Rosetta Administration Guide

- **The Rosetta System Architecture**
  No image available
  - Rosetta System
  - Rosetta Application Roles
  - Administering the Solr Server
  - Rosetta-Supported Topologies
  - Load Balancer Requirements
  - Application Roles
  - Managing Plug-Ins
  - Disk Space Management
  - Rosetta Storage Management
- **General Parameters**
  No image available
- **Updating the Rosetta System**
  No image available
- **Working with the Rosetta System**
  No image available
- **Working with the Rosetta Database**
  No image available
- **Backing Up and Recovering Data**
  No image available

The Rosetta System Architecture

This page was not added to the PDF due to the following tag(s): article:topic-guide

Rosetta System

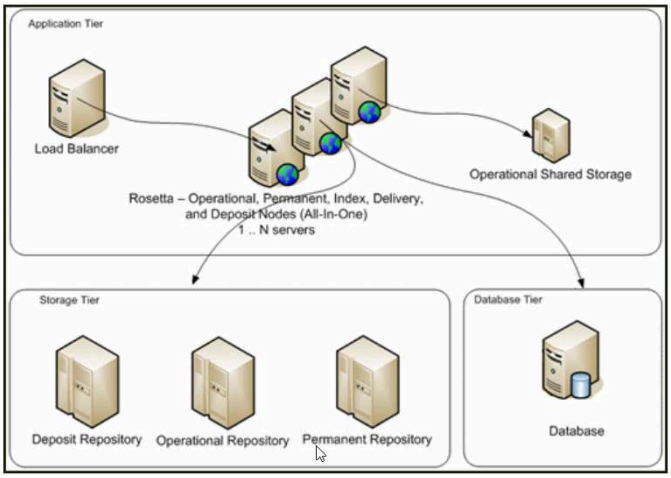
About the Rosetta System Architecture

The Rosetta system architecture is based on the multi-layer concept. Components of each layer can interact with components of other layers, as well as with components of the same layer.
The table below describes the layers and their components:

The Rosetta System Layers

Layer	Components	Description
Storage	• Deposit Storage • Operational Storage • Permanent Storage	Provides physical storage of Producer Agent content for all files that are processed and preserved.
Application roles	• Deposit • Repository • Delivery • Index (SOLR) • Permanent	Executes all Rosetta processes and activities - SIP processing, Maintenance tasks, Delivery, Preservation Actions.
Database	Schemas: • DEP • SHR • REP • RPT • PER	Stores data, configuration items, and is used for operating Rosetta.

The figure below illustrates the components that each layer contains, and provides a general overview of the interaction between these layers and their components:



Rosetta System Architecture

The different application roles communicate each other by calling web services (WS) or by using queues managed in the shared database scheme.

The following sections contain detailed descriptions of the processes and components in each stage.

Rosetta Application Roles

Rosetta Application Roles

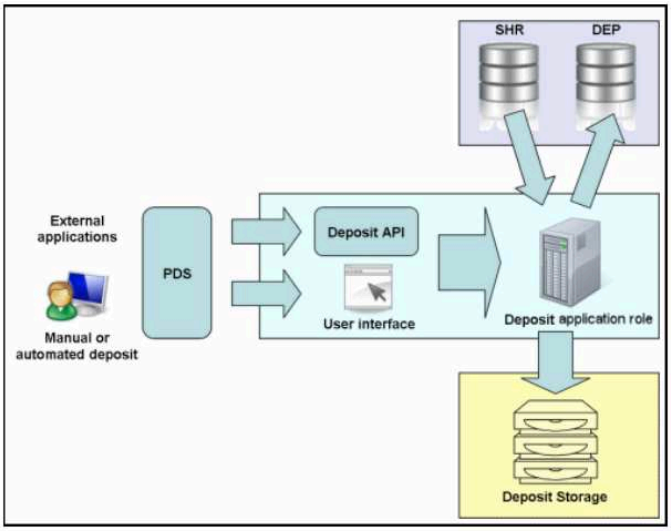
Each server used by Rosetta can be assigned to one or more application roles. Each application role is in charge of different activities. For example, all Deposit activities—acquiring the deposited files, converting the input data to METS structure—are performed by servers whose APP_ROLE = DEP (Application Role is set to Deposit). If more than one server is assigned to the same application role, a load balancer is required to balance the load between the servers. This section describes the following Rosetta application roles:

- Deposit Role
- Repository Application Role
- Permanent Application Role Server

Deposit Role

The Deposit role acquires the deposited material and stores this content in the Deposit Storage.
The Deposit role interacts with the following components, as illustrated in the figure below:

- Patron Directory Service (PDS)
- Database Schemas
- Deposit Storage



Patron Directory Service (PDS)

Users can register using either an external legacy application or the Rosetta system. The PDS enables the authentication and login of users regardless of the registration method. For example, the PDS can be configured to work with an external user database, such as an LDAP directory service.
The PDS is configured by a System Administrator.

