RHEA PoC Deployment Manual

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Implemented By** | **Revision Date** | **Reason** |
| 1.0 | *Wayne N. and Desire M* | *xx.xx.2012* | Initial version |
| 1.1 | *Suranga K* | *01/07/2013* | Updates and improvements |
| 1.2 | *Suranga K* | *06/07/2013* | Added Selenium scripts |
| 1.3 | *Desire M* | *15/07/2013* | Made updates and corrections |
| 1.4 | *Hannes V* | *31/07/2013* | Added configurator module details |

Contents

# Introduction

The purpose of this document is to specify the steps required to setup a POC environment at any location, configure it, and test the necessary connections and workflows against specified HIE requirements. The first sections of this document (sections 1 – 11) describe the process for a fresh POC installation, section 12 describes the process of upgrading an existing instance from OpenMRS 1.6.5 to 1.9.0.

***See Appendix 1 for details about using the PoC Configurator module that automates much of the configuration steps described in this document***

1. Installation Requirements

* Java 1.6
* OpenMRS war file (versions 1.6.5 or 1.9.0)
* External modules and database scripts
* Apache Tomcat 6/7
* MySQL

Based on the users’ requirements, either OpenMRS 1.6.5 or 1.9.0 must be used to setup the installation. Note that different versions of modules should also be installed based on the users choice of OpenMRS version. For detailed specifications on which module versions are required for each instance, please refer to table 1.2 (below).

**Location of required configuration Files**

1. Modules and database scripts

Dropbox  RHEA Production Deployment  Infrastructure OpenMRS-PoC-Setup  “Check for appropriate version (1.6.5 or 1.9)”

This folder will contain all omod files and database scripts required to install the PoC.

1. The OpenMRS war files must be downloaded from: <http://openmrs.org/download/> (select versions 190 or 165)
2. Apache Tomcat server and MySQL must also be downloaded separately.

Note that Apache Tomcat version 6.035 is recommended for use with both OpenMRS versions 165 and 190.

# 2. Installation process:

1. Install Java
2. Install MySQL.
3. Create the POC database and assign it a user. This can be done using the following commands,

* CREATE DATABASE openmrs CHARACTER SET utf8 COLLATE utf8\_general\_ci;
* CREATE USER 'openmrs\_user'@'localhost' IDENTIFIED BY '<password>';
* GRANT ALL PRIVILEGES ON openmrs.\* TO 'openmrs\_user'@'localhost';

1. Restore the database using the provided database script. When doing so, it is important to specify the UTF8 character set.

This can be done as follows,

mysql -uopenmrs\_user -p<password>--default-character-set=utf8 -D openmrs < staging-openmrs-db.sql

1. Load the concept dictionary into the newly created database using the following command,

mysql -uroot -p openmrs < “script\_name”.sql

N.B. This step is only done when creating a fresh install of a POC instance and is not required when upgrading an existing POC instance

1. Install Apache Tomcat into the server, and copy the OpenMRS war file into it.
2. Start Tomcat. You will be navigated to the OpenMRS Installation page, and guided through a configuration wizard. Complete these, entering all the database names and credentials accurately as specified earlier. Once this process is completed you will be able to log in to the system.

For further details on installing OpenMRS, refer to,

(<https://wiki.openmrs.org/display/archive/Step-by-Step+Installation+for+Implementers>)

# 3. Required Modules and Their Dependencies

Table 1.1 (below) lists the required modules and their dependencies.

|  |  |
| --- | --- |
| Module | Dependencies |
| Appointment Management Module | MOH Trac portal module |
| Address hierarchy Rwanda module | HTMLFormEntry Module |
| Rwanda Primary Care Module | RHEA POC Adapter module, Address hierarchy Rwanda module, ID Generation module,  Name Phonetics module |
| Maternal Summary module | RHEA POC Adapter module |
| Person attributes module (OpenMRS 165 only) | None |
| RHEA POC Adapter module | None |
| HTMLFormEntry Module | None |
| HTML Form Entry Extensions for OpenMRS 1.9 Module (OpenMRS 190 only) | None |
| HIE Patient Search ext. module (OpenMRS 190 only) | None |

