

Implementer Documentation

- [Overview](#)
- [Choosing OpenMRS](#)
- [Requirements](#)
- [Initial Setup - From Beginning to Launch](#)
- [Ongoing Support - Launch and Beyond](#)
- [Reporting and Data Analysis](#)
- [Standards](#)
- [References](#)

Overview

Implementation is the process of business analysis, system customization, server installation and configuration, user training and support, with the aim of collecting data for analysis and decision making purposes.

These documents describe how to implement OpenMRS.

1. [Getting Started with OpenMRS](#)
2. [OpenMRS from Scratch](#)
3. [Field Guide](#)
4. The [EHSDI Training Course](#), section 202, which covers implementation

There is an "Openmrs Book" / Guide online as well: om.rs/guide

Choosing OpenMRS

This should discuss how you define your problem, set goals and scope. Who are your users? What do they want? Do you have funding for equipment and support? What do you want? Will OpenMRS fit your needs? Should add a checklist and/or grid with features. ([copy from Field Guide?](#))

- [Assessing Needs](#)
- [Field Guide](#)

Requirements

Once you have chosen OpenMRS, what are the preliminary steps.

- [Project design](#)
 - [Collaborating with government and NGOs](#)
 - [Connecting multiple sites](#)
 - [Point of Care versus Retroactive data entry](#)
 - [Vertical program\(s\) versus primary or comprehensive care](#)
- [Infrastructure requirements](#)
 - [Power Infrastructure](#)
 - [Connectivity](#)
 - [Security](#)
- [Machines for entering, storing, and accessing data](#)
 - [Server requirements](#)
 - [Work stations for retroactive data entry](#)
 - [Work stations for point of care](#)
 - [Mobile devices](#)
- [Organizational requirements](#)
 - [Primary implementer](#)
 - [Long-term IT support](#)
 - [Data entry clerks](#)
 - [Data managers](#)
- [Infrastructure requirements](#): power, connectivity, security
- [System requirements](#)
 - [For server](#)
 - [For entering, storing, and accessing data](#)
- [Implementations](#)
 - The [OpenMRS Implementation Survey](#) describes several organizations' implementations
 - A few [Point of Care examples](#)
 - Case study: [Rolling out OpenMRS in Rwanda](#)
 - More examples and information on the [Projects](#) page

Initial Setup - From Beginning to Launch

- [Implementation process overview](#)
- [Installing OpenMRS](#)
 - Initial configuration is created by the Installation Wizard
- [Constructing forms: An iterative and non-trivial process](#)

- Customizing look and feel
 - [change the default theme](#) to a different color palette
 - use the [custom branding module](#) to add your own logo
 - add your own CSS files and login, registration, and dashboard JSPs by [overriding the default runtime properties](#)
- Using OpenMRS modules
 - Finding and selecting modules is covered in detail in [Getting Started with OpenMRS](#)
 - [The module repository](#) contains descriptions, downloads, and documentation links for each module.
 - [Installing modules](#)
 - If you encounter a problem after installing a module, the first step is to stop and restart it from the Administration > Manage Modules page. It will show an error message if there was a trouble during startup. If the problem persists, stop the module and recheck the symptoms. This will confirm that the problem comes from that module. See the module's documentation for troubleshooting suggestions. You can receive further help from the [Implementers mailing list](#).
- Role and User development
 - OpenMRS uses roles to [manage permissions](#). Typical roles include:
 - system administrator - configures OpenMRS, installs and updates modules, manage user accounts
 - registrars - adds new patients to OpenMRS at check-in; adds patients to programs
 - data entry clerk - creates and updates encounters after a visit
 - care providers - views patient records at point of care; creates or updates orders or encounters; assigns regimens
 - content editors - creates or updates the forms that collect encounter data; adds or changes concepts in the concept dictionary; adds or updates programs
 - Users, roles, and permissions are [managed through the administrator control panel](#) at Administration > Users.
- Patients
 - Identifiers
 - Overview: [administering identifiers](#)
 - the [ID Generation](#) and [Registration](#) modules generate identifiers.
 - The [Registration module](#) supports biometric ID through fingerprints.
 - best practices
 - Deciding what information to collect
 - patient attributes
 - caretakers
 - Registration
 - [modules](#)
 - best practices
- Concept development
 - How to translate an idea to a concept - read [EHSDI training course EH202](#), lecture 4, "Concepts, Observations, and Encounters"
 - [Best practices](#)
 - Jump start with the [OpenMRS Concept Collaborative \(OCC\)](#)
 - [Mapping - why and how](#) to map concepts
- Form development
 - Implementation options - XForms, InfoPath, HTMLForms - [HTMLform Implementers Training](#)
 - [Administering forms](#)
 - Best Practices for Translating Paper Forms - [HTMLform Implementers Training](#) - slides 22-24, "Guidelines for Managing Forms"
 - Examples at the [Form Bank](#)
- Some [best practices for implementations](#)
- [Data migration](#)
- Receiving information from external applications into OpenMRS

Ongoing Support - Launch and Beyond

- Begin with the [Administrator Guide](#).
- [System maintenance and performance](#)
 - [Server management](#)
 - [Performance tuning](#)
 - [Database](#) and server backups
 - [Upgrading OpenMRS](#)
 - [Updating modules](#)
- Managing multiple servers
 - [Data Replication 101](#)
 - [MySQL replication](#)
 - Sync servers
 - VMs
 - [Multiple instances on the same server](#) (for development or testing)
 - [Monitoring that they are running](#)
 - Monitoring the [status of java application using java melody](#)
- [Troubleshooting](#)
- User administration
 - [Administering users](#): Adding users, Disabling logins
 - [Setting role permissions](#)
- Managing metadata
 - [Concept administration](#)
 - [concept-related modules](#)
 - Default concept dictionary
 - [Millennium Villages Project](#) dictionary
 - [Maternal concept lab](#) - The Maternal Concept Lab exists to unify and amplify efforts to use mobile devices to improve maternal health primarily in resource-poor settings. This site is a resource for organizations looking to utilize mHealth for maternal health, to improve interoperability and component sharing, and to foster experience sharing about building and using such tools
 - other [concept related projects](#)

- Import/export of metadata
 - The [Metadata Sharing Module](#) allows all kinds of metadata (concepts, htmlforms, locations, roles, programs, etc.) to be exchanged between different OpenMRS installations. It supports metadata defined in the core as well as in modules provided appropriate handlers are registered. Conflicts between local and incoming metadata can be identified and resolved. The module can be used both through an API and a web interface.

Reporting and Data Analysis

- The [Reporting Module](#) is the most popular tool for generating reports. It's extensively documented; see the module's project page.
- AMPATH [exports data through SQL](#) for analysis with other tools
- [De-identifying data](#) to protect patient privacy
 - Data policy (IRB, country and/or organization restrictions, encryption)
- Data analysis in OpenMRS
 - There is a proof of concept project to use [Pentaho](#) for analysis

Standards

- OpenMRS concepts can be mapped to medical coding standards and other terminology sets, such as ICD-10, SNOMED, etc. [Read more about mapping.](#)
- Working with [HL7](#)
- The [i2b2 module](#) exports OpenMRS data in XML format

References

- [Reference implementations](#)
- [Additional resources](#)