Service deliverable technical specification: CU Digital Repository – curation task for automated item mapping

The goal of the programming services requested as a subject of this tender is to create „DSpace curation task“, that will allow automated item mapping from single or multiple source collections to single of multiple target collections and the appropriate reverse operation – cancelation of the item mapping within single or multiple collections (“unmapping”).

Table of Contents

Requested functionality

Variants of the item mapping across DSpace collections

Curation task allows automated mapping of items from multiple source collection to multiple target collections (M:N relation), of which the following are the special cases of such automated mapping of items:

- automated mapping from single source collection to single target collection,
- automated mapping from single source collection to multiple target collections,
- automated mapping from multiple source collections to single target collection.

Aforementioned variants of automated item mapping across DSpace collections are to be available for two modes of the curation task:

- “with a mapping file”,
- “without a mapping file”.

Curation task shall allow a reverse operations of both modes as well:

- “reversed operation with a mapping file”,
- “reversed operation without a mapping file”.

All modes are further explained and described later in this document. Following conditions have to apply for items at all times for these items to be considered eligible for a given curation task’s operation mode[1]:

- “mapping mode”:
  - item scheduled for mapping must not be mapped in the target collection(s) already,
- “unmapping mode”:
  - item scheduled for “unmapping” must be mapped into at least one of specified target collections.

Curation task must check if both the source collection(s) and target collection(s) exist in DSpace (if specified as an input parameter or defined in mapping file). Curation task shall be executed in two independent ways, by:

- automated process immediately after the item is deposited & archived in DSpace; or
- manual execution of the curation task command in the terminal / command line.

DSpace Administrator (role given to a single or multiple DSpace E-Person) shall be informed about both successful and unsuccessful curation task execution through the curation task’s output in terminal and via the appropriate log file specifically crated for this curation task.

[1] Even more specific conditions might apply; based on the curation task’s operation mode and its use of a mapping file.

Automated process running the curation task upon item's deposition / archival in DSpace

Curation task shall be executed automatically after the item is deposited & archived in DSpace. This usage assumes a default operation mode of the curation task to be 'with the mapping file' and requires an independent configuration / mapping file. With the possibility of complex mapping rules in mind, we require the format of the mapping file to be a JSON or XML. JSON mapping file format is preferred.

Manual execution of the curation task from terminal / command line

Curation task should allow to be executed / started by DSpace Administrator from terminal / command line on top of / outside the default "automated" execution described before. Manual execution of the task shall allow DSpace Administrator to run the curation task in the following modes:

- with a mapping file,
- without a mapping file,
- reversed mode with a mapping file,
- reversed mode without a mapping file.

Curation task's behavior in all aforementioned operation modes, its parameters and description of its functionality are explained and described in the following sections of this document.

Common input parameters of the curation task

Common input parameters of the curation task are used in case of task being executed from terminal / command line. In case automated task execution after a successful deposition & archival of the item in repository, task will be executed in a “with mapping file” operation mode. When executing task from terminal / command line, DSpace Administrator has to set the operation mode using the –operation (-o) parameter, as described in the table below.

parameter name	operators (long variant)	operators - short variant	mandatory (TRUE/FALSE)	parameter description
operation mode	--operation	-o	TRUE	Valid value of this parameter is one of the following terms: unmapped, mapped, reversed, reversed-mapped. When term „unmapped“ is used, mapping will be executed in the „without mapping file“ operation mode. When term „mapped“ is used, mapping will be executed in the „with mapping file“ operation mode. When term „reversed“ is used, mapping reversal / cancelation („unmapping) will be executed in the „without mapping file“ operation mode. When term „reversed-mapped“ is used, mapping reversal / cancelation („unmapping“) will be executed in the „with mapping file“ operation mode. All these mapping and „unmapping“ operation modes are described below; in appropriate sections of this document.