Table 1.1 Required modules and their dependencies

For OpenMRS 1.6.5 and 1.9.0 Installations of the POC, the following versions of the abovementioned modules should be used,

|  |  |  |
| --- | --- | --- |
| Module | OpenMRS 1.6.5 | OpenMRS 1.9.0 |
| Appointment Management Module | 0.0.7 | 0.1.9 |
| MOH Trac portal module | 0.2.2 | 0.2.4 |
| Address hierarchy Rwanda module | 1.2.4 | 1.2.6 |
| HTMLFormEntry Module | 1.10.0 | 2.0.1 |
| Rwanda Primary Care Module | 2.0.4 | 2.0.7 |
| Maternal Summary module | 0.2 | 0.2 |
| RHEA POC Adapter module | 0.5.1-SNAPSHOT | 0.5.1-SNAPSHOT |
| ID Generation module | 2.1 | 2.1 |
| Name Phonetics module | 1.3.1 | 1.3.1 |
| Person attributes module | 1.0-SNAPSHOT | Not required |
| HIE Patient Search ext. module | Does not support | 1.0-SNAPSHOT |

Table 1.2 Different versions of modules required for OpenMRS versions 165 and 190

# 4. Building external modules from the source

In order to build modules on your developer environment, you must have Java, Git and apache maven installed.

The source code for each of these modules are available on GitHub. They are hosted in either OpenMRS or Jembi repositories. It is possible to check out the code into your developer environment, and build the omod file required for installation.

**Checking out the code from GitHub**

To do this, please follow the instructions defined in,

(<https://wiki.openmrs.org/display/docs/Using+Git>)

Once the module source code had been checked out, it can be built as follows,

1. Navigate into the project folder on the command line
2. Execute the command ‘mvn clean install’

\*\* Running mvn clean install will also execute the unit tests. To save time, or to avoid failing unit tests, alternatively execute the command,

‘mvn clean install –Dmaven.test.skip=true’

This command, if completed successfully, will build the omod file in the /omod/target folder of your project. This omod file can be used as an alternative to the files provided in the drop box folders.

# 5. Required Global Properties

The following global properties must be configured on the POC instance for each of the modules to function as desired.

**• Rwanda Primary Care**

1. registration.barCodeCount = 4

2. registration.defaultLocationCode = enter location code of the facility (e.g Musha= 357)

3. registration.healthCenterPersonAttribute = health center

4. registration.insuranceNumberConcept = 6741

5. registration.insuranceTypeConcept = 6740

6. registration.insuranceTypeConceptAnswers = 6738, 6739, 6955, 6956, 6957, 1107

7. registration.maxRecentlyViewed = leave to default value

8. registration.nationalIdType = NID

9. registration.otherIdentifierTypes = check the database to determine the ID of all identifiers that can be searched against.

10. registration.parentChildRelationshipTypeId = check the database to determine the ID

11. registration.primaryIdentifierType = check the database to determine the ID

12. registration.restrictSearchByHealthCenter = false

13. registration.rwandaLocationCodes = enter name and MOH (fosa-id) location code pair for the facility (e,g Rwinkwavu:416 or Musha:357)

14. registration.serviceRequestedConcept = 6702

**• HTML Form entry**

1. htmlformentry.dateFormat = MM/dd/yyyy

**• RHEA PoC Adapter**

1. rheapocadapter.encounterType = ANTENATAL CLINIC, ANC, ANC Referral, ANC Referral Confirmation, ANC OB and Past Medical History, ANC Physical, ANC Testing, ANC Maternal Treatments and Interventions, ANC Delivery Report
2. rheapocadapter.hostname = the server url to connect to shared health record
3. rheapocadapter.username = the server username to connect to shared health record
4. rheapocadapter.password = the server password to connect to shared health record
5. rheapocadapter.sendingFacility = The Sending Facility id is the FOSAID for the facility. Each of the facilities in the district would have a different FOSAID.

