Configuring ReDBox 2.0

Updated October 8, 2018

This is a workshop guide for configuring a preview version of ReDBox 2.0 on a Linux VM. It was originally written for the Open Repositories 2018 Introducing ReDBox workshop.

- This guide assumes a ReDBox preview 2.0 installation has been completed.
- Assumes commands are entered in the VM shell using Putty or similar.
- Also assumes you are not logged in as root but have sudo access. Note if you installed ReDBox on your own server, then you have root.
- Get your VM's IP address (e.g. `hostname -I`) or assigned URL.

1. Set Up

1.1. Get a local text file editor

If you don't have an editor of choice, then notepad++ is absolutely fine. Windows notepad isn't.

https://notepad-plus-plus.org/download/v7.5.6.html

1.2. Get a local CSV file editor

Excel or similar will do. A free CSV editor you can use is Data Curator, which runs on Mac and Windows

http://www.data-curator.io/

1.3. Get your SSH Key or Login

If you are trying one of the workshop instances, the login/password will be workshop/workshop.

1.4. Get an SSH app

Putty. Set up instructions will be on screen.

https://www.chiark.greenend.org.uk/~sgtatham/putty/

1.5. Get a file transfer app

Mac users can use Cyberduck

For Windows, download and install winscp (www.winscp.net). Allow default options for install.

Launch WinSCP (hint, use the same key you used for ssh/putty).

1.6. Logging in

There are a few logins to know:

Portal.

- This will be at http://rbX.redboxresearchdata.com.au where X is the number assigned to you.

- Login/Password: admin/rbadmin

SSH/WinSCP

- This is the VM login.
- Login/Password: workshop/workshop
- Sorry, you do not have sudo.

Mint

- This will be at http://rbX.redboxresearchdata.com.au/mint where X is the number assigned to you.
- Login/Password: admin/admin
- You won't need it for the workshop, but you can play with Mint afterwards.

GitLab

- A GitLab login is required for the provisioning demo. On your instance the login will be rbX where X is the number assigned to you. Password is workshop.
- You will only need to give your GitLab credentials once.

2. Loading Mint

2.1 Overview

Detailed instructions on loading mint can be found at:

The Mint Name Authority runs independent of ReDBox and can integrate with other systems via the Mint API (REF).

You can log in to mint via its web portal: http://rbX.redboxresearchdata.com.au/mint or http://[hostip]/mint. Demo login/password is admin/admin.

2.2. Harvesting to Mint

In order to load data into Mint you'll need to undertake what we call a "harvest". To run a harvest you use the 'tf_harvest' script in the 'server' directory. The control scripts page has some broad information about these scripts, but for the purposes of this page, we'll assume that you are on Unix/Linux running the 'tf_harvest.sh' script:

```
cd /opt/mint/server
./tf_harvest.sh
```

Using the script in this way (without a parameter) lists the available data sources.

By default Mint is configured for the following static data and sample data.

2.3. Static Data

These data sources are static, in that they are unchanging and the same for everyone. They are not assigned *Handles*, and do not have *rendered templates*. They are used in support of autocomplete features for ReDBox form controls. Note that ANZSRC codes are relevant to Australian and New Zealand usage only. This does not mean ReDBox has to use these codes.

- Languages: This is the set of ISO 639-2 languages (485 records)
- ANZSRC_FOR: The ANZSRC codes for Fields of Research (1417 records)
- ANZSRC_SEO: The ANZSRC codes for Socio-Economic Objectives (989 records)

2.4. Sample Data

This sample data is drawn from public data used by the University of Newcastle, Australia (UoN), as well as fabricated people and groups. Most of this data is configured to seek access to a Handle server per object, and to generate *OAI_DC* and *RIF-CS* templates. In live usage it would be replaced with your institution's 'real' data, and there are notes on adding additional data as well.

- Parties_Groups: 5 fake university business units. Handles are created and RIF-CS 'party' templates are rendered.
- Parties_People_Multi: 7 fake people. Handles are created and RIF-CS '*party*' templates are *rendered*. Multi means they can be assigned to multiple groups.
- Funding_Bodies: A selected sample of 36 project funding bodies from UoN's data. No Handles or templates.
- Activities_*: 3 different data sources for project data (filtered for UoN). Handles are created and RIF-CS 'activity' templates are rendered. ~397 records

2.5. Harvest Configuration

Each of these harvests are described in a harvest configuration, stored in the '/opt/mint/home/harvest' folder. In this folder you'll find two types of files:

.json - these are the configuration files for the various defined harvests.

.py - these are rules files for the harvests, they dictate how to index that data.

2.6. Harvest Sample Data

To try out harvesting the Language data, run:

```
./tf_harvest.sh Languages
```

This will harvest the ISO 639-2 language codes into the system. Give them a few minutes to process (you can monitor the logs) and they will become visible in the web interface.

2.7. Adding Data to Mint

In this example you will add your organisational unit to Parties_Groups and your details to Parties_People_Multi. These files have already been harvested on your preview ReDBox, but we will reharvest to demonstrate how to add records.

Using your file transfer app navigate to /opt/mint/home/data and open Group_Sample.csv (note, you may need to pull it across to your local folder and open it).

Add a Group record for yourself (i.e. your real or imaginary institution/research group). Save it (or pull it back if you couldn't open it in-situ).

Repeat the above steps for People_Multi_Sample.csv, adding yourself and linking yourself to the new group you created via the GroupID_1 column.

Why these files? The harvester is configured in Parties_Groups.json and Parties_Peoples_Multi.json. These file names are identified in there. Using these files as a template you can configure to import anything.