Database Schemas

The Deposit application role uses the following database schemas:

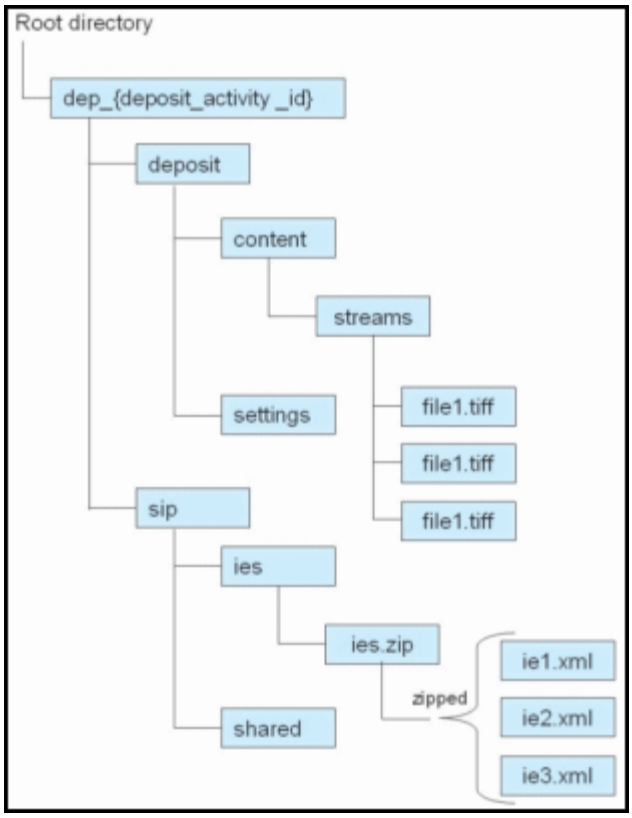
- SHR, which contains configuration information required for processing Producer Agent content (such as material flow and Producer profile configuration).
- DEP, which stores information about deposit activities that a Producer Agent submitted

Deposit Storage

The Deposit storage area contains

- deposit activities, which are deposited by Producer Agents, and
- submission information packages (SIPs), which are generated by the Rosetta system.

For each deposit activity, the Rosetta system creates a Deposit folder. To differentiate between Deposit directories that store different deposit activities, the Rosetta system adds an automatically generated ID to the name of each Deposit directory.
The figure below illustrates the organization of a Deposit directory.

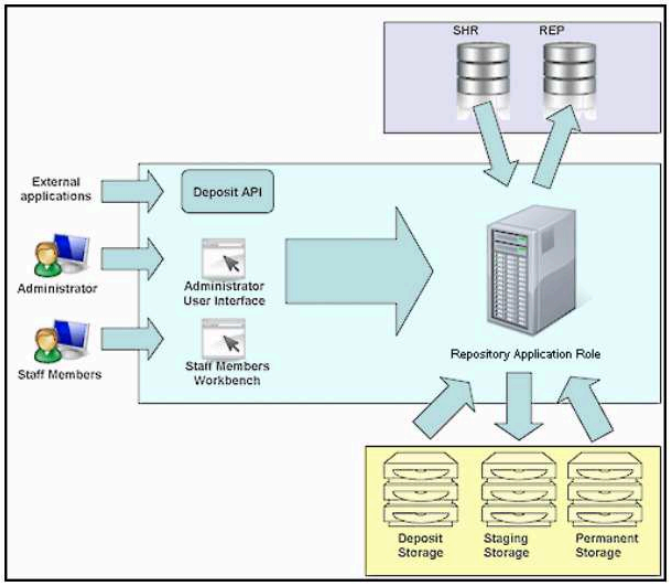


Deposit Directory Organization

Repository Application Role

The repository application role (sometimes known as the staging area) manages all activities associated with SIPs (validation stack, enrichment, 3A/TA, Web editor) and IEs (processes, publishing, Web editor).
The repository application role interacts with the following components, as illustrated in the figure below:

- **Database Schemas**
- **Deposit Storage**
- **Operational Storage**
- **Permanent Storage**



Staging Repository Application Role Architecture

Database Schemas

The repository application role uses the following database schemas:

- SHR, which contains configuration information required for processing SIPs and IEs (such as material flow and Producer profile configuration).
- REP, which stores information about content stored in the Operational Storage, including intellectual entities, metadata, SIP processing configuration, storage rules, and delivery rules.

