

OpenMRS Release Policy

This document documents the OpenMRS Community's policy on its various software releases. The information in this document will benefit people who would like to know more about OpenMRS products, new volunteer release managers, people interested to learn more about the release process.

- Release Products/Units -
 - Types of Builds:
 - Types of Releases -
 - Software Products:
- Roles/Process Involved in OpenMRS Releases:
 - Roles involved:
 - Process:

Release Products/Units -

Types of Builds:

Snapshot builds: These are built from openmrs github project, usually once a day. These builds are intended for regular testing of the development happening on the project. These versions are labelled as SNAPSHOT. They are not intended for use by the general public.

Release Candidates: are builds that have been built and considered acceptable for a release but have not yet been accepted as a release by the project. These are built by the Release Manager for a specific release. These are intended for developers/testers to test and report any issues back to the project prior to a release. Many release candidates are possible prior to a release approval. These versions are also labelled as SNAPSHOT. Users that are not interested in development testing should wait until a release is formally approved.

Releases: are builds that have been accepted for general public release, with varying degrees of caveat regarding their perceived quality or potential for change. Releases that are intended for everyday usage by non-developers are usually available in the [OpenMRS Downloads](#) page. Releases are believed to be usable by testers and developers and often go through "alpha" and "beta" testing followed by "User Acceptance Testing(UAT)". These releases builds have the SNAPSHOT tag removed from their release versions.

Types of Releases -

Major Releases: OpenMRS Major Releases are products that have had major enhancements with multiple bug fixes. These versions have usually gone through "alpha", "beta" and "UAT" testing. These versions are denoted with two level numbers e.g. - 1.10, 2.3

Maintenance Releases: These releases are minor openmrs releases released to handle security issues and small feature enhancements. These do not usually go through any formal testing process. These versions are denoted with three level numbers e.g. - 1.10.1, 2.3.1

Non-LTS Releases: These releases are targeted at developers who want a cutting-edge release of OpenMRS Core. These versions will usually have had alpha and beta releases, but will not have gone through any formal UAT testing process. These versions are numbered just like Major or Maintenance releases, but they will be a release of *OpenMRS Core* as opposed to the Platform or the Reference Application.

Software Products:

OpenMRS releases **2 major Software products** and multiple minor modules that act as additions to the major products.

Reference Application(RA) – this is the web application (reference application) that packages OpenMRS as an Electronic Medical Record application. Reference application is the combination of Platform and multiple different modules built on Platform. Reference Application is **released twice a year**. As this is a packaging of different OpenMRS releases (that have been tested) to form a product, this usually only goes through UAT testing.

We release two different packagings of the reference application:

- (a) a runnable standalone, and
- (b) a zip file of modules that can be manually installed on the appropriate Platform version.

Platform – this is the foundational OpenMRS API and web services shared by many distributions based on OpenMRS and made available to build OpenMRS web applications. Platform is **released once a year**. This release goes through "alpha", "beta" and "UAT" before being released for general public to use.

We release two different packagings of the Platform:

- (a) a runnable standalone, and
- (b) a WAR file that can be deployed in a servlet container

Core (no Long Term Support) – sometimes a developer or organization will do a release of (only) openmrs-core, to be able to use cutting-edge code in production, without waiting for an annual platform release. These releases are targeted at developers, and *have no guaranteed support period*. In general once the next version of openmrs-core is released, we will stop backporting bugfixes and security patches to the non-LTS release.

Roles/Process Involved in OpenMRS Releases:

Roles involved:

- **Project Owner:** A Project Owner oversees and provides guidance and facilitates discussion among managers and developers in design /project management calls.
- **Developers:** Work on issues, participate in sprints and test OpenMRS before a release. These include dev/null to dev/5.
- **i18n Manager:** Manages OpenMRS translations through transifex. Creates an integration path to include the necessary translations in a specific release.
- **Infrastructure Team:** Provides support and required permissions for the necessary users. As well as handling server and various other needs.
- **Release Management Manager:** Initiates the release process. acts as a guide, support and first point of contact for any issues or guidance for a Release Manager.
Release Manager: A new volunteer is selected from the community for each major release.

Process:

1. The community discusses about major changes being pursued in the community and includes them in the Technical RoadMap under future releases. This discussion is held in the Project Management Calls by Project Owners attended by Developers and Release Management Manager and the community.
2. A request to be a volunteer Release Manager is sent out in the community by the Release Management Manager. Based on the applied volunteers for the role, a Release Manager is selected.
3. Release Manager coordinates, the developers and the community through the release process. He acts as a facilitator and organizes meetings among various different roles, teams and the community to make a release successful overseen by the project owner.
4. Release Manager based on the discussions can decide what stays and gets removed in a release. Also coordinating with the i18n Manager to decide what translations can be included in the specific release.
5. Release Manager organizes sprints with the developers to get the work completed for a release and also organizes "alpha", "beta" and "UAT".
6. After a release, Release Management Manager reviews and reflects upon the release process and improvises the release process for the next releases.