• **Scheduler**

1. scheduler.username = <an admin user>

2. scheduler.password = <the password for this user>

# 6. Required Identifier Types

The following patient identifier types must be created in the POC instance prior to use.

* NID
* Primary Care ID Type
* Mutuelle
* RAMA

On how to create new patient identifier types, please refer to,

(<https://wiki.openmrs.org/display/docs/Adding+A+New+Patient+Identifier+Type>)

# 7. Rwanda Primary Care Server Side Set-Up

1. Download the module from the repository and install it.

This module depends on the idgenerator, addresshierarchyrwanda, and namephonetics modules, so these must be downloaded and installed prior to installing the primary care module. The id generator module provides an architecture for assigning ID numbers to patients during registration. The addresshierarchyrwanda module contains the entire address structure for the country of Rwanda, and provides an architecture for structured and unstructured addresses. The addresshierarchyrwanda module (since version 1.2.3) provides an htmlformentry custom tag that essentially renders its AJAX-style rwanda address fields and then records the user-selected values to the database. Finally, the namephonetics module ensures that searches by name are smart about common phonemes in the Kinyarwanda language.

2) Go through global properties and save all registration properties, in particular registration.serviceRequestedConcept, registration.rwandaLocationCodes and registration.primaryIdentifierType. The Rwanda location code is expected to be the three-digit code assigned to each health center by the Rwanda Ministry of Health. For example, the three-digit code for the Rwinkwavu Health Center is 416. The primaryIdentifierType should be the identifier type that you just created for primary care registration IDs. If there is the possibility of receiving transfer patients from other health centers where they may have received a barcode, setup the global property: registration.rwandaLocationCodes. These are pipe delimited pairs of Health Centers by name, and Health Center MoH three-digit location code. For example, a valid entry would be "Rwinkwavu Health Center:416|Kirehe Health Center:477|Rusumo Health Center|412".

1. Then, to prepare primary care-issued IDs, first make sure all id generator tables are empty, if migrating from a test database.

4) Create a Primary Care ID Type – choose the Rwandan ID validator.

5) Navigate into the id generator module administration pages. Choose the newly created primary care ID type and Local Identifier Generator

6) Here's the base character set: 0123456789ACEFHJKMNPUWXY. The first identifier base is 00001, and the length is 10.

7) Click 'view', and test by exporting several identifiers.

8) Check the relationship type used during registration to record a patient's mothers and father's name (global propertyregistration.parentChildRelationshipTypeId). If this global property is empty, check the database to determine the ID of the parent to child relationship type (or create one).

# 8. Configuring the Appointment Module

Go to OpenMRS homepage and under the useful links section select,

Manage Service Provider Link  Select manage service  Select Add new service

1. Create new service and link it to a related service concept and save

2. Select manage service provider

3. Select add new service/provider

4. Select a provider and link them to a service, select date they started in the service and save.

# 9. Encounter Types

The following encounter types must be added to the OpenMRS instance prior to use. To do so, visit (Administration -> Manage Encounter Types -> Add Encounter Type) of the OpenMRS instance.

The encounter types to be added are,

• ANC OB and Past Medical History

• ANC Physical

• ANC Testing

• ANC Maternal Treatments and Interventions

• ANC Referral

• ANC Referral Confirmation

• ANC Delivery Report

• RapidSMS Notification BIRTH

• RapidSMS Notification Maternal Death

• RapidSMS Notification RISK

# 10. Adding forms to the OpenMRS instance via HTML Form Entry

The forms used by the POC must be added to the OpenMRS instance as follows,

• Ensure that OpenMRS is running the latest concept dictionary

• Create the forms

1. On the admin page click on the manage HTML Forms link under the HTML Form entry section.
2. Click on add new html form
3. Create the forms using the names and encounter types in the table below. For each form, use the script from the dropdown folder at (Dropbox/RHEA - Jembi/RHEA Workstreams/RHEA OpenMRS/12\_09\_2012\_POC\_Setup/forms).
4. Specify the form version as 1.0, give form an appropriate description, click save.