Deposit Storage

The repository application role interacts with the Deposit Storage in order to move the stream files and METS files to the Operational Storage for further processing.
For more information about the Deposit Storage, see **Deposit Storage**.

Operational Storage

The Rosetta system stores the stream files, as defined by storage rules. (For more information on storage rules, see the Configuring Storage Rules section in Part III, Configuration Components, of the Rosetta Configuration Guide.)
Information about the physical location of the stream files, along with their identifiers, is stored in the Oracle database. These identifiers are also stored in METS files in order to enable the Rosetta system to establish a relationship between a METS file — which contains metadata about the stream files — and the files themselves.
METS files are grouped into submission information packages (SIPs) and stored as described in **Deposit Storage**.

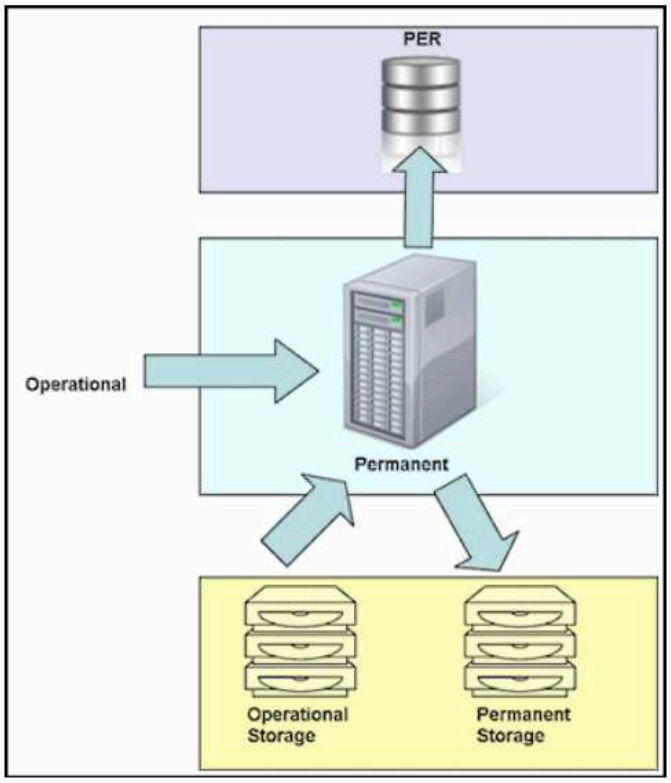
Permanent Storage

The repository application role interacts with the Permanent Storage when a content object (such as an intellectual entity, representation, or file) that has already been moved to the Permanent Storage must be edited.
Because no changes can be made in the Permanent Storage, the Rosetta system retrieves the content object from the Permanent Storage and stores this object in the Operational Storage.
After the object is edited, the Rosetta system returns this object to the Permanent Storage.
For more information about the Permanent Storage, see **Permanent Storage**.

Permanent Application Role Server

The permanent application role server provides permanent storage for content objects (such as IEs, representations, and files) that have been processed and approved.
The permanent application role server interacts with the following components, as illustrated in the figure below:

- Database Schemas
- Operational Storage
- Permanent Storage



Permanent Application Role Architecture

Database Schema

The permanent application role server uses the PER database schema to store information about content objects (such as intellectual entities, representations, and files) that are permanently stored in the Rosetta system.

Operational Storage

The permanent application role server interacts with the Operational Storage to retrieve content objects — that have been approved by Staff Members — from the Operational Storage, and move these objects to the Permanent Storage. For more information about the Operational Storage, see Operational Storage.

Permanent Storage

The Permanent Storage is designed to store intellectual entities (IEs) that were approved by Staff Members for permanent preservation. As a result, content objects that are stored in the Permanent Storage cannot be updated, deleted, or rearranged.