“Without mapping file” operation mode

In this operation mode, curation task executes mapping of all DSpace items in a source collection(s) to a destination collection(s). When --source parameter is omitted, curation task will perform mapping of all items from whole repository to a set destination collection(s). Rules described in “Variants of the item mapping across DSpace collections” section of this document apply.

parameter name	**operators (long variant)**	**operators - short variant**	**mandatory (TRUE/FALSE)**	**parameter description**
source collection / collections	--source	-s	FALSE	Value of this parameter will be a HANDLE ID of the source collection, from which the items will be mapped. When items from multiple collections have to be mapped, curation task will accept either multiple occurrences of --source (-s) parameter (each with its own HANDLE ID value); or space-separated list of collection HANDLE IDs. When this parameter is omitted, curation task will perform item mapping of all items from the repository to set destination collection(s).
destination (target) collection / collections	--destination	-d	TRUE	Value of this parameter will be a HANDLE ID of the destination collection, into which the items from source collection(s) or whole repository will be mapped. When items should be mapped to multiple destination collections, curation task will accept either multiple occurrences of the --destination (-d) parameter (each with its own HANDLE ID value); or space-separated list of collection HANDLE IDs.

“With mapping file” operation mode

This operation mode will be used primarily for automated mapping of items after their successful deposition & archival in the DSpace repository. However, this operation mode will also be used on ad hoc mapping of items initiated by DSpace Administrator and therefore the curation task has to support it when executed from terminal / command line as well. This operation mode will be initiated by using term “mapped” as a value of --operation (-o) parameter when executing curation task from terminal / command line.

Decision which item from which source collection(s) should be mapped into which target collection(s) will be made according to the information that will be provided to curation task by the so called “mapping file”. This mapping file will contain following information:

- identifier of the mapping file itself (we expect it to be a human-readable string - “title”),
- name / identification of the primary metadata field; based on the values of this metadata field items will be mapped to a corresponding DSpace collection,
- name / identification of the secondary metadata field; this metadata field will server as a fallback field in cases, when item scheduled for mapping does not have (or could not have) a primary metadata field at all or if the value of primary metadata field is empty (null),
- information about the source collections, from which the items will be mapped; this information consists of:
    - HANDLE ID of the source collection,
    - human-readable name of the collection in Czech language,
    - human-readable name of the collection in English language,
- the mapping part itself, which will consist of the following information:
    - value of the primary or secondary metadata field,
    - appropriate HANDLE ID of the target collection(s) that corresponds to value of the primary or secondary metadata field

Curation task will allow reading / parsing of the mapping file from:

- local storage,
- URL address.

The location of the mapping file will be specified in an appropriate parameter when curation task is executed from terminal / command line. When the automated curation task execution is set-up for successfully deposited & archived items, local storage of the mapping file will be considered as a default. However, when needed, this default setting of mapping file’s location could be overridden / configured in a separate curation task’s configuration file. The configuration file for currently existing Virus Scan curation task[2] can be used as an example of such separate curation task’s configuration file.

Primary metadata field is the metadata field of ‘item’ type DSpace object, it will be used as a first option when the curation task will be trying to find appropriate target collection(s) HANDLE ID in associated with a defined metadata value. Only when item does not have a primary metadata field at all, or its value is empty (null), curation task should try to find a secondary metadata field in the item and use its value to find an appropriate collection(s) HANDLE ID in the mapping file. If the processed

[2] https://github.com/DSpace/DSpace/blob/a610d25114f109fd41209640f0f3b9e40abd778c/dspace/config/modules/clamav.cfg

item does not contain primary metadata field, nor secondary metadata field, or their values are empty (null), the curation task will move to processing next item.

Primary and secondary metadata fields will be defined in the mapping file as a strings, following the convention specified in DSpace (6.x) Documentation[3], e. g. “schema.identifier” or “schema.identifier.qualifier”. This operation mode of the curation task will be using parameters defined and described in the following table when executed from terminal / command line.

