

# OpenMRS Reference Application Release Process

## On this Page

- [Release Timeline for April Releases](#)
- [Release Timeline for October Releases](#)
- [Future Releases](#)

### Prerequisites

#### Release Process

- [Begin the process of release management:](#)
- [Releasing the Reference Metadata module:](#)
  - [Required accounts:](#)
  - [Release steps:](#)
- [User Acceptance Testing:](#)
- [Final Release Process:](#)
- [Finalizing and announcing the release:](#)
- [Maintenance Release Process:](#)

## Release Timeline for April Releases

Even numbered releases usually happen in this timeline Eg - Reference Application 2.4.0, Reference Application 2.6.0 etc.

- **Early February:** Select release manager for October release. April release manager announces modules.
- **Mid February:** Code freeze for April release.
- **Early March:** Release all required modules for April release.
- **Mid March:** Set up UAT for April release
- **Early April:** Address any issues from UAT for April release
- **April End:** Announce the release and post it in the OpenMRS blog and update download links

## Release Timeline for October Releases

Odd numbered releases usually happen in this timeline Eg - Reference Application 2.5.0, Reference Application 2.7.0 etc.

- **Early July:** October release manager announces modules.
- **August:** Select release manager for next April release. Code freeze for October release.
- **Early September:** Release all required modules for October release.
- **Mid September:** Set up UAT for October release.
- **Early October:** address any issues from UAT for October release
- **October End:** Announce the release and post it in the OpenMRS blog and update download links

## Future Releases

Release Version	Release Date	Release Process Start Date
2.10.0	TBD	

Release Version	Release Date	Release Process Start Date
<a href="#">2.9.0</a>	08 Apr 2019	20 Sep 2018
<a href="#">2.8.0</a>	27 Jun 2018	
<a href="#">2.7.0</a>	10 Oct 2017	
<a href="#">2.6.0</a>	28 Feb 2017	01 Nov 2016
<a href="#">2.5</a>	30 Sep 2016	01 May 2016
<a href="#">2.4</a>	15 Jul 2016	12 Jan 2016
<a href="#">2.3</a>	27 Oct 2015	03 Aug 2015
<a href="#">2.2</a>	23 Apr 2015	11 Feb 2015
<a href="#">2.1</a>	07 Oct 2014	~

2.0	21 Feb 2014	~
-----	-------------	---



Pro-tip: **READ THIS PAGE COMPLETELY** before you start. That means, go through each point to make sure you can anticipate any need for clarifications.

**OpenMRS Reference Application** releases are different than [OpenMRS Platform releases](#). Platform releases usually consist of alpha, beta releases and have Sprints to complete tickets in a short span of time. OpenMRS 2.x is a collection of modules and hence its release consists of releasing all the modules planned to be included in the 2.x release and testing if all of them function collectively.

## Prerequisites

The release manager needs access to these accounts:

1. **OpenMRS ID:** This is the basic account that you create when associating with OpenMRS.
2. **CI Bamboo Account:** The actual release takes place here. You will need your access to this account to finally release the version. Create a new case at [OpenMRS helpdesk](#) requesting access for this account.
3. **Administrator role in JIRA for Reference Application Project:** Request for being an administrator of the Reference Application project in JIRA from a current Administrator - Presently Darius Jazayeri is an administrator for the project.

## Release Process

### Begin the process of release management:

1. Analyse the release stage requirements for the particular release by following [Technical Roadmap](#).
2. Attend and organize [Project Management Meeting](#) and [Design Forum](#) to communicate and decide on what needs to be done throughout the release process.
3. Create a release management process page to help you and others understand the outstanding tasks to be done.
  - a. This can be done on a release tracking wiki page like [this one for Ref App 2.7 release tracking](#).
  - b. This can be done on [OpenMRS Talk](#) as done [here](#) (using the wiki has advantages, so use it if you are uncertain which to use)
4. Search for tickets in [JIRA](#) that have the release version or having tag as OpenMRS 2.x (the version to be released) and have them fixed before having that module released.
5. Before people start releasing Reference Application modules, set the bamboo global variable [preparing.refapp.distro.release](#) to 'true' (by asking helpdesk if you don't have access)
6. Make sure that for each module that needs a new version to be released, that the user documentation in the new wiki page has been updated to reflect the UI changes if necessary.
7. Set an initial deadline to release all the modules scheduled to be included in the release and request each modules owner to release the modules in that time (make this announcement in Talk, mentioning each module owner so they get notified). Excepting the following :
  - Reference Application distro,
  - Reference Application,
  - Reference demo data,
  - Reference metadata.
8. All the other module can be released by this time. For the modules already existing in the release bundle check in [OpenMRS github account](#) if any new changes have been made after the previous release if don't request the any action just include it in the release management process that it is in a release ready state. If there are changes request the module owners if they would like to release a new version to be included in the new OpenMRS release.
9. A module should be released and available via [Addons](#) to be considered released. Please verify that each module has been released properly.