Whenever a content object must be changed (for example, its metadata requires editing), the Rosetta system moves it back to the Operational Storage. When the editing process is complete, the system creates a new METS XML as the new version of the IE and writes it to the Permanent Repository.
The Rosetta system stores files in the Permanent Storage, as defined by storage rules. (For more information on storage rules, see **Configuring Storage Rules**.)
Information about the physical location of the files, along with their permanent identifiers, is stored in the Oracle database. These permanent identifiers are also stored in METS files in order to enable the Rosetta system to establish a relationship between a METS file — which contains metadata about the permanent files — and the files themselves.
To store METS files, the Rosetta system creates a folder for each IE. The system uses the IE's permanent identifier as the name of the folder.

Delivery Application Role

The delivery application role is in charge of managing the Delivery requests that come from outside of Rosetta, for example from Primo patrons or the Aleph Web-OPAC.
The delivery application role can reside on the same server as other application roles (such as repository) but it can also reside on a dedicated server that can be accessed by external users.
Delivery interacts with both storages, operational and permanent, in order to deliver the requested metadata and files. It also interacts with the SHR scheme where the delivery rules are managed and the REP scheme where information about the files is stored.

Index Application Role

The index application role manages the SOLR indexing that allows the search of objects in Rosetta. The index application role can reside on the same server as other application roles (such as repository) but it can also reside on dedicated server(s) to allow better search performance. All index (IDX) servers must be accessible to all backoffice (REP) servers.
For more details about the SOLR, see the SOLR section of this guide.

Administering the Solr Server

The Solr indexing mechanism uses a properties configuration table that is managed by Ex Libris and updated as needed with every release or service pack. All METS and DNX elements are indexed and searchable.
The Solr server also indexes all attributes, each of which has a corresponding row in the UI Labels code table. Customers can control the label of each indexed attribute through this table.

Index Rank

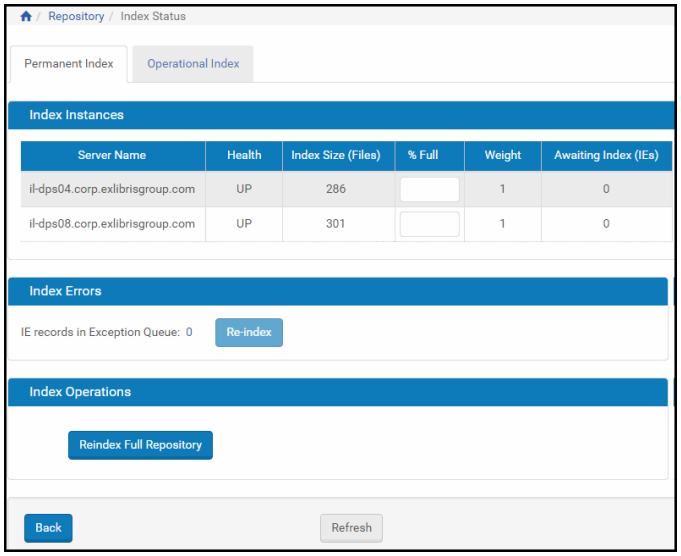
A new attribute added to the APPLICATION_VERSION table controls whether an index server (a server that has the IDX role) is active. Setting that attribute to one (1) means the index server is active and will be used by Solr for new/modified indexed information. Setting it to zero (0) means the index server will not be used by Solr for any new index rows. Modified index rows are still maintained in the same index server on which they were initially created.

Index Size and Location

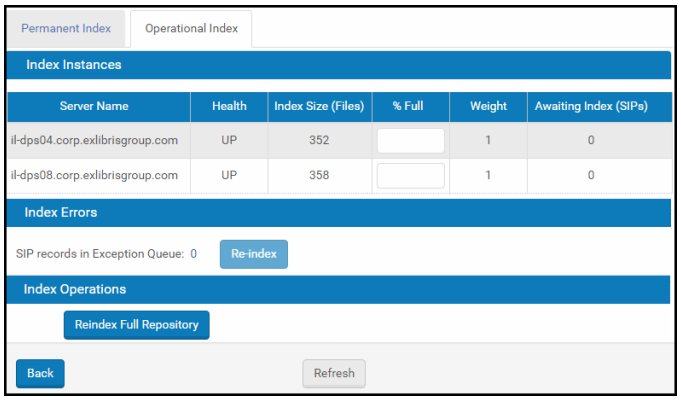
The index size parameter, solr_server_capacity, defines the index size for all index servers. The value is defined in KB. Generally, for every 1 MB of files indexed, the system requires 2 GB of space.
The index size is informational only and not enforced by Rosetta. Index size is displayed in the Index Status UI and is also used to calculate the percentage of currently used index size.
The location of the index files is stored in the global properties file (for each server that is defined as IDX in its application role). The property name is dps.storage.idx. The default value is /exlibris/dps/d4_1/profile/solr/data.