[3] https://wiki.lyrasis.org/display/DSDOC6x/Metadata+and+Bitstream+Format+Registries

parameter name	operators (long variant)	operators - short variant	mandatory (TRUE/FALSE)	parameter description
path to mapping file – network location (URL address)	--link	-l	FALSE	Value of this parameter is a path to mapping file in a network location – URL address leading to the mapping file. URL address has to be accessible and valid. Of this parameter is omitted, curation task will use location type and its path defined in curation task's configuration file as fallback value.
path to mapping file – local (system) storage	--localpath	-p	FALSE	Value of this parameter is a path to mapping file in local (system) storage. If this parameter is omitted, curation task will use location type and its path defined in curation task's configuration file as a fallback value.
source collection / collections	--source	-s	FALSE	Value of this parameter is a HANDLE ID of the collection, from which the items should be mapped according to mapping rules defined in the mapping file. If item mapping should be executed on items from multiple source collections, curation task should accept either multiple occurrences of the --source (-s) parameter (each with its own HANDLE ID value) or space-separated list of collection HANDLE IDs. If this parameter is omitted and mapping file does not contain definition of source collection(s), all items in repository will be mapped according to rules defined in the mapping file. If this parameter is omitted and mapping file contains definition of source collection(s), only items in source collection(s) defined in mapping file should be mapped according to rules defined in this mapping file. Using this parameter has a priority over information on source collections provided in the mapping file.

Reversed operation mode without mapping file

This operation mode will allow cancellation of mapping ("unmapping") of the DSpace "item" type objects either in the target / destination collection(s) identified by HANDLE ID(s) in the --destination (-d) parameter of in the whole repository when the --destination (-d) parameter is omitted. This operation mode should provide a necessary means how to fix possible errors in item mapping for multiple items in one or multiple collections at once; without the need to interact with DSpace User Interface. Curation task in this operation mode could only be executed from terminal / command line and it will accept parameters defined and described in the following table.

parameter name	operators (long variant)	operators - short variant	mandatory (TRUE/FALSE)	parameter description
destination (target) collection / collections	--destination	-d	FALSE	Curation task will accept either: single DSpace collection HANDLE ID (when items from single collection should be „unmapped“), or it will accept either multiple occurrences of the --destination (-d) parameter (each with its own collection HANDLE ID) or space-separated list of multiple collection HANDLE IDs (when items from multiple collections should be „unmapped“). If this parameter is omitted, all mapped items across all collections in the whole repository will be unmapped.

Reversed operation mode with mapping file

This operation mode will allow item mapping cancelation ("unmapping") from collections associated with a value of primary or secondary metadata field defined in the mapping file. For successful "unmapping" process, item needs to fulfill all of the conditions listed below:

- item has to have a primary metadata field with a non-empty value OR,
- item has to have secondary metadata field with a non-empty value,
- value of either primary or secondary metadata field is found in the mapping file,
- item is mapped in at least one of the target / destination collection(s) associated with a given primary or secondary metadata field value.

When the aforementioned conditions are met, item is unmapped from the target / destination collection(s) identified by HANDLE ID of such collection(s). Main goal of this operation mode is to return item mapping to its previous state; before the items were mapped using the curation task's "with mapping file" mode of operation.

Curation task will allow execution in this mode only when its execution is initiated from terminal / command line and will accept parameters defined and described in the table below.

parameter name	operators (long variant)	operators - short variant	mandatory (TRUE/FALSE)	parameter description
path to mapping file – network location (URL address)	--link	-l	FALSE	Value of this parameter is a path to mapping file in a network location – URL address leading to the mapping file. URL address has to be accessible and valid. Of this parameter is omitted, curation task will use location type and its path defined in curation task's configuration file as fallback value.
path to mapping file – local (system) storage	--localpath	-p	FALSE	Value of this parameter is a path to mapping file in local (system) storage. If this parameter is omitted, curation task will use location type and its path defined in curation task's configuration file as a fallback value.
source collection / collections	--source	-s	FALSE	Value of this parameter is a HANDLE ID of the collection, from which the items should be checked and „unmapped“ from target / destination collection associated with a value of their primary or secondary metadata field. If item "unmapping" should be executed on items from multiple source collections, curation task should accept either multiple occurrences of the --source (-s) parameter (each with its own HANDLE ID value) or space-separated list of collection HANDLE IDs. If this parameter is omitted and mapping file does not contain definition of source collection(s), all items in repository will be checked and „unmapped“ from target / destination collections associated with the value of

				their primary or secondary metadata field. If this parameter is omitted and mapping file contains definition of source collection(s), only items in source collection(s) defined in mapping file should be checked and „unmapped“ from collections associated with the value of their primary or secondary metadata field. Using this parameter has a priority over information on source collections provided in the mapping file.