Harvest the updated files into Mint.

```
./tf_harvest.sh Parties_Groups
```

```
./tf_harvest.sh Parties_People_Multi
```

You can now see your group and yourself added to Mint via the portal.

2.8. Updating a Record

Updating a record is easy - just alter the value in the CSV and reharvest the CSV. Mint will actually update every field in the CSV - no checking is done, fields are "updated" even if they haven't changed. Consider a faster differential update where the CSV only contains the records that have changed.

2.9. Deleting a Record

You cannot delete a record via the harvest. **Deleting a record runs the risk of breaking the Curation Model.** See Curating Linked Data for a discussion on this. You can delete a record via the portal as admin, but this is strongly discouraged and you must make sure that you

aren't breaking any relationships with other objects (in Mint or ReDBox) or feeds to other organisations.

2.10. Mint Documentation

The Mint documentation is still on our ReDBox 1 website, here: http://docs.redboxresearchdata.com.au/documentation

Keep an eye on https://www.redboxresearchdata.com.au/documentation/ for changes.

3. Configuring the ReDBox Portal

3.1 Configuring ReDBox

Where're the config screens? In ReDBox 2.1.

For now, WinSCP (or CyberDuck) will do the job

3.2 Sails.js

The ReDBox Portal is built on the Sails.js framework and therefore takes advantage of the highly extensible configuration framework provided.

Please see the Sails.js documentation for more information on its structure and configuration options.

3.3. Environment Configuration

It is possible to use environment variables to modify configuration, this is particularly useful when running the portal in a containerised environment such as docker. Please see Sails configuration document for more information.

Note the environment is set by setting the NODE_ENV environment variable in `/opt/redbox-portal/ecosystem.json`.

For now, it is possible to override environment configuration for the current environment (e.g development, test and production) by creating or modifying the environment specific config file in the location `/opt/redbox-portal/config/env/<environment-name>.js`. Environment specific variables would include things like wait periods, selecting authentication methods, etc.

3.4. Configuration Files

Configuration files are in `/opt/redbox-portal/config`. Configuration files that are explicitly important to ReDBox are:

1. record.js manages configuration around record management and the portal's interaction with the ReDBox Storage and Mint services
2. auth.js manages configuration around authorisation and authentication.
3. emailnotification.js manages configuration of email notifications
4. recordtype.js manages definitions of record types and their key search fields.
5. form.js manages form configuration

Often these files will reference additional configuration files. Open `/opt/redbox-portal/config/form.js` in a text editor and you will see it points to a Sails.js config file for each of the record types and workflows for that record type. e.g.:

```
var rdmpForm = require('../form-config/default-1.0-draft.js');
```

Open `/opt/redbox-portal/form-config/default-1.0-draft.js`

- This is the form config file for the 'draft' RDMP workflow

- Draft means the plan is in Draft status. In this deployment the plan will remain in perpetual draft, but you can add workflows for approving, activating, archiving (the plan).

3.5. Record Types

Record types are defined in `/opt/redbox-portal/config/recordtype.js`.

There are three default record types:

- rdmp – A Research Data Management Plan.
- dataRecord – A data record, a metadata record for research data.
- dataPublication – A Data Publication Record, publication record for the data record.

3.6. Populating Fields from Mint

Using `/opt/redbox-portal/form-config/default-1.0-draft.js`, find the CI (Chief Investigator) lookup in the Peoples Tab.

Search for `contributor_ci`. The Mint lookup is then identified by:

```
vocabId: 'Parties AND repository_name:People',
sourceType: 'mint',
```

Which links it to two lines in `/opt/mint/home/harvest/Parties_People_Multi.json`

```
"repository.name": "People",
"repository.type": "Parties"
```

Basically Mint has indexed the data coming from the Mint harvest step into a searchable database (SOLR). The lookup then queries SOLR to retrieve the entries for Parties/People.

3.7. Configuring Forms

The following is a bare-bones example to configure a form to include an additional field. It is not a comprehensive description.

Open `/opt/redbox-portal/form-config/default-1.0-draft.js`

Find `DataAnalysisResources`

Copy and paste (repeat) the class block.

We will create a free text field for describing licenses of the analysis tools. Change the repeated block fields as follows

```
name: 'vivo:Dataset_redbox:DataAnalysisLicenses',
label: '@dmpt-vivo:Dataset_redbox:DataAnalysisLicenses',
help: '@dmpt-vivo:Dataset_redbox:DataAnalysisLicenses-help',
```

Save the file.

Open `/opt/redbox-portal/assets/locales/en/translation.json`

Find `DataAnalysisResources`

Copy and paste (repeat) the lines starting with:

```
"@dmpt-vivo:Dataset_redbox:DataAnalysisResources":
  "@dmpt-vivo:Dataset_redbox:DataAnalysisResources-help"
```

Change the fields to follows

```
"@dmpt-vivo:Dataset_redbox:DataAnalysisLicenses":
  "@dmpt-vivo:Dataset_redbox:DataAnalysisLicenses-help"
```

Change the Text and Help Text as well. e.g. "Licensing for Software/equipment used to manipulate/analyse the data"

Save the file.

Restart the portal. In SSH cd to `/opt/redbox-portal` and then

```
pm2 restart ecosystem.json
```

Log in to ReDBox and navigate to Create RDMP -> Data Collection and Analysis.

Your new field should be in the form.

3.8. Configuration Documentation

This is a really quick look at how to configure the ReDBox portal, but extended documentation on configuring the portal can be found at:

https://redbox-mint.github.io/redbox-portal/