Rosetta 5.2 Highlights

March 2017

ExLibris The bridge to knowledge

CONFIDENTIAL INFORMATION

DISCLAIMER

TRADEMARKS

Document released: March 2017

Author: Adi Alter, Rosetta Product Manager

Web address: http://www.exlibrisgroup.com

Table of Contents

Contents

1

Infrastructure

1.1 SAML/Local Authentication

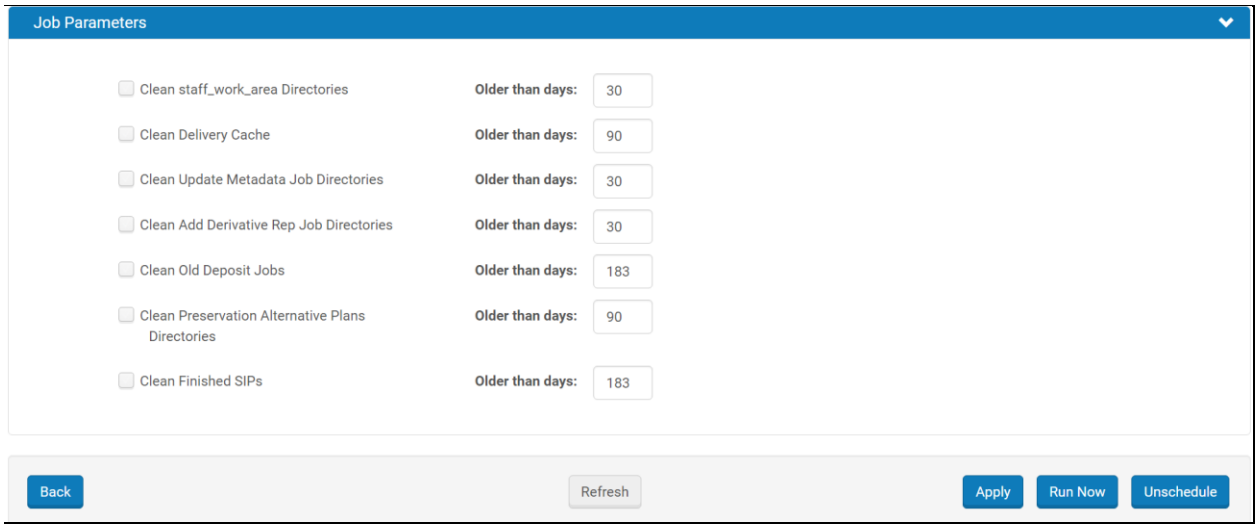
Rosetta now supports authenticating users without the need to use a PDS server. Users can either be authenticated locally by Rosetta or can be authenticated by an external system using SAML (Security Assertion Markup Language) 2.0, an XML-based, open-standard data format for exchanging authentication and authorization data between parties.

SAML enables Rosetta to exchange authentication and authorization information with your institutional identity provider (IdP), allowing a single sign on for the institution's users. A user who will sign in or out of an external system will be automatically signed in or out of Rosetta, or vice versa.

1.2 Cleanup Job

A new cleanup job has been added to Rosetta's system jobs, deleting data older than a configurable number of days. The following areas are covered by this new job:

- Staff work area Directories
- Delivery Cache
- Update Metadata Job Directories
- Add Derivative Representation Job Directories
- Old Deposit Jobs
- Preservation Alternative Plans Directories