Example of curation task’s configuration file for automated item mapping

Configuration file of this curation task should have at least the following configuration properties:

- location of the mapping file
  - value of this property will be used for getting information about the location of the mapping file when curation task will be executed on items successfully deposited & archived in the repository
  - value of this property will be used as a fallback value when the location is not provided via the appropriate parameter when curation task is executed / initialized from terminal / command line
- type of mapping file location; available types are following:
  - local: mapping file is stored in local (system) storage
  - url: mapping file is stored on a network location

For each type of mapping file location (local, url), curation task has to confirm, that the defined path is valid and accessible. Example of this configuration file (see file curation_task_config_en.cfg) is part of the **Annex 1: Configuration examples** of this specification.

Example of curation task’s mapping file

Since the mapping file has to be machine-readable and should also maintain some level of human-readability, the expected format of the mapping file is JSON or XML. Preferred format of the mapping file is JSON. Apart from the defined key -> value pairs / elements with information about the actual mapping of a value of primary or secondary metadata field to target / destination collection(s), the mapping file should also contain the following additional information:

- information on the source collection(s) from which the items will be checked and subsequently mapped / unmapped based on the value of the primary or secondary metadata field,
    - HANDLE ID,
    - collection title in Czech,
    - collection title in English,
- information on the primary and secondary metadata field, from which the values will be used for mapping / “unmapping” of items,
- information on the target / destination collection(s) associated with a given value of item’s primary or secondary metadata field,
    - HANDLE ID,
    - collection title in Czech,
    - collection title in English,
    - additional collection identifiers or description of a given collection(s)

Existence of the additional information (e. g. collection title, description additional identifier apart from HANDLE ID) on source / destination collection(s) is not a key factor for curation task’s functionality; HANDLE ID is a key factor for its functionality. Examples of the curation task mapping files are provided in **Annex 1: Configuration examples** of this specification (see files json_mapfile_example.json and xml_mapfile_example.xml).

Usecases

"Without mapping file" operation mode

This operation mode tries to mitigate the administrative burden of mapping items in current DSpace. As of now, item can only be mapped individually through DSpace User Interface.

The main usecase for this operation mode of this curation tasks is the need to map large quantities of documents already archived in our DSpace repository, e. g. digitized study material for students with special needs. These digitized documents are currently submitted and archived in two separate collections. The reasoning for these separate collections is:

- repository is structured to communities based on the organization structure (each community = a faculty or independent part of the university)
- communities structured in such a way allow each faculty / independent university part to:
    - have a clearer picture about its works and related statistical information,
    - promote just its own part of the university repository and link to it,
    - collections (created based on the "type" of content (e. g. theses, OA articles, OA books, etc.) allow greater granularity of user authorization, since creation and / or administration of each "type" of content is heavily decentralized and spread across different departments of the faculty or independent part of the university)

Recently however, cooperation on creation of more specific and unified / theme oriented collections is taking place among various faculties and / or independent parts of university, e. g. creation of a unified collection for digitized study materials for students with special needs. This collection should provide such repository users with a centralized place, where they can find all relevant documents. However, to allow a (needed) decentralized collection administration and at the same time centralize the submitted content without multiplication of items in the repository, items from various collections have to be mapped to this one centralized collection. To summarize, the main reasons for existence of this curation task and this curation task's operation mode are these:

- to allow mapping items from multiple (individually administrated) collections of digitized documents for students with special needs to one, centralized special collection,
- to allow easy creation of temporary thematically oriented collections of items from items already archived in the repository,
- to allow easy change of the repository structure.

