# ReportingCompatibility Module

## About

The reporting compatibility module was written for the 1.5 release of OpenMRS. It is so named because it contains pages and features that were previously included into OpenMRS core code itself.

The module adds a top-level link called "Cohort Builder" and an admin section called "Reports" when it is installed.

## Downloads

- Download omod module file from module repository: https://addons.openmrs.org/#/show/org.openmrs.module.reportingcompatibility
- View/download source code from svn: http://svn.openmrs.org/openmrs-modules/reportingcompatibility/trunk/

## Features

### Cohort Builder

A cohort is a list of patients. This can be a static list of patient_ids that never changes (like a cohort of patients in a study) or a dynamic list of patient_ids that is regenerated every time (like a cohort of patients of 'all male patients over age 65').

TODO: finish this section (probably deserves its own page with screenshots, etc)

### Data Exports

A data export is a method of turning the key-value pairs of observations and patient data into a one-row-per-patient csv file. This comma separated file can be opened with excel or some other data analyzing tool for reporting purposes.

The output file will always only have one row per patient. If you need to see *all* observations of a certain concept for a patient, you might want to use the row-per-obs data export below.

**To create/run a data export:**

1. Click Manage Data Exports on the Admin page
2. Click Add New Data Export
3. Fill in a name and description
4. Click to the 'Define Columns' tab
5. Click 'Add another column'
    a. Use the dropdown box on the 'Simple Column' tab to choose OpenMRS Identifier Number
6. Click 'Add another column'
    a. Click the 'Concept Column' tab
    b. Click 'select' next to Concept
    c. Search for and select 'Weight (KG)'
7. Click 'Save Data Export'
8. Check the box next to your newly defined export and then click the button for 'Generate'

The data export will be cached for your locale until you regenerate it again. If there are other users on the system in your locale they will be able to see and download the cached export.

You will need to regenerate (step 8 above) EVERY time you modify a data export

TODO: Explain the 'Define Cohort' tab

TODO: Explain the other parts of the 'Concept Column' tab

### Row-per-obs Data Exports

Similar to the Data Export above except it if a patient has multiple observations stored for the selected concept, then a new row for each observation will be printed.

TODO: finish this section

### Schema Reports

??XML Reports

### Cohort Reports

Cohort Reports build on the features listed above to provide a more user-friendly interface for defining a specific type of report - one which contains indicators with values that represent the number of patients in a particular Cohort.

**To create/edit a Cohort Report**:

The first stage in creating a Cohort Report is in creating your Cohorts. This is done through a combination of Cohort Builder and Manage Patient Searches. As an example, we'll create a Cohort for Adults:

1. Cohort Builder
   a. Go into Cohort Builder and click on the Patient Attributes tab
   b. Next to age, enter a minimum age of 15, and click search. This will return all patients at least 15 years old.
   c. To the right of the results here, click on the Save icon, and name this "Adults"
2. Manage Patient Searches
   a. Click Manage Patient Searches from the Admin page
   b. Click into the "Adults" patient search that you created. You'll see that this is defined as a 'org.openmrs.reporting.PatientCharacteristicFilter' with an attribute of "minAge" set to 15.
   c. Now, to know exactly what other attributes are available, you may need to look at the Java code. In this case, looking at the PatientCharacteristicFilter java class, we can see that the following are available to set: gender, minBirthdate, maxBirthdate, minAge, maxAge, aliveOnly, deadOnly, effectiveDate. We can add in any of these that we want at this point - most should have been available to set via our search in step 1 with Cohort Builder, but others, such as effective date, need to be added here.
   d. Under "Add a new search argument:", add in:
      i. Name: "effectiveDate" -- this must match exactly with the property name from PatientCharacteristicFilter
      ii. Value: ${date} -- the important thing here is that the value contains ${ }. This tells the reporting engine that this is a parameter, not a fixed value
      iii. Java Class: "java.util.Date" -- this should be set to the Java class name of the parameter (in this case, it's a Date)
3. Manage Reports
   a. Click Manage Reports on the Admin page
   b. Click Create Cohort Report (to create), or click on the name of an existing Cohort Report (to edit)
   c. Fill in a name and description and save. Once saved, you will have access to the next workflow steps.
   d. Click to the 'Parameters' tab
   e. Click 'Add new parameter' for each parameter your report contains. Typically you will want "name" to contain no spaces. In our case we want: Name: reportDate; Display Label: Date of the Report; Type: Date
   f. Click to the 'Indicators' tab and follow the instructions as listed. In our case, we're just going to add one indicator using the saved Patient Search we created above:
      i. Indicator: 1
      ii. Description: Number of adults on the date of the report
      iii. Click into the text field for Indicator Specification. Then click, from the right hand menu, the patient search you want (in our case, it should be [Adults|effectiveDate=${?}].
      iv. Replace the "?" in the above indicator specification with the parameter name from your report. In our case, it would change to: [Adults|effectiveDate=${reportDate}]
4. Run Report
   a. Click Run Reports on the Admin page
   b. Click on the Test Report link
   c. Try entering different values for the "Report Date" parameter and clicking Run Report for Cohort report web preview. It should work!

## Release Notes

- **1.5.8**
  - Bug in change language - RCM-80
  - Include Patient Attributes when doing a data export - RCM-15
  - Removing extra plus sign in error message - RCM-86
- **1.5.7**
  - Fixed Row per obs data export editing: RCM-66
- **1.5.3**
  - Changed dependencies so that there are no 1.8 dependencies and is compatible with 1.5.x and up.
  - Copied the PatientSetService/Impl/DAO/HibernateDAO into the module as the ReportingCompatibilityService, changed references in DataExportFunctions to refer to this service
  - Changed the implementation of 2 methods: getPatientPrograms and getDrugOrders, to improve DataExport performance - RCM-46
- **1.5.2.1**
  - Added NPE catch in report controller in case someone didn't provide a reportId - RCM-45
- **1.5.2**
  - Sped up data exports and forced them to use less memory (less OOMs) - RCM-20 (requires 1.8 for full speed up effect)
  - Fixed some spelling errors in translation messages
- **1.5.1.5**
  - Fixed a few messages.properties translation elements
  - Significant speedup for program and drugorder data exports
  - Significant speedup for identifier data exports (1.8+ only)