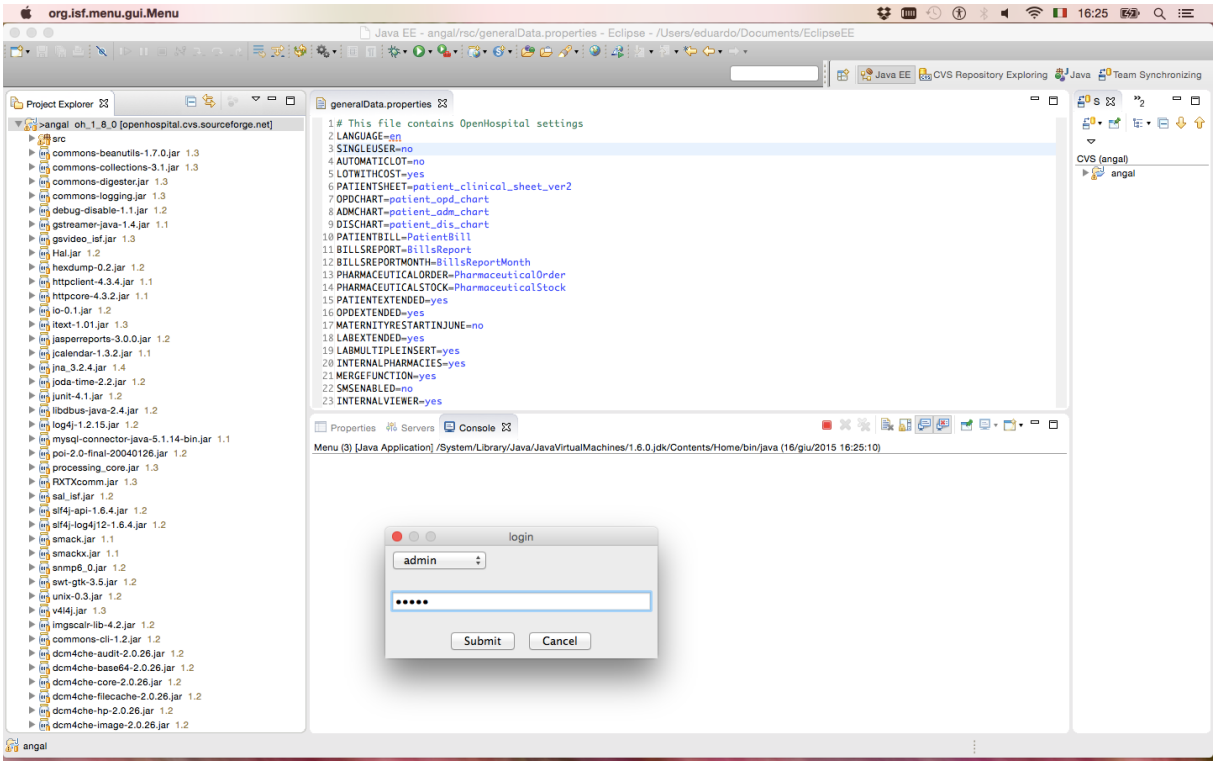
Now that the run configuration settings have been selected, OH can be launched by just clicking on the “Run” button (shown below by an arrow).



Because of the default settings, Open Hospital’s first run does not show the login window. The system defaults to “single-user” mode by default. This can be verified by the “SINGLEUSER=yes” string on the third row of the settings.properties file. In production use this is unacceptable as Open Hospital must be accessible only to the administrator and logged users. Click on the triangle near the project’s name, do the same on the “rsc” subfolder and then double-click on the settings.properties file.



Replace “yes” with “no” on the “SINGLEUSER” row. Press CTRL+S (or Command+S on Mac) to save the change. Now rerun the application and after the splash window, a “login” window opens. Click on the drop-down menu to choose the role. To access as the administrator, select “admin” and type the password. The default is “admin”. The administrator has the right to register users as guests. Guests have to choose “guest” from the drop-down menu and then type their password.



If the password is not correct, an error message is shown, asking the user to try again.

To better understand the processes built in the software, also enable the “Standard Output” in *log4j2-spring.properties* as explained in the chapter log4j2-spring.properties.