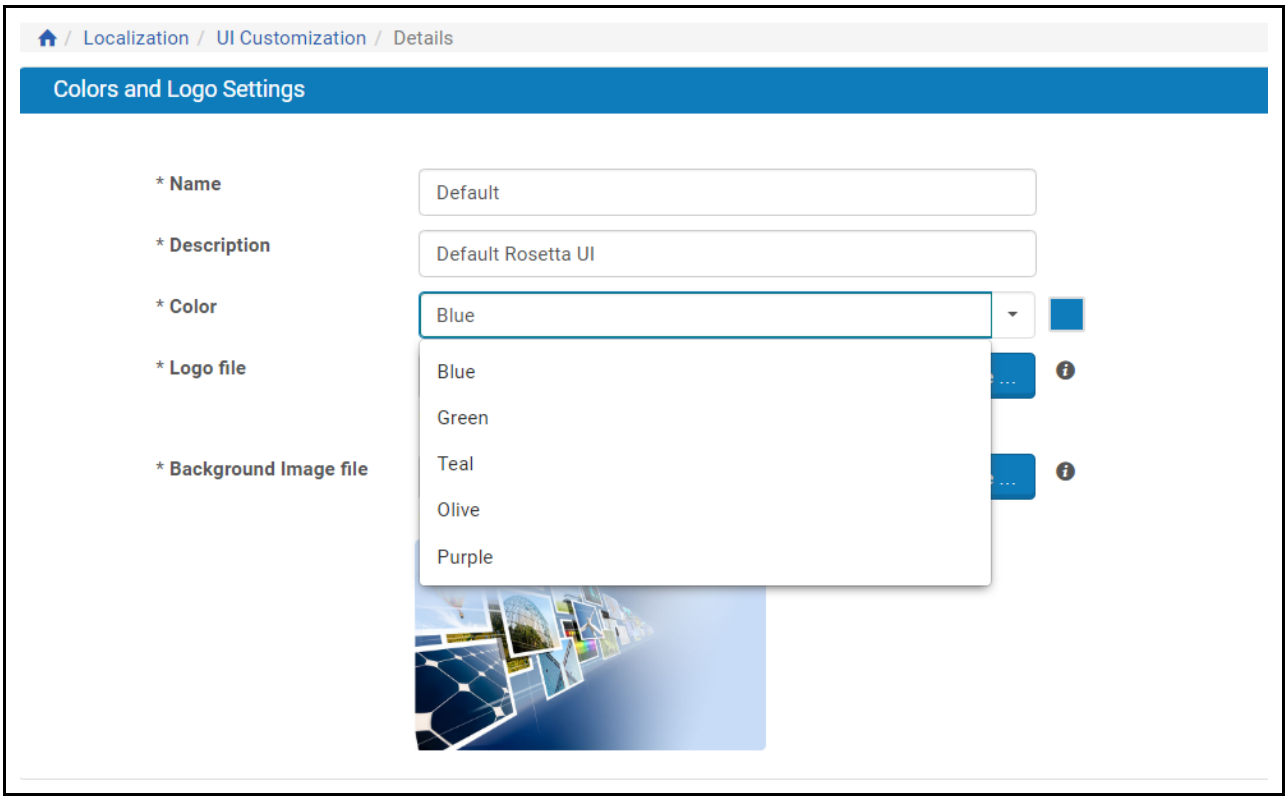
Cleanup Jobs Configuration

2

User Interface

We continued the work started in version 5.1 by further enhancing Rosetta's UI and introducing multiple improvements throughout the system.

As part of this work, Rosetta now supports once more UI color customization. It is possible to choose a color scheme from the following five colors: purple, blue, teal, green and olive. For more information about these colors, their meaning, and their variance, refer to the Colors section available on [ProQuest's UX Framework site](#).