Index Status

A page displaying the indexing progress is available from Administration > Repository Configuration > General > Index Status. The Permanent Index tab displays the indexing progress for IEs and the operational Index tab displays the indexing progress for SIPs.



Permanent Index Status Page from the Administration UI



Operational Index Status Page from the Administration UI

Details about Index Instances appear in the first section of the Index Status page. In the second section, Index Errors, the number of IEs/SIPs that failed during indexing, is shown. An option to resubmit the failed IEs/SIPs and re-activate the index process is also available in this section (Re-index button).

If there are records in the Exception Queue, clicking the number opens a list of the PIDs in the IEs/SIPs. These IEs/SIPs are not searchable by their metadata, but they can be reached by the quick search (using their PID).

You can run a full repository re-index from the third section of the page, Index Operations. A full re-index should be run in the following cases:

- New index fields or functionality requiring a global re-index
- Index corruption or hardware failure for an index node
- Addition or removal of an index node

Index Shards

When the SOLR index is managed on multiple servers, and one of the servers is down, the search results may not be returned from this server.

When this happens, objects that are indexedon that server are still available for delivery through a deep link URL (for example, from Primo).

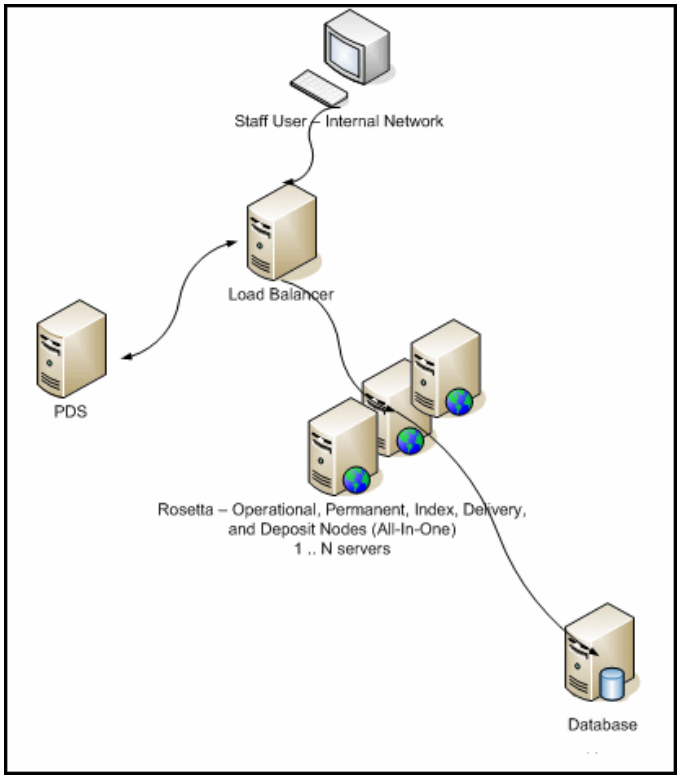
In addition, searching IEs in Rosetta by PID works for all IEs, even the ones that are indexed on the server that is down.

Rosetta-Supported Topologies

The following topologies are supported in Rosetta:

- All-in-One: See All-in-One Topology.
- 2-Tier: See 2-Tier Topology Diagram and 2-Tier Topologies.

