DBMS Import framework

This functionality has been introduced in DSpace-CRIS 5.6.0 and it is considered experimental. It hasn't tested extensively so we like to receive feedback from the community

DSpace-CRIS provides a lot of way to import, update and manipulate both native dspace objects than CRIS objects in bulk. Other than the ones offered by a basic DSpace it is possible to use (also from the UI) excel files (CRIS Objects) or adhoc simplified database tables (currently only DSpace items) to perform operation over the data.

future plan

We hope to extend the framework to perform operations directly over the CRIS entities as well. It should be noted that right now operation on DSpace item can result in creation or update to related CRIS entities automatically as by the Filler functionality

The following database tables have been introduced:

- imp_record: contains information about the operations to perform. Each row represent a specific operation on a single item
- imp_metadatavalue: contains all the metadata associated with an item that need to be created or updated (optional)
- imp_bitstream: contains all the information related to bitstreams to attach / replace in the item (optional)
- imp_record_to_item: this table is populated by the framework to track the result of creation action so that subsequent operation over the same origin record will result in update instead of duplication of entries

To elaborate the imp_* tables you need to run the following script

org.dspace.app.cris.batch.ItemImportMainOA

-p Send the email for the in archive event to the authors, coauthors, etc. - the workflow email are EVER disabled
-E BatchJob User email
-x Indexing disabled (improve performance)
-n Summary EMail disabled (improve performance)
-b Delete bitstream related to the item in the update phase (you need to provide details about the new bitstream or the bitstream to keep in the imp_bitstreams table)
-m List of metadata that are cleanup before to perform the operation. By default all metadata are delete, specifying only the dc.title it will obtain an append on the other metadata. Use this option many times on the single metadata e.g. -m dc.title -m dc.contributor.*
-s Invert the logic for the -m option, using the option -s only the metadata list with the option -m are saved (ad es. -m dc.description.provenance) the other will be delete
-S muted logs
-t Threads numbers (default 0, if omitted read by configuration). Very experimental.

imp_record

- **imp_id:** the unique ID used to link the operation with the additional data in the other imp_* tables
- imp_record_id: an unique ID for the record in the external source system. This is used togheter with the imp_sourceref to guarantee that subsequent operation over the "same" source record will be performed always on the same DSpace object without forcing the external system to know about DSpace-CRIS
- imp_sourceref: an unique acronym for the system that have provided the data
- imp_eperson_id: the id of the eperson to use to perform the action
- imp_collection_id: the collection where create the item if relevant
- status: can be one of the following values:
  - p = workspace
  - w = workflow step 1
  - y = workflow step 2
  - x = workflow step 3
  - z = in archive
  - g = withdrawn
- operation: can be one of update or delete. Update is used also for record creation
- integra: not used, to be revisited to manage versioning
- last_modified: must be empty. It will be populated when the record is used
- handle: only for creation of new item is it possible to force a specific handle , otherwise the system will assign a new one in the usual way

imp_metadatavalue

- imp_metadatavalue_id: an unique id sequence generated
- **imp_id: link to the imp_record main table**
- imp_schema: the shortname of the schema (dc, dcterms, etc.)

- imp_element: the element
- imp_qualifier: the qualifier
- imp_value: the textual value of the metadata
- imp_authority: the authority key if any for this value. Since 40eeb989c4354731c0ee3fce6e80d6df64b80c94 the authority and confidence values are used by default as is forcing the metadata creation to skip the getBestMatch method of the authority framework. To guess a potential match it is possible to use the value, case insensitive, **[GUESS]**, to force the use of the authority framework getBestMatch method.
- imp_confidence: the confidence of the authority if any (600 mean accepted match)
- imp_share: not used, for future use
- metadata_order: used to sort the metadata values within the same schema.element.qualifier to insert/update
- text_lang: the lang for the metadata value (en, it, etc.)

imp_bitstream

- imp_bitstream_id: an unique id sequence generated
- **imp_id: link to the imp_record main table**
- filepath
- description
- bundle: the name of the Bundle where put the bitstream (ORIGINAL, TEXT, etc.)
- bitstream_order: to sort the processing of the rows
- primary_bitstream: flag to mark the bitstream as primary
- assetstore
- name
- imp_blob: the content of the bitstream (alternative to filepath)
- embargo_policy: can be one of:
    - 0 --> mean open access
    - 1 --> embargo (need to use also the embargo_start_date column)
    - 2 --> assign a READ policy to epersongroup ID 2 (you need to create a epersongroup with such ID for "authorized users")
    - 3 --> assign a READ policy only to the administrators group
- embargo_start_date: to use as start date of Anonymous READ policy when embargo_policy = 1