

Step by Step Installation for Developers



Below is a listing of software applications that you will need to install in order to develop on the OpenMRS Platform code base. The installation process may vary somewhat for Windows, Linux ([Step By Step Instructions](#)), or Mac OS X platforms ([Step-By-Step Instructions](#)). Still, the OpenMRS Platform depends mostly on Java which runs the same across those systems. This list will try to describe the software which you should install to start work on any development computer.

See also <http://notes.openmrs.org/OpenMRS-University-2012-12-19> where you can find videos showing the same steps.

Dependencies to install

Java Version

Java is the language in which the OpenMRS Platform is written. The latest released version of the OpenMRS Platform requires at least Java version 1.8. You should be able to build and run supported maintenance branches with versions 1.6 or later. If you plan to edit the source, you will need the [JDK](#); if you plan just to run the OpenMRS Platform, you will only need the [JRE](#).

If using a JDK in Eclipse, then navigate to *Window-> Preferences-> Java-> Installed JRE's* to check if the JDK has been set. If not, do so.

Note: It seems that attempting to use Java 10 on Windows causes a Maven build error with "error: package com.sun.tools.doclets does not exist". Using Java version 1.8 resolves this.

MySQL

MySQL is the database software most people use for the OpenMRS Platform database. You should know your MySQL root password, or have a database ready with username and password to install the OpenMRS Platform into. The latest version of openMRS core supports mysql 5.7

Install And Configure Your IDE

Refer to the [How-To Setup And Use Your IDE](#) to either install an IDE or configure the one you have for OpenMRS development.

(Optional) Tomcat

Jetty, a development-oriented J2EE server, is automatically provided with the OpenMRS Platform 1.8+, via Maven, and it integrates nicely with Eclipse's Run and Debug workflows. It is perfectly sufficient for most developer purposes.

Tomcat is a J2EE server on which the OpenMRS Reference Application can be deployed. Most production installations of the OpenMRS Reference Application run on tomcat, so if you want to test against the most common environment, you should install tomcat. You will want the core that matches your system specs the closest: probably 32-bit zip or 64-bit zip or bz2 on Linux.

- [Download Tomcat 8.x \(recommended\)](#)
- [Download Tomcat 7.0.73](#)

Check out the core from GitHub

Most importantly, you need to get the OpenMRS Platform code from GitHub: <https://github.com/openmrs/openmrs-core>

See [Using Git](#) for the overall docs.

For specific instructions for using Egit and Maven tools you have installed above, see [Git IDE Integration](#).

If you are contributing to the OpenMRS Platform code base, you want to check out "openmrs-core". If you want to contribute to a module, you should check out the release branch for the module's lowest required OpenMRS Platform version (e.g. *1.8.x* or *1.9.x*). OpenMRS began using Maven in 1.8.x, so for modules that run on earlier versions of the OpenMRS Platform, you may find it easier to just check out *1.8.x* and test against that.

Compiling the OpenMRS Platform

Use [Maven](#) to compile and generate source files.

Go to the most outer project directory and run:

```
mvn clean install
```

It will compile, run tests, build all artifacts and install them to your local repo.

Running OpenMRS

Before trying to run the OpenMRS Platform ensure you have MySQL running. Also, it is highly recommended to increase the memory available by following the steps in the [Maven](#) setup, or see [Troubleshooting Memory Errors](#).

You can now run the OpenMRS Platform by deploying the built WAR file into a J2EE container such as Tomcat or Jetty, or you can run it directly from the source code using the [Maven Jetty plugin](#). The latter method is easier for developers as you can launch it from your IDE. If you choose to use the [Maven Jetty plugin](#), you can start the OpenMRS Platform as follows: move into the "*webapp*" directory and execute following command

```
mvn jetty:run
```

Wait for the *[INFO] Started Jetty Server message* and open the web browser at <http://localhost:8080/openmrs>. Follow the instructions.

Note: The latest version of openMRS core supports mysql 5.7 Server and above. An error might flag if the version is below 5.6.

[Installation for Developers On Windows supplemented with images.](#)

(Optional) Module development

To contributing to an existing module, you need to check out its code.

If the module's source code is in Git, follow the [Using Git](#) instructions. If the module's source code is in subversion and the module is mavenized (i.e. it has a *pom.xml* file) then you should follow the same process [described for the core code](#).

Older modules may be in subversion, and not yet mavenized (i.e. they have no *pom.xml* file). To check these out do *New -> Checkout project from SVN*.

To create a new module, we recommend [Using Git](#) and [Using the Module Maven Archetype](#).

After you cloned a module you need to build it with `mvn clean install`.

To install a module, start OpenMRS Platform and go to Administration -> Manage modules -> Add or upgrade module. You need to upload here an omod file, which you can find under `cloned_module/omod/target`.

Known issues

Once checked out, the pom file of each OpenMRS module will display the following error in Eclipse - "Plugin execution not covered by lifecycle configuration: org.codehaus.mojo:build-helper-maven-plugin:1.5:parse-version (execution: parse-version, phase: initialize)"

This has no effect on the operation of the application, and can be safely ignored. (You may also choose "quick fix" and install the scm connector for this plugin.)