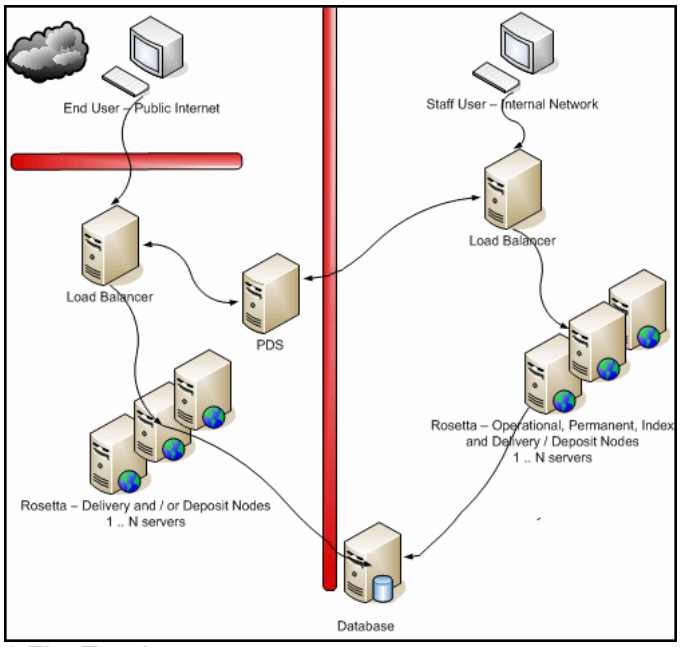
All-in-One Topology



All-in-One Topology

Deposit, Delivery, Staging, and Permanent all reside on the same server. This configuration can be scaled wide to support redundancy and through-put.

2-Tier Topology Diagram



2-Tier Topology

Deposit and/or Delivery are on a separate tier in a DMZ, and Staging and Permanent reside on an internal tier. This configuration can also be scaled wide, and is only used in environments where deposit or delivery functionality must be available to users outside of the organization.

PDS can be run either on Rosetta application servers with a REP or DEL application role or on an external server, either as a single instance or highly-available with a DB backend (with PDS 2.x). For detailed information and instructions on configuring PDS, see the Patron Directory Services Guide, available under Cross-Product Information in the Documentation Center.

Load Balancer Requirements

Load Balancer Requirements

Purpose

Load-balancing functionality can be achieved by either hardware or software components. The purpose of this section is to describe the functional aspects of the load balancer as used by Rosetta regardless of whether it is a hardware or software solution.

Sticky Session

Session stickiness must be provided by the load balancer.
A load balancer redirects end users, identified by source IPs, to the same physical Deposit server, during a single user session.

Forwarding

Configure the load balancer to pass the client's IP address in the X-Forwarded-For HTTP header value.

Load Balancing Web Requests

In any of the supported topologies, a load balancer should be defined for each Fully Qualified Domain Name (FQDN). A separate FQDN is required for each module/role that must be accessed separately. The following modules can be accessed separately:

- PDS
- Deposit
- Back Office
- Delivery

Configure the load balancer to redirect all requests to the specified domain names to be forwarded to the real server which made up the cluster (for example, port 1801). For example, requests to http://delivery.rosetta.myorg.org/ should redirect to http://delivery1:1801 and http://delivery2:1801.
If your topology requires that the Web service response includes the load balancer hostname (and not a specific application server hostname), make sure the load balancer forwards the request's original host header (for example, the mod_proxy ProxyPreserveHost directive is turned on).
All UI and Web service requests are directed to the appropriate load balancer FQDN as defined in global.properties. For further details, see **Managing Application Roles and Other Server Settings**.

Worker Pools for Back-End Activities

Workers are managed by the database direct communication with the application servers. No load balancer is necessary. Users can manage the number of workers (threads) that run on each server in order to optimize the performance of the system. For further details, see **Managing Workers**.

Application Roles

Managing Application Roles and Other Server Settings

The settings of each server, including the assignment of an application role to a server, are defined in the configuration file called global.properties, which is locatedunder $dps_dev/system.dir/conf/.

global.properties

This file holds all the settings that might vary between servers, while all the settings that are relevant for the entire installation are defined in the databaseא€™s general parameters table (see below).
Changes in the global_properties file take effect only after running the script $dps_dev/system.dir/bin/set_globals.sh. Rosetta must be shut down before running set_globals.sh.

2-Tier Topologies

In a two-tier topology, before changing the the APP_ROLE of a certain server, the System Administrator must update [module].server in global.properties with the correct FQDN for each module (before running the set_globals script). If a load balancer is applied, update load.balancer.[module].host and load.balancer.[module].port for each relevant module. If the changes affect more than just the global.properties, the System Administrator must run the $dps_dev/system.dir/bin/rosettings.csh script.
Cases that require the Administrator to run the rosettings.csh script include:

- Changing FQDN/LB name / port for deposit, operational, delivery application roles
- Changing the PDS URL / port
- Changing the database credentials
- Turning the PDS SSL on or off, or changing the PDS SSL port
- Setting the Rosetta SSL via load balancer

In any of these cases, we strongly recommend that you consult first with the Rosetta Support team.