### Releasing the Reference Metadata module:

Each release of Reference Application should be preceded with a release of the Reference Metadata module in a version matching the Reference Application version. The release is typically done by the Reference Metadata module maintainers, but as a release manager you need to make sure it happens.

#### Required accounts:

**CIEL dictionary dropbox account:** This account is updated with the latest CIEL dictionary and each OpenMRS 2.x release includes the latest released CIEL dictionary.

Andrew Kanter manages the account and you would need to [mail him](#) (Please refer to his new [mail here](#) ) requesting the access for the account. CIEL dictionary is an SQL dump file

1. **mdsbuilder:** It is an OpenMRS server and you would need access to the server. Please create a new case requesting these accounts at [OpenMRS helpdesk](#) .
  - For requesting server access you need to have an SSH key pair. (For windows users new to using an SSH key pair for accessing a linux server , [this guide](#) might help you create one. You will have to request helpdesk to give access to this SSH key).

- You also need need to request for Login Credentials with **System Developer** role at [OpenMRS helpdesk](#) to be able to update MetaData Sharing Packages After Updating the CIEL dictionary.

## Release steps:

1. Before You update CIEL on the mdsbuilder , Ensure the server runs on the platform version on which the Reference Application version to be released will run. In case the server doesn't run on the platform version as the reference application to be released , Carry out the following step below,

- Check [openmrs mdsbuilder Docker hub repo](#) for the latest Docker image of the **openmrs/openmrs-distro-mdsbuilder** . Ensure the right version of the platform was set as per the [ditro file here](#) . And if the Server is not running on the desired platform version , Pull the updated docker image and deploy it to the server using build [the CI build](#)
- Then Login into the server and complete the installation wizard

2. Update CIEL on the mdsbuilder to the latest version using the sql dump that you download from the CIEL dictionary dropbox account.

Here's the list of commands needed to do the import. Replace "<username>" with your username.

**NB:** Before you Upload the sql dump file to the mdsBuilder Server , ensure the sql dump file version you upload matches the Openmrs Platform version on which the mdsBuilder server run . ie Platform 2.2 would match "openmrs\_concepts\_2.2\_20190701.sql" file and Platform 2.0 .6 would match "openmrs\_concepts\_2.0\_20190701.sql" file

```
scp openmrs_concepts_2.0_20181012.sql.zip <username>@mdsbuilder.openmrs.org:/tmp/openmrs_concepts_2.0_20181012.sql.zip
ssh <username>@mdsbuilder.openmrs.org
cd /tmp
unzip openmrs_concepts_2.0_20181012.sql.zip
sudo -i
cd /root/docker/mdsbuilder
docker cp /tmp/openmrs_concepts_2.0_20181012.sql mdsbuilder_db_1:/tmp/
docker exec -it mdsbuilder_db_1 /bin/bash
# copy as is, do not attempt to replace it
mysql -u ${MYSQL_USER} -p${MYSQL_PASSWORD} ${MYSQL_DATABASE} < /tmp/openmrs_concepts_2.0_20181012.sql
exit
```

After Updating CIEL , to have a better performance of the Database , Reuild the index