|  |  |
| --- | --- |
| Form Name | Associated Encounter Type |
| RHEA ANC 7: Delivery Report | ANC Delivery Report |
| RHEA ANC 2: Physical | ANC Physical |
| RHEA ANC 6: Referral Confirmation Form | ANC Referral Confirmation |
| RHEA ANC 3: Testing | ANC Testing |
| RHEA ANC 4: Maternal Treatments and Interventions | ANC Maternal Treatments and Interventions |
| RHEA ANC 1: OB and Past Medical History | ANC OB and Past Medical History |
| RHEA ANC 5: Referral Form | ANC Referral |

# 11. Provider Privileges

The following is the list of privileges that a registered provider required in order to view all patient related information.

1. Add Appointments

2. Add Encounters

3. Add Observations

4. Add People

5. Can view a result of patient lab test

6. Delete Appointments

7. Edit Appointments

8. Edit FormEntry Archive

9. Edit FormEntry Queue

10. Edit People

11. Exit a patient from care

12. Form Entry

13. Manage Implementation Id

14. Patient Dashboard - View Demographics Section

15. Patient Dashboard - View Forms Section

16. Patient Dashboard - View Overview Section

17. Patient Dashboard - View PMTCT

18. View FormEntry Error

19. View Forms

20. View Locations

21. View Navigation Menu

22. View Patient Programs

23. View Patients

24. View People

25. View PMTCT patients in ANC

26. View PMTCT pediatric tests

27. View Regimens

28. View Users

29. Add ANC and follow-up information

30. Edit Encounters

31. Edit Observations

32. Manage Services and Providers

33. Manage Tokens

34. Patient Dashboard - View Appointments Section

35. Patient Dashboard - View Encounters Section

36. Patient Dashboard - View Graphs Section

37. Patient Dashboard - View Patient Summary

38. Patient Dashboard - View Regimen Section

39. Search Appointments

40. View Appointments

41. View Concepts

42. View Encounters

43. View FormEntry Archive

44. View FormEntry Queue

45. View lab results

46. View Observations

47. View Orders

48. View PMTCT

49. View Programs

50. View Unpublished Forms

51. View provider appointments

# 12. Upgrading an existing POC Instance

The upgrade process for an existing OpenMRS instance involves two steps upgrading the core OpenMRS code and the database data. The code is contained inside the openmrs.war file; the code contains a file that lists all the changes needed to go from one version to the next. When you start the OpenMRS 1.9 war file, this file is checked against what your database currently has.  If there are any updates that are needed you will be prompted to run those updates automatically (if no updates are needed OpenMRS starts normally).  The 1.9 war file has knowledge of all previous releases and the changes needed.  So if your database was being used with 1.6.5 previously, the 1.9 war file will walk your database change by change up to the 1.9 levels.

The steps for running the upgrade are as follows:

1. Stopping Modules:

* Log into OpenMRS
* Click on Administration
* Click on Manage Modules
* The modules should be listed. Next to each module, there should be a red button that says stop. Stop **ALL** modules prior to continuing the upgrade.

1. Back up the current database: run a mysql dump command to get a back-up of the current database as a precaution

* “mysqldump –uroot –p [password] openmrs > dd-mm-yyyy\_openmrs\_backup.sql ”

1. Undeploy current openmrs.war file:

* The easiest way to undeploy the current openmrs.war file is to to use the Tomcat Manager
* To manually undeploy the current openmrs.war file:

Go to tomcat home directory webapps folder

* cd ./var/lib/tomcat6/webapps/

Delete the current openmrs.war

* eg: rm openmrs.war

1. Deploy new openmrs.war file

* Get the openmrs.war upgrade file from the dropbox folder

“ Dropbox  RHEA Production Deployment  Infrastructure OpenMRS-PoC-Setup  Set-up 19  openmrs.war.”