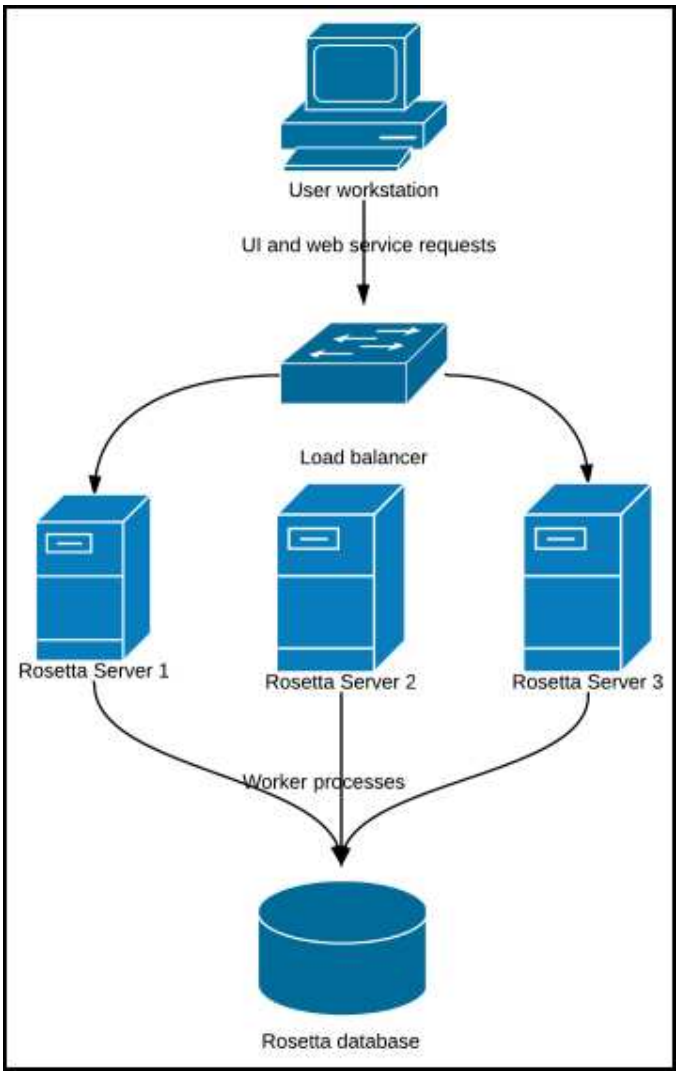
Rosetta Scalability

To increase the computing power of Rosetta, you can use more than one server with the same application role. In this case, all the servers that work with the same application role should have a load balancer that balances the load between them.
When adding or modifying the load balancer settings, the URL must be changed in the global.properties file of each server. To apply the changes, the System Administrator must run the $dps_dev/system.dir/bin/rosettings.csh script. Changing the application role of a server or adding a load balancer should be undertaken only after consulting the Rosetta Support team.

Managing Workers

Rosetta uses worker threads to handle back-end processing. Such back-end processes include:

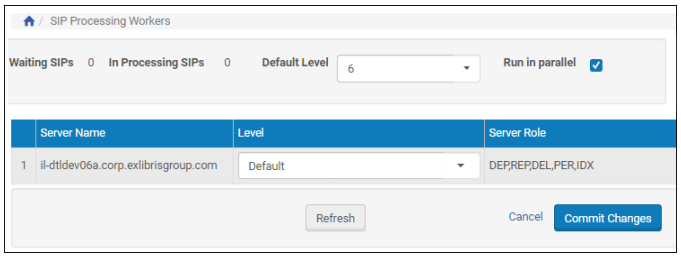
- Deposit work
- Event processing
- Indexing
- SIP processing
- Permanent processing
- SIP loading