UI Customization

3

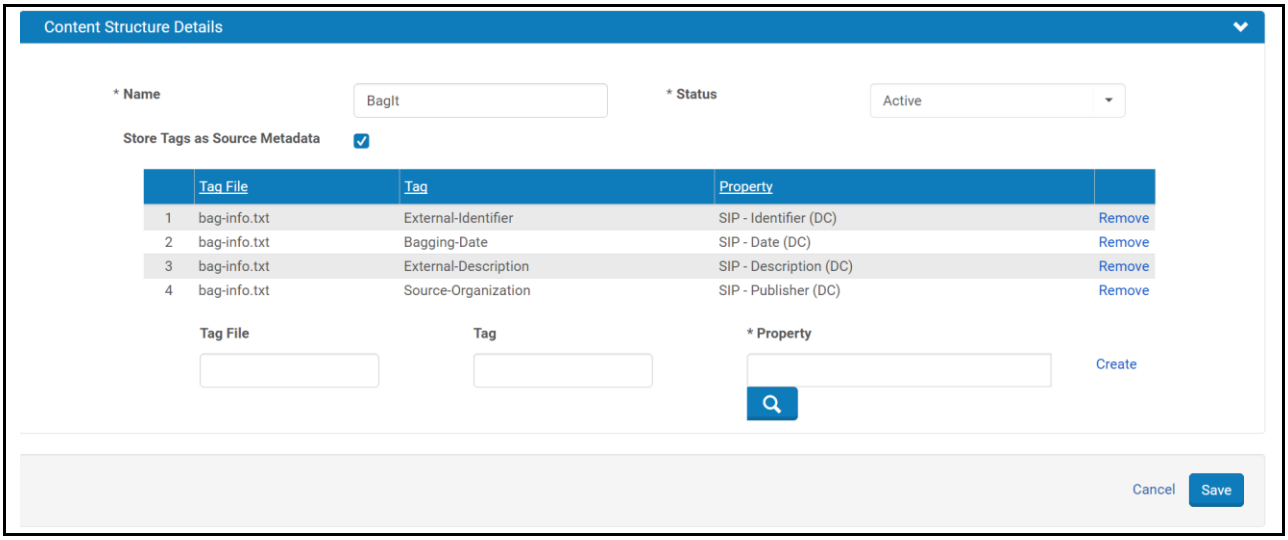
Deposit and SIP Processing

3.1 BagIt Deposit

It is now possible to upload SIPs into Rosetta using BagIt, a hierarchical file packaging format designed to support disk-based storage and network transfer of arbitrary digital content.

BagIt bags are ideal for digital content normally kept as a collection of files. They are also well-suited to the export, for archival purposes, of content normally kept in database structures that receiving parties are unlikely to support. BagIt is easy to implement and is cross-platform, therefore making it widely adopted by many digital libraries.

The new BagIt Content Structure Converter enables uploading content in the BagIt format and mapping BagIt tags to Rosetta's metadata. Rosetta extracts the bag, deposits the files composing it into Rosetta, and creates an intellectual entity out of them.