- Go to Administration -> Maintainace Search Index Rebuild Index

3. Update Metadata Sharing packages to the latest versions of CIEL concepts

1. Login to [mdsbuilder](#) OpenMRS instance.
2. Go to Administration -> Metadata Sharing -> Export Metadata and choose a package (repeat b. and c. for the following packages: **Reference Application Concepts, Reference Application Diagnoses, Reference Application Order Entry and Allergies Concepts**)
  - a. Click New Version.
  - b. Follow the creating package wizard leaving all defaults.
3. Click Download Latest to get the zip file for that package.
4. Export concepts using data exchange to preserve IDs for later updates of CIEL.
  - a. Login to mds builder.
  - b. Go to Administration -> Data Exchange Module / Export. Upload zip files from all you downloaded in 2. c, this generates an xml data set, copy the contents of the generated files to their respective xml files under <https://github.com/openmrs/openmrs-module-referencemetadata/blob/master/api/src/main/resources>
  - c. Add any missing calls to dataImporter.importData(...) in <https://github.com/openmrs/openmrs-module-referencemetadata/blob/master/api/src/main/java/org/openmrs/module/referencemetadata/ReferenceMetadataActivator.java> to load all files with the exported concepts.
  - d. Set ReferenceMetadataConstants.METADATA\_VERSION to a value higher by 1
  - e. Remove zip packages containing concepts (if any) from <https://github.com/openmrs/openmrs-module-referencemetadata/tree/master/api/src/main/resources> and their declarations in <https://github.com/openmrs/openmrs-module-referencemetadata/blob/master/api/src/main/resources/packages.xml>
  - f. Commit and push changes.
  - g. Release the Reference Metadata Module with the version matching the release version of OpenMRS 2.x

## User Acceptance Testing:

1. Schedule a time period after the initial release deadline for releasing the modules set under the section Begin the process of release management- 4th point. Even if some modules are not released by this time the testing can begin.
2. Setup a User Acceptance Testing(UAT) server with the OpenMRS instance having the latest released modules by deploying from CI
3. Request the Module Owners/Implementers to verify the module's functionality individually.
4. Organize events to encourage people to participate in testing the new version before it's release.
5. If the UAT results in discovery of new bugs please file them and after they are fixed , request releasing a new version for the specific module where the issue has been identified.

## Final Release Process:

1. Release all modules to be included in the distribution
  - a. Go to <https://github.com/openmrs/openmrs-distiro-referenceapplication/blob/master/pom.xml> and look for SNAPSHOT dependencies
  - b. Change `preparing.refapp.distiro.release` bamboo variable to 'true' (if it's not already set).
  - c. Release all SNAPSHOT dependencies. Verify that the released version shows up in <https://github.com/openmrs/openmrs-distiro-referenceapplication/blob/master/pom.xml>
  - d. Release all SNAPSHOT dependencies and update their versions in <https://github.com/openmrs/openmrs-distiro-referenceapplication/blob/master/ui-tests/pom.xml>
  - e. Make sure all declared versions of modules are the latest compatible versions
2. Release the distribution from CI
  - a. Go to <https://ci.openmrs.org/browse/REFAPP-OMODDISTRO>
  - b. Every build stops/pauses at the release stage (you can observe this in the left side column nearing the bottom of the page with section titled **Release** having a small right-arrow/play symbol beside it).
  - c. After verifying the artifacts trigger the release by clicking on the play/right-arrow button next to the **Release** section located to the left bottom. At this point, you have the opportunity to override the variables on release version (`maven.release.version`) and the next snapshot on the pom file (`maven.development.version`); otherwise, maven release plugin will just use its default values. This begins the process of releasing the OpenMRS 2.x version in maven and packaging the released modules and generating the `openmrs-standalone.zip`.
  - d. After a successful Release, confirm that `preparing.refapp.distiro.release` bamboo variable is set to 'false'.
  - e. Verify that `demo.openmrs.org` has been updated to the latest released version and it runs correct modules.
  - f. Verify that `modules-refapp.openmrs.org` has been updated to the latest released version and it runs correct modules.
  - g. Verify that standalone, modules and war were correctly published at <https://sourceforge.net/projects/openmrs/files/releases/> and work as expected.