6 Update Open Hospital

New releases, as well as this Admin Manual and the User Manual, are regularly released on GitHub/SourceForge. See the Download section for more details. NOTE: Following instructions only apply between “official releases”.

Minor releases (if not otherwise specified) do not need changes to the database structure, so the working copy can be replaced on each client with the new one and then connect to the same DB (see database.properties and log4j2-spring.properties).

It is important to keep/preserve files that have been created or modified, especially configuration/settings and data files (see Configuration). The easiest way is to backup the working copy and re-apply those settings in the new installation (see the Backup & Restore chapter).

Major Releases: Changes are documented in the RELEASE_NOTES file.

6.1 Update Database

Major OH releases usually require changes to the database structure, and these changes are implemented via different SQL scripts included in the *sql/update* folder (e.g.):

Previous OH version	New OH version	Database update script
1.8.4	1.9	*update_1_8_4-1_9.sql*
1.9	1.10	*update_1_9-1_10.sql*
1.10	1.10	*update_1_10-1_11.sql*
1.11	1.12	*update_1_11-1_12.sql*
1.12	1.13	*update_1_12-1_13.sql*
1.13	1.14	*update_1_13-1_14.sql*

To perform the database update process follow these steps:

1. Close the program if it is still running
2. Backup the current database for safety (see Backup & Restore)
3. Open a terminal in the folder sql/update (e.g.):

```
C:\WINDOWS\system32> cd C:\Users\OH\OpenHospital-v1.15.0\sql\update

C:\Users\OH\OpenHospital-v1.15.0\sql\update>
```

4. Run the following commands (e.g.):

```
C:\Users\OH\OpenHospital-1.15.0\sql\update> mysql -u root -p [-h hostname/ip address]
```

```
Enter password: ****
```

5. Use the 'root' password chosen during the installation process. The terminal should reply with the MySQL client command line prompt:

```
MariaDB>
```

6. Select the open hospital database;

```
MariaDB>USE oh;
```

7. Execute all the SQL scripts from your current OH version up to the new one:

Example: updating version from 1.10.X to 1.13.X

```
MariaDB> source update_1_10-1_11.sql

MariaDB> source update_1_11-1_12.sql

MariaDB> source update_1_12-1_13.sql

...
```

8. If any error message occurs you can report an issue here: https://openhospital.atlassian.net/

6.2 Update Client

If the database upgrade procedure is successful, it is possible to update the OH software by replacing the OH working copy on each client with the new one. (see database.properties and log4j2-spring.properties). Remember to maintain the configuration files that have been changed, to preserve the custom settings. (see Configuration). The most common way is to backup the working copy and re-apply those settings in the new one (see Backup & Restore). Start the program and check the log files (data/logs) to ensure that the new version is running and has no error messages.

6.3 Update Portable



Open Hospital, used in PORTABLE mode, is not meant to be used in a production environment. To migrate a portable installation to a full client/server configuration, keeping the existing data, the following steps must be performed:

1. Close Open Hospital;
2. use oh.sh / oh.ps1 **e** option to export the database
3. overwrite local working copy with the new version files

4. use oh.sh / oh.ps1 **r** option to restore exported database

If needed, start the PORTABLE instance in SERVER mode, connect to the local running database and apply the required SQL scripts.

7 Support

For support, or questions about updating, migrating, or maintaining an Open Hospital installation, please check our website at https://www.open-hospital.org. You can contact us at: https://www.open-hospital.org/contact and specify:

- A brief history of the organization;
- The country where the installation is and the language used;
- Technical details of the OH installation;
- The application logs (see Help chapter in the User's Guide)
- The current Open Hospital version;
- Complete the Open Hospital Assessment online-form form

8 License

 Informatici Senza Frontiere Onlus, 2026

Administrator's Guide, © 2026 by Informatici Senza Frontiere Onlus
Policies is made available under a Creative Commons Attribution-ShareAlike 4.0 International License: https://creativecommons.org/licenses/by-sa/4.0//.

[1] Informatici Senza Frontiere - https://www.informaticisenzafrontiere.org

[2] DataBase Management System

[3] See https://www.mariadb.org

[4] https://mariadb.org/download/

[5] https://mariadb.org/documentation/