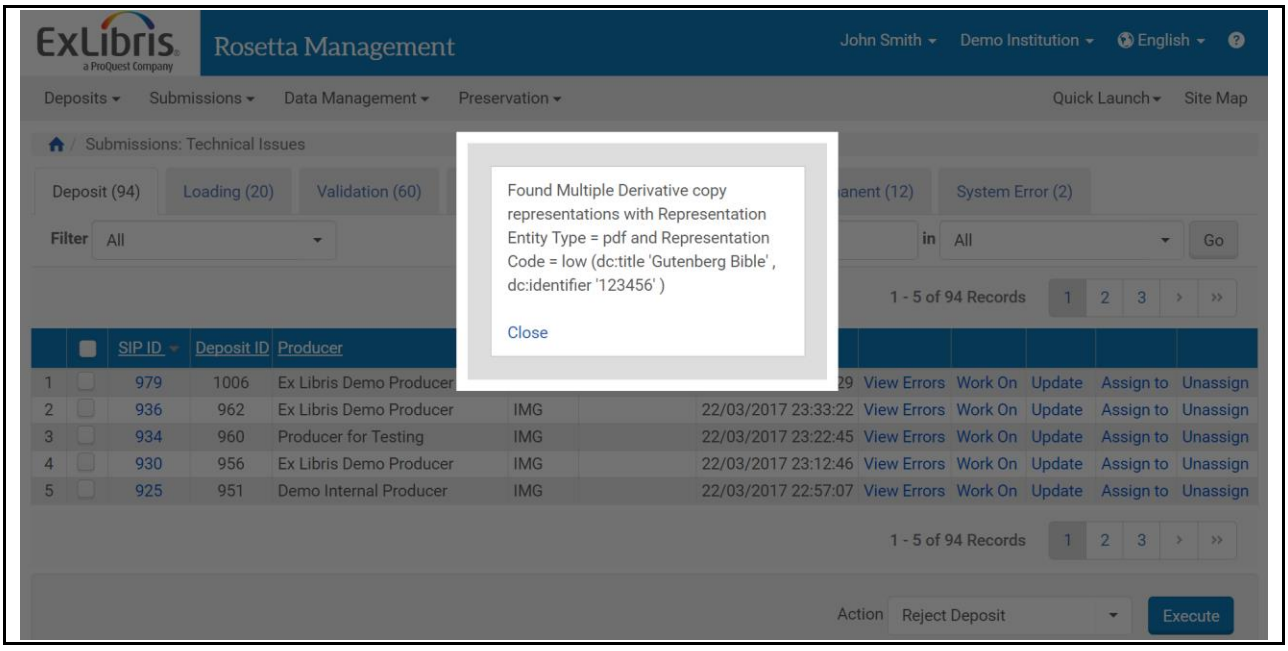


BagIt Content Structure Converter

3.2 Improved TA Workbench and TA Error Visibility

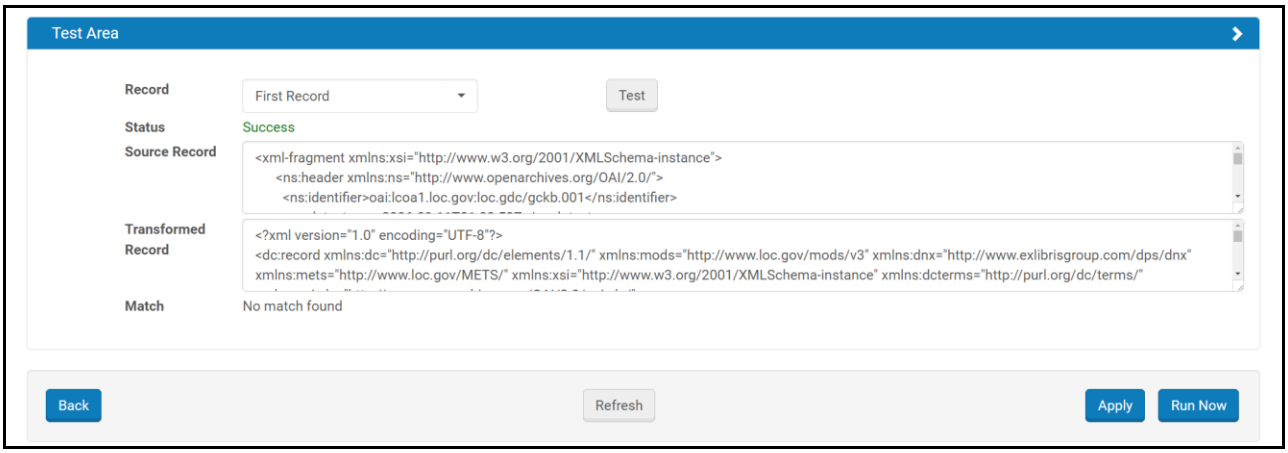
The columns of the different tabs on the Technical Issues page have been standardized, ensuring consistent user experience across tabs.

In addition, a new **View Errors** action has been added to the relevant tabs. This action displays the error details of the relevant event, containing the failure reason. As part of this effort, the CSV Content Structure error logging was enhanced to include the problematic IE/REP information, thus enabling easier troubleshooting of the submission error.



3.3 OAI-PMH Test UI

The OAI-PMH Harvester allows users to load objects into Rosetta directly from an OAI-PMH digital repository. It is now possible to test the OAI-PMH Harvester job and check its connectability, record transformation, and match per selected job configuration, thus ensuring successful execution of the job once it is scheduled to run a full harvest.



