

Figure DB Schema

**Introduction**

Only UTF-8 encoding for harvested files.

Stereotypes above the table names indicate the component who owns the writing permissions on the table, for instance the table Aggregator can be modified by Repox component while the table URIS can be modified by Sip component. Some tables are shared and can be modified by both, see table descriptions.

**DataSet table**

The value of the “Language” field is the three letters ISO code (http://en.wikipedia.org/wiki/ISO\_3166-1\_alpha-3)

About “Type” field: currently ESE is the only accepted metadata format but in the future we almost certain extend the harvesting to other formats such as LIDO. This field will contain the name of the metadata format.

**Provider table**

Country is two letter ISO code 3166-1-alpha2 (http://en.wikipedia.org/wiki/ISO\_3166-1\_alpha-2)

The “Type” values (for the moment) can be:

Museum

Archive

Library

**Request table**

This table identifies a specific harvesting for a given data set. It also indicates if this request can be sent to production.

During the loop where Repox is parsing the file the request should be in the state “under construction”, and Repox may abort the request and set the status accordingly, when this process is done the state must move to “import completed” and after this point Repox cannot change the data.

The field “status” can assume the following values:

under construction - repox is creating a new request

import completed - repox ready, sip can take control when ready

aborted  - something went wrong

sip processing - sip has found the request repox may not any more delete the request

completed - all records for this request completed

ready for production - \_always\_ a human decision!

**RequestMDRecord table**

This table links all the metadata records belonging to a given request.

Repox can only insert records in this table whose request status is “under construction”, if the request is aborted don’t remove links from this table, this task will be done manually.

**MDRecord table**

This table contains the original record and all its refinements. ContentHash will be provided by the Harvester and is used to identify the ESE record. The Repox may only insert new MDRecord and cannot change or delete existing items.

The two fields that must be actually filled are “Source\_data” with the delivered content dump and content\_hash, all other fields should have the default values.

The algorithm for generating content\_hash is: sha256 hash with all the linefeeds stripped.

The field “status” can assume the following values:

created – (default) the record is created but not yet processed in any way

idle - nobody is touching it, waiting for more checks

processing - a checker is working on this record

problematic - something went wrong, human intervention might save the record

broken - record is invalid, some check decided this ese is not acceptable

verified - all checks succeeded, could be sent to production

**URIS table**

A Uri record is created by a thread and end as completed or failed

The field status can assume the following values

created - (1) the record is just created not processed

uri verified - (2) the uri responds and returns an OK

object downloaded - (3)

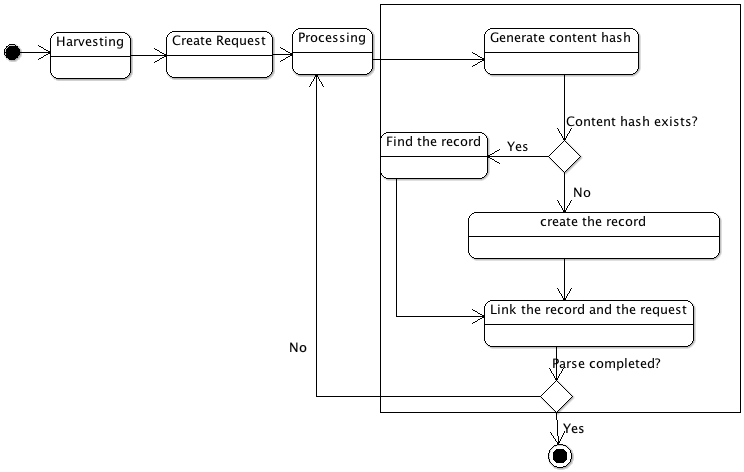
full\_doc generated - (4)

brief\_doc generated - (5)

completed - (6)

failed - (7) something went wrong, see err\_msg for details in this msg also should be logged what step failed

**Repox statechart**



**Overall Description**

-For each ESE record:

Generate content-hash

- if item identified by content-hash is not found

    - Create new md record, fill it with content-hash and xml dump of ese record

       other fields can be left as defaults

       relevant fields:

         Source\_data - xml dump

 Content\_hash

- Link request and mdrecord identified by content-hash in link table