* Copy the new openmrs.war file into the webapps folder

“cp openmrs.war /var/lib/tomcat6/webapps/”

1. Restart Tomcat: run “sudo service tomcat6 restart”
2. Approve database updates:

* Launch the OpenMRS webapp
* Log into OpenMRS
* Approve database updates

1. Install Module upgrades: After all upgrades are complete log into openmrs go to

Administration  manage modules  upload all the 1.9 compliant modules

# 13. Testing the completed POC Instance

The tests prescribed below makes the following assumptions,

* The remaining components of the HIE (CR, PR, FR and SHR) have already been installed and configured correctly.
* The abovementioned components use valid mappings which would allow it to work together with the mappings specified in the POC.

## 13.1 Test the successful entry of a patient form (ORUR01 messages)

1. login to OpenMRS
2. Navigate to the find/create patient tab
3. search for an existing female patient, if not found, then create one
4. Navigate to the ‘form entry’ tab of the patient dashboard
5. Select a form, and enter appropriate data into it
6. Click ‘enter form’ to submit the form
7. Now, navigate to the HIE transaction console (ex : <http://hie.jembi.org/auth/login>).
8. The outcome of each transaction submitted to the HIE should be displayed in a separate row of the transaction console.
9. Verify that the transaction was completed successfully. Now, navigate into the transaction detail page, and view the components/data of the said transaction. Ensure that they are accurate, and matches with the data that you entered.

If configured correctly, this test would result in a successful outcome.

**Testing the POC installation independently from other components of the HIE**

Assuming that the other components of the HIE are not yet available, and that the poc must be tested as a standalone service, a partial testing scenario can be implemented as follows,

From the previously mentioned test scenario, complete test steps a to g.

Since the remaining components of the HIE are not configured, this transaction would have ended in failure.

However, you can navigate into the transaction details page of the HIM management console, and view if the transaction data matches with those specified when entering the patient form. This is especially useful to confirm whether the POC was able to create a valid ORUR01 message representing the form data.

## 13.2 Test the successful creation of a patient (ADTA05 messages)

1. Login to OpenMRS
2. Navigate to the find/create patient tab
3. Create a new patient by providing names/ identifiers/ addresses and other optional data
4. Click ‘enter’ to save the patient
5. Not navigate to the HIE transaction console (ex : <http://hie.jembi.org/auth/login>).
6. The outcome of each transaction submitted to the HIE should be displayed in a separate row of the transaction console.
7. Verify that the transaction was completed successfully. Now, navigate into the transaction detail page, and view the components/data of the said transaction. Ensure that the completed ADTA05 message is accurate, and matches with the data that you entered when creating the new patient.

If configured correctly, this test would result in a successful outcome.

# 14. Automated testing of the POC instance

Once the POC installation is complete, it can be tested using a set of four selenium scripts prepared for this purpose. These scripts test,

1. The defined global properties for the POC
2. New data types created for encounters and patients
3. The creation and submission of a form

For a detailed explanation on what selenium is, please refer to,

(<http://docs.seleniumhq.org/>)

Using the selenium scripts require the installation of Firefox. For detailed instructions on how to install selenium to operate the test scripts, refer to,

(<https://www.cs.drexel.edu/~spiros/teaching/SE320/slides/selenium.pdf>)

**Authentication for selenium scripts**

Since different installations of the POC would use different username/passwords, the selenium script would need to use these in order to execute successfully. Therefore, it was decided that the script should not include any authentication data. Instead, the user should manually log into the POC instance under test before executing the script, thereby bypassing the need to include authentication details in the selenium scripts.

The four scripts created for this purpose are supplied as attachments to this document.

These files are in the form of \*.html files, and contains the commands to be executed and the validations to be made. To run them, simply open the Firefox browser, navigate to Tools  Selenium IDE, and open the scripts in the IDE.

Now, you will note that the default base URL of each test script has been set to <http://poc.jembi.org:8080/> (refer to figure 1.1 below).

To edit this, click on the ‘Source’ tab of the IDE, and edit the URL to point to your own implementation.