OAI-PMH Test Area

3.4 Submission Job Reports

The following two new reports were added to the Submission Reports section, enabling better tracking of submission jobs' status:

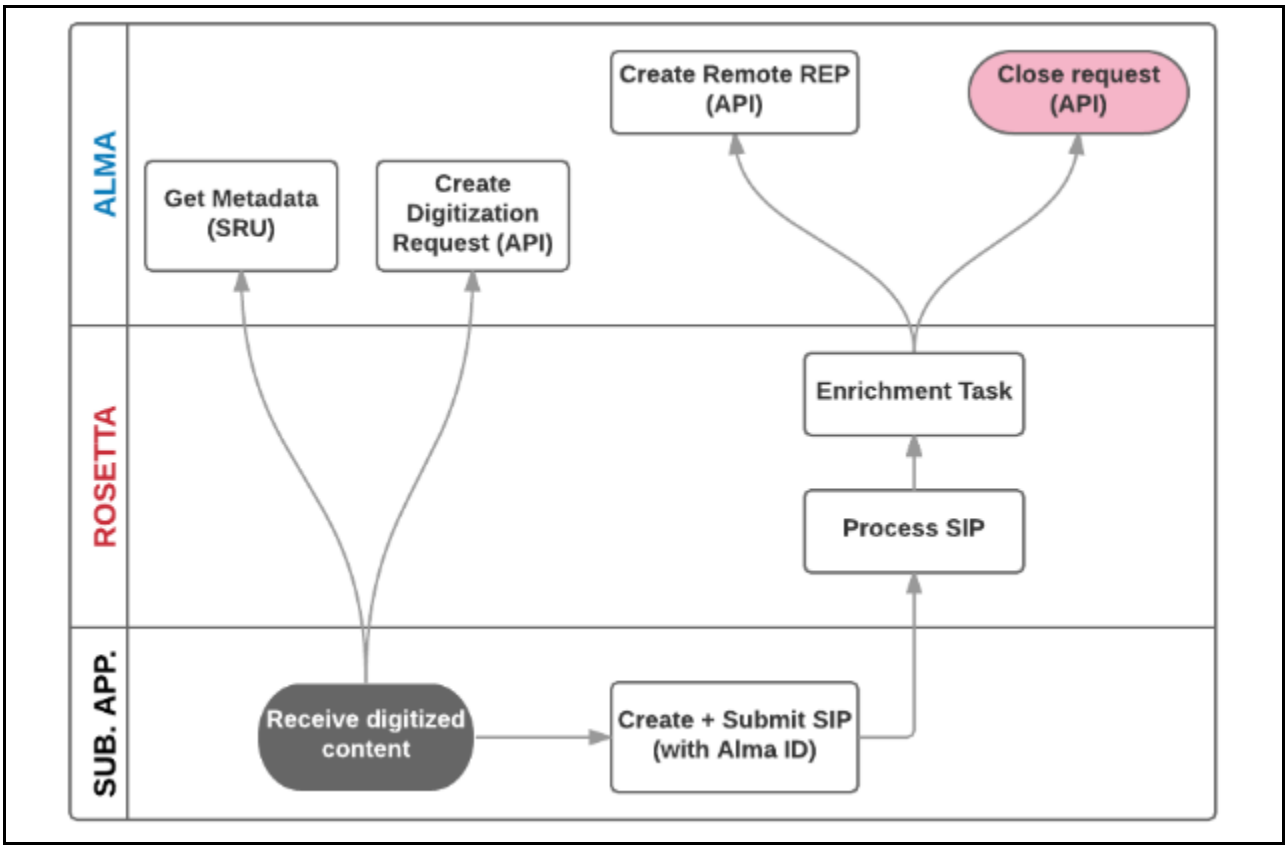
- Submission Job Details by Job Name and Execution Date
- Submission Job Details by Source Directory

4

Data Management

4.1 Alma-Rosetta Digitization Flows

The new Alma-Rosetta integrated digitization flow provides a submission application that creates Rosetta SIPs for digitized content and a Rosetta repository task that uses Alma APIs to create remote digital inventory for the ingested content, as can be seen in the following diagram:



4.1.1 Alma Submission Application

The Alma Submission Application is used for submitting to Rosetta digitized content related to an Alma bibliographic record (and, optionally, a fulfilment of a related digitization request). It retrieves descriptive metadata from Alma using SRU, creates a Rosetta METS file, and submits Rosetta SIPs for a set of files.

The application is based on the Rosetta SDK and Alma bibliographic APIs. The application code is available on GitHub and it can used as is, or modified to fit specific needs and workflows.