## Finalizing and announcing the release:

1. Make a copy of the current "Reference Application (version)" documentation page (e.g the one for RefApp 2.5 is [here](#)), and name it for the next release. Add a note that this version is not yet released.
2. Remove a note from the current "Reference Application (version)" documentation page that this version is not yet released.
3. Edit the [OpenMRS Reference Application 2.x Implementer Documentation](#) page so that includes the "Reference Application (version)" page for the new release.
4. **Maintain the list of Contributors for the release.**
5. Create the Release notes:
  - a. Create a release notes child page in [Release Notes](#) with the release version name.
  - b. Update the [Release Notes](#) page meta tag with the new page you created.
6. Announcing the Release
  - a. Announce the release in [OpenMRS Talk](#) similar to the blog post. Please do not forget to include the list of contributors in the announcement.
  - b. Create a new case at [OpenMRS helpdesk](#) requesting to update the [OpenMRS.org](#) home page and downloads page to the latest release. Please provide links to the downloads.

In preparation for the release of OpenMRS Release X on 21/04/2017 10:00 UTC please do:

1. Update <http://openmrs.org/help/report-a-bug/> to allow bug reports for X.
2. Update <http://openmrs.org/download/> to point to Reference application at <http://sourceforge.net/projects/openmrs/files/releases/<path>>
3. Update <http://openmrs.org/download/> to point to Reference application war at <http://sourceforge.net/projects/openmrs/files/releases/<path>>

7. Release in JIRA
  - a. Request for being an administrator of the Reference Application project in JIRA from a current Administrator - Presently Darius Jazayeri is an administrator for the project. Ensure that all tickets are in the ACCEPTED state, before you release in JIRA.
  - b. Go to this link - <https://issues.openmrs.org/plugins/servlet/project-config/RA/versions> and enter the date beside the version you are about to release and click on the small settings icon indicating a dropdown that appears when you point the mouse over the list value for the version and select release.
8. **Update the Release Process wiki with the latest changes if any.**

## Maintenance Release Process:

1. Release all modules to be included in the maintenance distribution from CI, see [releasing a module from bamboo](#) and update the modules' releases in jira.
2. Release the platform if necessary from bamboo, see [platform release process](#).
3. Create a maintenance branch off the latest released tag if it doesn't exist, for instance if the current release of the reference application is 2.6 or 2.6.0, create a 2.6.x branch, below are the steps;
  - On the command line, navigate to the root of the distro project and run the commands below assuming you've already cloned the [reference application distro project](#) from git.

```
git checkout -b new_branch_name tag_name
```

- Change the version in the pom files in all the sub projects to a snapshot version e.g if the latest release is 2.6 or 2.6.0 and the branch name is 2.6.x, then the new version in the pom files would be 2.6.1-SNAPSHOT and commit the changes.
  - Then push the new branch out to origin and upstream.
4. Release the distribution from CI
    - a. Go to <https://ci.openmrs.org/browse/REFAPP-OMODDISTRO>

- b. At the top of the page in the title section, there is a drop down to switch build branches, you need to set it to the branch you created above.
- c. You need to override the maven variables for the branch build, click the **Actions** button and select **Configure branch** from the drop down, at this point you have the opportunity to set the plan variables below otherwise, maven release plugin will just use its default values;
  - i. **preparing.refapp.distro.release:** true
  - ii. **maven.release.version:** The maintenance version you intend to release
  - iii. **maven.development.version:** The next snapshot version in the pom files
  - iv. **openMRSVersion:** The version number of the platform you intend to package
- d. To start the release process, you need to click the **Run** button from the build plan page, and select **Run branch** option from the drop down.
- e. Every build stops/pauses at the release stage (you can observe this in the left side column nearing the bottom of the page with section titled **Release** having a small right-arrow/play symbol beside it).
- f. Trigger the release by clicking on the play/right-arrow button next to the **Release** section located to the left bottom, this begins the process of releasing the OpenMRS 2.x version in maven and packaging the released modules and generating the openmrs-standalone.zip.
- g. After a successful Release, confirm that [preparing.refapp.distro.release](#) bamboo variable is set to 'false'.
- h. Verify that the artifacts have been uploaded to sourceforge i.e the openmrs war file, read me file, the reference application standalone and modules zip files.
  - i. Verify that [demo.openmrs.org](#) has been updated to the latest released version and it runs correct modules.
  - j. Verify that [modules-refapp.openmrs.org](#) has been updated to the latest released version and it runs correct modules.
- k. Verify that standalone, modules and war were correctly published at <https://sourceforge.net/projects/openmrs/files/releases/> and work as expected.