"With mapping file" operation mode

This operation mode is tight to our current needs in regards of setting up institutional repository for scientific publications. For this repository, we plan a similar, but much more deeper structure than in the CU Digital Repository. Scientific outputs (articles, books, etc.) will be automatically submitted to the DSpace installation via the SWORDv2 Server from CU CRIS system.

Currently, the scientific outputs are registered in the CRIS system to a department of the primary author (so called department with the primary responsibility for the research output) and to (possibly multiple) departments in cases when the research output is created in cooperation with co-authors from different departments within the same (or different) faculty (so called department(s) with the secondary responsibility for the research output).

Research output will be submitted to DSpace only once; to the collection representing the department with the primary responsibility for the research output. However, to be able to link the research output to department(s) with secondary responsibility and to do that without multiplication of items within the repository, our repository needs an efficient way how to map large amounts of submitted & archived items to different collections; without the need for extensive manual labor.

Submission to a collection representing the department with primary responsibility for a given research output will be done based on the identifier of this department provided from and by our CRIS system; the submission itself will be carried out by the automated process within the CRIS system itself. However, CRIS system won't be able to submit the same item to collection(s) representing departments with the secondary responsibility for the research output (to avoid multiplication of data). Instead, CRIS system will provide identifiers of these departments with secondary responsibility in defined metadata fields. Examples of such primary and secondary metadata fields could be:

- uk.secondaryDepartment.identifier (primary metadata field)
- uk.secondaryFaculty.identifier (secondary metadata field)

Thus, these values / identifiers in defined metadata fields can be used to automate the mapping of the items to collection(s) mapped to a given value of these defined metadata fields, as described above; avoiding extensive labor and possible multiplication of data in the repository.

Since the structure of the Charles University and processes connected to submission of data about research outputs to universities' CRIS system allow the research output to be connected to not only the department (lowest level in the CRIS systems organizational structure) but also to the faculty (higher level in the CRIS systems organizational structure), curation task has to take this into account. That is the reason why the primary and secondary metadata fields were introduced into the mapping file and the whole process of checking into which collection the item should be mapped.

The main ideas behind the primary and secondary metadata fields are the following:

- curation task in this mode will check if primary metadata field exists in items' metadata and if it has non-empty value; if so, it maps the item to the target / destination collection(s) according to mapping rules
- if the primary metadata field does not exist or if it has an empty (null) value, curation task in this mode will check, if secondary metadata field exists in items' metadata and if it has non-empty value; if so, it maps the item to the target / destination collection(s) according to mapping rules
- if neither primary, nor secondary metadata field is found in items' metadata; or if both have an empty (null) value, curation task should report that and continue processing next item
- if no mapping rule is found for a given value of primary or secondary metadata field, curation task should report that a continue processing next item

Since the expected amount of submitted research outputs reaches thousands a year, it is not feasible to do the mapping manually. This curation task also finds its uses in cases, when the repository structure has to be altered or completely changed, but according to a pre-defined rules which can be easily set-up in the mapping file. We consider the following (simplified) workflow for this operation mode & usecase:

1. researcher from Department of Ethnology of the Faculty of Arts submits information and fulltext of his research output to CRIS system
    a. Department of Ethnology is set as a department with primary responsibility for the research output in CRIS system
2. however, researcher was working in cooperation with his colleague from Department of Sociology of the Faculty of Social Sciences, researcher from Department of Sociology is set as a co-author of the submitted research output
    a. by doing so, Department of Sociology is set as a department with secondary responsibility for the research output in CRIS system
3. CRIS system automatically creates an import package for DSpace; as part of the metadata CRIS system sets two metadata fields to the following values:
    a. uk.primaryDepartment.identifier = 12345
    b. uk.secondaryDepartment.identifier = 67890
4. CRIS system submits the package to DSpace collection associated with ID of the department with primary responsibility for research output (ID = 12345); according to its own internal mapping, e. g. ID 12345 -> HANDLE ID 123456789/12
5. DSpace creates the item object from submitted package and archives it
6. DSpace runs automated mapping curation task in “with mapping file” mode on the archived item
    a. curation task, based on the configuration and mapping file, finds primary metadata field used for mapping item to another collection – field uk.secondaryDepartment.identifier and its value (67890)
    b. curation task compares this value (67890) with a mapping file and finds, that this value is associated with DSpace collection HANDLE ID = 123456789/33
    c. curation task maps item to DSpace collection with HANDLE ID = 123456789/33

If primary metadata field is not found, curation task checks the existence of the secondary metadata field and continues according to step 6b and if successful, continues to step 6c.