4.1.2 'Create Alma Digital Inventory' Repository Task

This new repository task creates digital inventory in Alma using information coming from IEs created by the Alma submission application.

A successful **Create Alma Inventory** task results in the following:

- A new remote representation in Alma.
- If provided, the Alma request ID is added as a representation property, removed from the IE, and then the request is closed (if it is in the appropriate stage).

4.1.3 'Delete Alma Digital Inventory' Repository Task

The delete Alma inventory task can be manually run to delete digital inventory created incorrectly or to remove digital inventory before deleting the IE.

4.2 Delete Derivative Copy Representation Task

Derivative copy representations can now be deleted using a Rosetta job. This is especially helpful when you need to delete such copies in bulk.

5

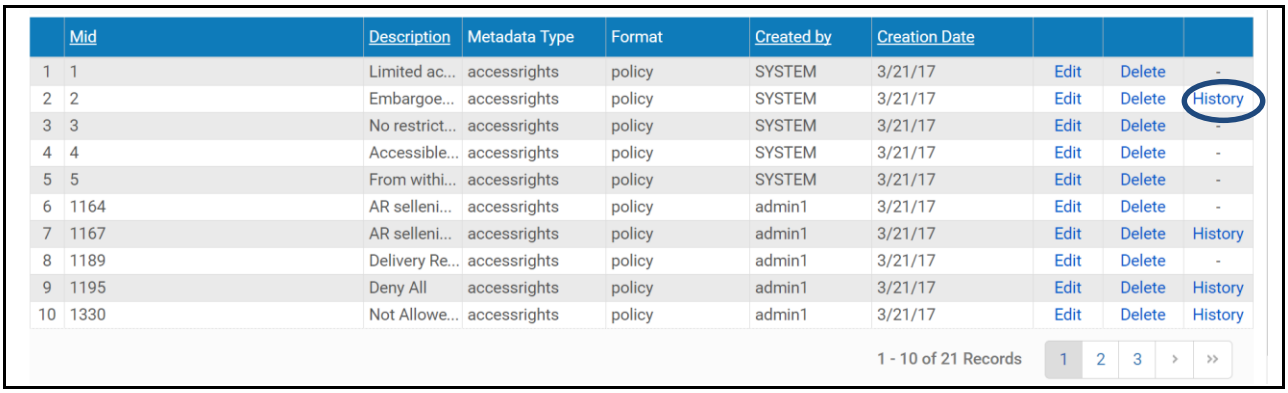
Preservation

5.1. Retaining Timestamps for Digital Content Uploaded to Rosetta

Original file timestamps are now preserved for storage that uses the copy method for all kinds of submission formats, thus ensuring that original date characteristics are not modified when moving content into Rosetta.

5.2. Tracking Changes to Access Rights Policies

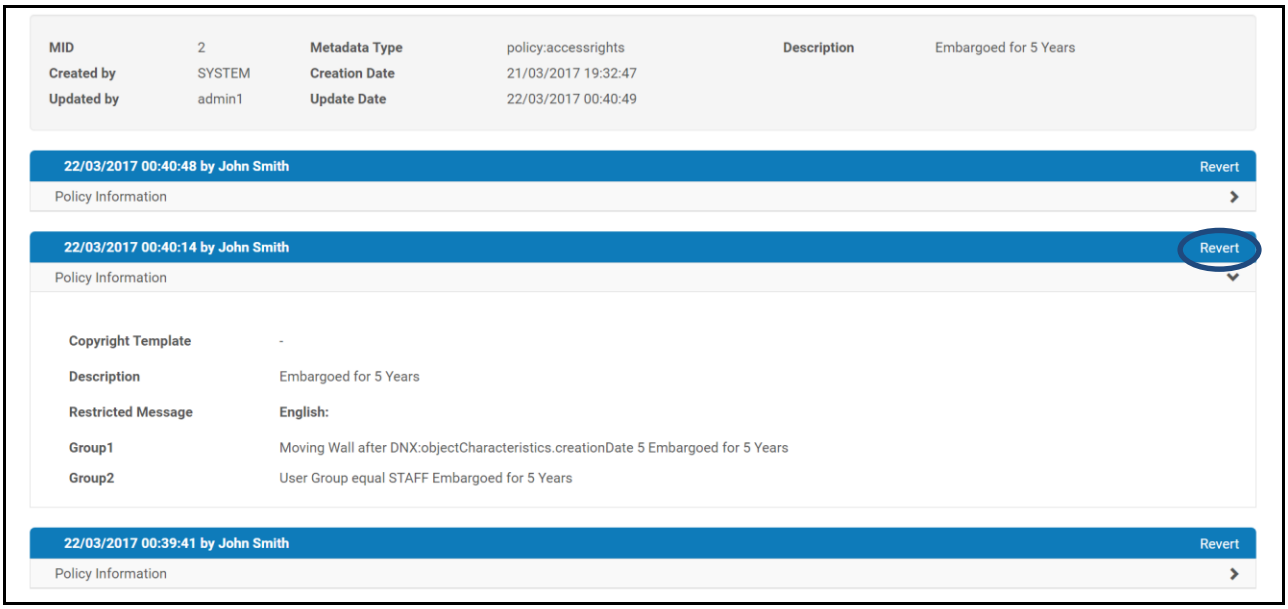
Access Rights, Access Rights Exceptions, and Retention Policies now have a **History** action, enabling users to view and revert to previous MD versions.