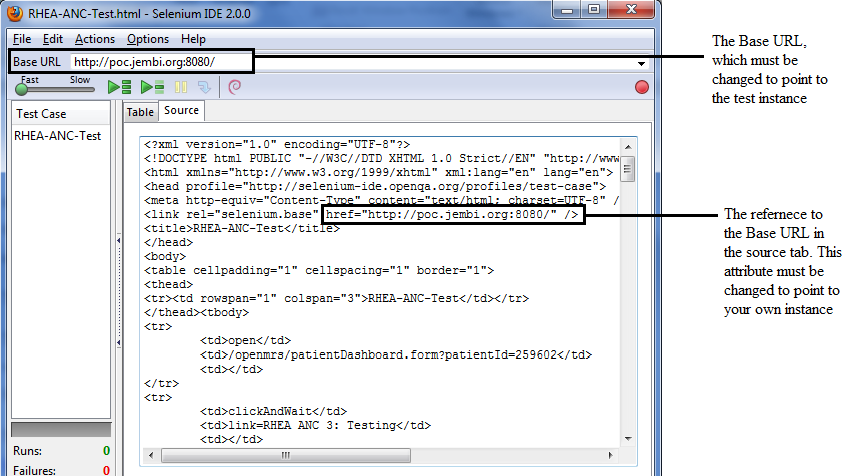


Figure 1.1 Editing the selenium script to point to your base URL

Now, execute the edited scripts, and verify that the tests complete successfully.

# Appendix 1: Using the PoC Configurator Module

The PoC Configurator module is now available for setting up the configuration for a PoC instance. The module is available at <https://github.com/jembi/rhea-poc-configurator>

This module will automatically set up the configuration for sections 5, 6, 9, 10 and 11 in this manual. It can also run a connection and authentication test against the OpenHIM. It will also run validation checks to ensure that all the required concepts needed for the ANC forms are setup on the system, with mappings available.

To use this module, simply load the module as normal, and then navigate to the page **Configure System** under the admin page. There you can enter several required details for the installation, and then simply click on the **Perform Configuration** button to run the configuration process. The tests can be run from the **Test System** page.

RHEA PoC Server Deployment Manual

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Implemented By** | **Revision Date** | **Reason** |
| 1.0 | *D. Futerman* | *02/08/2013* | Initial version |

## Introduction

This document describes the process used during the August 2013 RHEA implementation sprint to update and deploy a series of RHEA PoC servers and implement them on site. In this instance, the servers have been previously set-up and configured to have OpenMRS 1.9 running. The process is then to add/upgrade any required modules, complete the process of configuration and testing, and ensure the server is able to connect to the OpenHIM. This process is done before implementing the server on site, however some configuration and testing should be repeated once installed on site, to verify the implementation.

## Assumptions

The following is assumed before starting the server update:

* The server has been upgraded to Ubuntu 12.04.
* OpenMRS 1.9 is installed and running.
* All MoH-related modules have been loaded with the latest version

## Backup and Clean-up of Server

1. ssh into the server using the *moh-admin* user

$ ssh moh-admin@[server-ip]

1. Back-up the OpenMRS database:
   1. Check for an existing script in the home directory:

$ ls ~/dump\_openmrs\_db.sh

and execute the script:

$ ./dump\_openmrs\_db.sh

* 1. If the script is not on the server run the following mysqldump command:

$ mysqldump –uroot –p [password] openmrs > [dd-mm-yyyy]\_openmrs\_backup.sql

* 1. Execute the script:

$ ./dump\_openmrs\_db.sh

1. Back-up the OpenMRS config directory:
   1. Copy the OpenMRS config directory:

$ cp –R /usr/share/tomcat6/.OpenMRS ./openmrs-conf-dir-[dd-mm-yyyy]

* 1. Compress the config directory:

$ tar –zcf openmrs-conf-dir-[dd-mm-yyy].tar.gz openmrs-conf-dir-[dd-mm-yyyy]

1. **NOTE**: If the config directory and/or database backup are moved off the server onto a local machine or flash disk, they need to be removed from these external sources once the process is complete.
2. Run the database cleanup script:
   1. Check on the server for the following script:

$ ls ~/mysqlExecutable.sh

* 1. If the script is not on the server, copy the file *cleanup\_ubudehe.sh* to the server using scp. The file should be located in the following Dropbox folder:

RHEA Production Deployment/Infrastructure/OpenMRS-PoC-Setup/Set-up 19/

$ scp [source-dir]/cleanup\_ubudehe.sh moh-admin@[server-ip]:~/

* 1. Run the script

$ ./mysqlExecutable.sh

or

$ ./cleanup\_ubudehe.sh

## Load Modules onto Server

1. Copy all required OpenMRS modules across to the server:
   1. The modules should be located in the following Dropbox folder:

RHEA Production Deployment/Infrastructure/OpenMRS-PoC-Setup/Set-up 19/Modules/Jembi-Modules/

* 1. Copy the modules to the relevant folder on the server using scp:

$ scp [source-dir]/\*.omod moh-admin@[server-ip]:/usr/share/tomcat6/.OpenMRS/modules/

1. Copy the Auto Config Module to the server:
   1. The module should be located in the following Dropbox folder:

RHEA Production Deployment/Infrastructure/OpenMRS-PoC-Setup/Set-up 19/Auto Config Module/

* 1. Copy the module to the relevant folder on the server using scp:

$ scp [source-dir]/rheapocconfigurator-0.5.omod moh-admin@[server-ip]:/usr/share/tomcat6/.OpenMRS/modules/

## OpenMRS Configuration

1. Log in to OpenMRS as an administrator.
2. Set up the Implementation ID:
   1. Go to the Administration Page.
   2. Under the *Maintenance* Tab, select the *Set Implementation Id* link.
   3. *Implementation Name / Id*:
      1. From the Administration page, under the *Locations* tab, select *Manage Locations* and find the relevant location/site.
      2. *Implementation Name* will be the same as the name of the location.
      3. *Implementation Id* will be *rwanda*+*FOSAID*, e.g.

rwanda534

* 1. *Pass Phrase*: This should be *firefox*.
  2. In practice, there were issues where these settings did not work, as the implementation Id had already been used. In this case, it is worth contacting someone from OpenMRS (such as Michael Downey), to try to reset the implementation Id.

1. Run the PoC Configuration Module configuration:
   1. Go to the Administration Page.
   2. Under the *RHEA PoC Confogurator Module* tab, select the *Configure System* link.
   3. *OpenHIM Host / Username / Password:*
   4. This information is available in the file *interface-pwds*. This should be located in the following Dropbox folder:

RHEA Production Deployment/Infrastructure/OpenHIM/

* 1. *Scheduler Username / Password*:
     1. Create a user within OpenMRS with the *System Developer* role and use their credentials here.
     2. In this instance the user *jembi* was created with an auto generated password.
  2. *Location ID / Name / FOSAID*:
     1. From the Administration Page, under the *Locations* tab, select *Manage Locations* and find the relevant location/site.

1. Run the PoC Configuration Module tests (in future, this functionality will be moved to the RHEA PoC Adapter Module):
   1. Go to the Administration Page.
   2. Under the *RHEA PoC Confogurator Module* tab, select the *Test System* link.
   3. The *Test Patient NID* should be auto-completed with a relevant patient to use to test the system.
   4. Click on the *Perform Tests* button to run a series of tests against the system.
   5. If *Connection & Authentication Test* fails:
      1. This means the system is unable to connect to the HIM
      2. In this case, the first steps to locate the source of the issue would be to check the logs, as well as the HIM Error Management Console.
   6. If *Form Concepts Validation* fails:
      1. The list of missing concepts will be shown on screen.
      2. The first step to resolve this issue is to make sure the most recent concept dictionary has been loaded into OpenMRS. This information will be in the most recent concept dictionary file *Concept\_Dictionary\_[yyyy-mm-dd].sql*. Copy this file to the server using scp. The file should be located in the following Dropbox folder:

RHEA Production Deployment/Infrastructure/OpenMRS-PoC-Setup/Set-up 19/

$ scp [source-dir]/Concept\_Dictionary\_[yyyy-mm-dd].sql moh-admin@[server-ip]:~/

* + 1. Pipe the sql file into the openmrs database:

$ mysql –uroot –p [password] openmrs < Concept\_Dictionary\_[yyyy-mm-dd].sql