If primary metadata field, nor secondary metadata field is present or their values are empty, curation task reports a problem, does not map item anywhere and waits for another submission from CRIS system (step 4).

The same basic rules will be applied even when the curation task will be executed from terminal / command line. The only distinction is the intended use. Automated item mapping after submission and archival of item in DSpace is meant to be ran continuously. On the other hand, curation task executed from terminal / command line is meant to be used when a one-off mapping has to be done or when a set of rules different from the rules used for continuous item mapping has to be applied.

Reverse operation “without mapping file”

This operation mode is focused on the possibility to cancel the mapping of items in a whole repository or given collections in an automated manner. Being able to reverse automated procedure (e. g. automated mapping of items) by running another automated but reversed operation is considered a good practice in our point of view. This operation mode will find its uses in cases, when no complex rules were applied for mapping items from source to destination collection or in cases, when DSpace Administrator needs to quickly remove item mapping from single or multiple collection or whole repository (when appropriate parameter is omitted). Basic example of the workflow related to this operation mode can be described in a following way:

- DSpace Administrator executes curation task with --operation (-o) parameter set to value ‘reversed’ and sets it to run on a collection identifier by setting the --destination (-d) parameter to a value of collections’ HANDLE ID; for example “123456789/12”
- Curation task iterates over mapped items in a given collection and cancels the mapping

If there are more collections defined in the --destination (-d) parameter, curation task continues to process mapped items in next collection when finished with processing items from the previous collection (either successfully or unsuccessfully).

Reverse operation “with mapping file”

This operation mode will be primarily used when there is an error in creation of the mapping file for automated item mapping “with mapping file”, e. g. when values of primary or secondary metadata field will be mapped to wrong DSpace collection(s) ID(s). In that case it is vital to be able to reverse repository content to its original state. The nature of the automated mapping in “with mapping file” operation mode can lead to situation, when:

- it won’t be possible to discover, in sufficiently effective or fast way, to gather information which items were affected by the error in mapping file,
- it won’t be desirable to cancel mapping / “unmap” every object in the defined collection(s) with the use of reversed operation “without mapping file”.

We consider the following workflow for this reverse operation mode:

1. curation task iterates over object in defined source collections (if these are set in the mapping file itself or set via the appropriate parameter), if no source collection(s) is set, curation task iterates over items from whole repository,
2. curation task check the existence of primary metadata field
    a. if primary metadata field exists and is non-empty:
        i. compares value of this field with mapping file and gets HANDLE ID of collection(s), into which the item should be mapped
        ii. checks if item was mapped into the collection(s)
            1. if yes, cancels mapping / “unmaps” the item
            2. if no, continues to process next item
    b. if primary metadata field doesn’t exist or its value is empty (null):
        i. checks the existence of secondary metadata field
            1. if secondary metadata field exists and has a non-empty value:
                a. compares value of this field with mapping file and gets HANDLE ID of collection(s), into which the item should be mapped
                b. checks if item was mapped into the collection(s)
                    i. if yes, cancels mapping / “unmaps” the item
                    ii. if no, continues to process next item
            2. if secondary metadata field doesn’t exist or its value is empty (null):
                a. continues to process next item

Examples of calling the curation task from terminal / command line

Set of examples of execution of the curation task from terminal / command line is part of the **Annex 2: Example calls** of this specification (see file curation_task_terminal_call_examples_en.pdf).

List of Annexes and their identification

Annex name	**Filename**
Annex 1a: Configuration examples	Annex_1a_Configuration_examples.zip
Annex 1b: Example calls	Annex_1b_Example_calls_en.pdf