	Mid	Description	Metadata Type	Format	Created by	Creation Date			
1	1	Limited ac...	accessrights	policy	SYSTEM	3/21/17	Edit	Delete	-
2	2	Embargoe...	accessrights	policy	SYSTEM	3/21/17	Edit	Delete	History
3	3	No restrict...	accessrights	policy	SYSTEM	3/21/17	Edit	Delete	-
4	4	Accessible...	accessrights	policy	SYSTEM	3/21/17	Edit	Delete	-
5	5	From withi...	accessrights	policy	SYSTEM	3/21/17	Edit	Delete	-
6	1164	AR selleni...	accessrights	policy	admin1	3/21/17	Edit	Delete	-
7	1167	AR selleni...	accessrights	policy	admin1	3/21/17	Edit	Delete	History
8	1189	Delivery Re...	accessrights	policy	admin1	3/21/17	Edit	Delete	-
9	1195	Deny All	accessrights	policy	admin1	3/21/17	Edit	Delete	History
10	1330	Not Allowe...	accessrights	policy	admin1	3/21/17	Edit	Delete	History

1 - 10 of 21 Records 1 2 3 › »

Access Rights Policies Management



Access Rights Policies History Management

5.3. Storage Migration

Rosetta supports now migrating a set of files from one permanent storage location to another. The migration is based on recommitting files to the permanent storage. A fixity check confirms that the source and target files are identical. This is helpful, for example, when there is a need to move some of the objects stored on one storage location to another, while still leaving other objects on the original storage.

6

Publishing and Delivery

6.1 Search and SRU Improvements

6.1.1 Exact Search for Identifiers

Exact search is now supported for identifiers both via the UI using the `equals` operator and via SRU using the == operator.

6.1.2 "Is Empty" Support for SRU

The `Is Empty` operator is now supported in SRU, by using the ==”” operator.

6.2 Fulltext Viewer Preprocessor (VPP)

A new Viewer Preprocessor (VPP) that converts all PDF files in a representation into a single HTML file was added to Rosetta. This VPP should be used with the StreamGate viewer to provide a fulltext representation to a harvester (for example, Primo).

7

APIs

7.1 Access Rights Check Service

Rosetta now offers a service to check permissions on an object without actually delivering it. This is done by accessing the delivery URL using a HEAD HTTP request. Rosetta will return a 401 or 403 HTTP status code if access is denied and it may also return a denied note. Such a request does not count as a delivery request.

Note that this service is now used by Alma to indicate in the service tab (ViewIt) to a patron that access to a remote representation is denied by the remote repository.

7.2 APIs Updates

- The getRipStatus web service was enhanced to return a **Finished** Status for successfully added/updated representations.
- Required roles for executing web services have been updated thus providing improved web services security.
- Web services Javadoc documentation has been improved