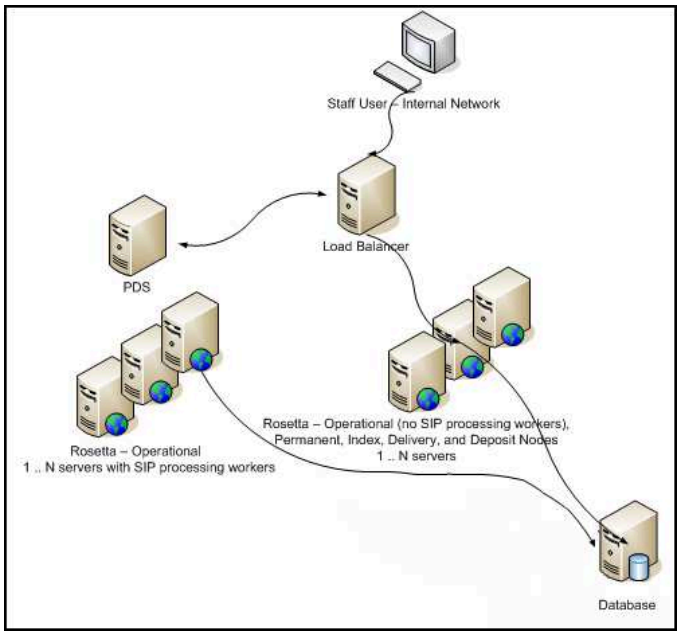
Rosetta Workers

SIP processing resources can be configured on each server in order to optimize the performance of the system. These settings are configured out-of-the-box for a server of typical computing power in line with the Rosetta system requirements. Installations with larger or smaller servers may need to adjust these settings to achieve an optimum utilization of system resources. In addition, worker threads can be configured to create a server that handles only

specific functionality within an application role. A typical use case could be the need to allocate enough computing power for front-end activity (such as Web deposit and Delivery) while back-end activities such as maintenance tasks run in the background.
To adjust these settings, log on to the Rosetta Administration site and follow the path from the Administration page: System Configuration > General > SIP Processing Workers.



SIP Processing Workers
The Level parameter is a 0-10 value that determines the SIP processing load for the entire environment by setting all servers to default and adjusting the default value, or by individually adjusting each SIP processing server (DEP, REP, and PER).
To ensure UI high-availability, lower the level on one or more of your REP servers, or consider reserving one of your REP servers for UI functionality by setting its level to 0.
An indication of a balanced and fully-utilized SIP processing configuration is one in which the number of waiting SIPs is constantly approaching 0 and the number of SIPs in process is steady (the actual number naturally depends on the number of servers). Adjust the Default Level to apply changes to all servers set to Default, or adjust servers' levels individually. If the number of waiting SIPs is rising and SIP processing level is at maximum (and/or cannot be increased in order to reserve UI resources), you may require additional hardware resources to achieve better throughput. Please consult the Rosetta Support Team and Ex Libris Sizing Manager for further analysis.
Users with heavy SIP loading needs may also want to reserve Operational server resources for end-user-facing modules by detaching the Operational servers that handle SIP processing from the load balancer. The load balancer would direct all requests to servers with a low level of SIP processing workers (or none at all), ensuring high availability. Currently these servers are not dedicated to end user requests, as they also handle Process Automation.



High Availability Server Flow
Libraries that are interested in setting up such a topology should contact Rosetta Support.

To have several worker processes allocated per SIP, select the Run in Parallel check box. This allows Rosetta to process a large amount of files much more quickly.

SIP Processing Throttling

SIP processing can be configured to refrain from processing additional SIPs when the system load average (during the last minute) is above a certain level. The load average is calculated similar to the top UNIX utility, taking into consideration memory usage and CPU utilization. This setting is useful when unusually large SIPs are deposited sporadically and unpredictably, in order to prevent Rosetta from unexpectedly running out of resources.

Managing Plug-Ins

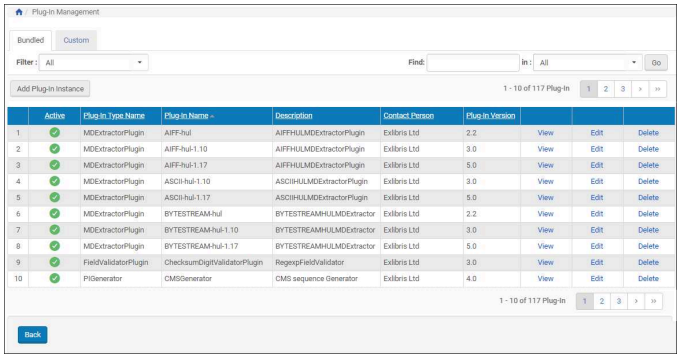
Managing Plug-Ins

Managing plug-ins with Rosetta involves accessing file directories on the server and working with the Rosetta Administration interface.
Before you begin:

- Have the JAR file containing your plug-in.

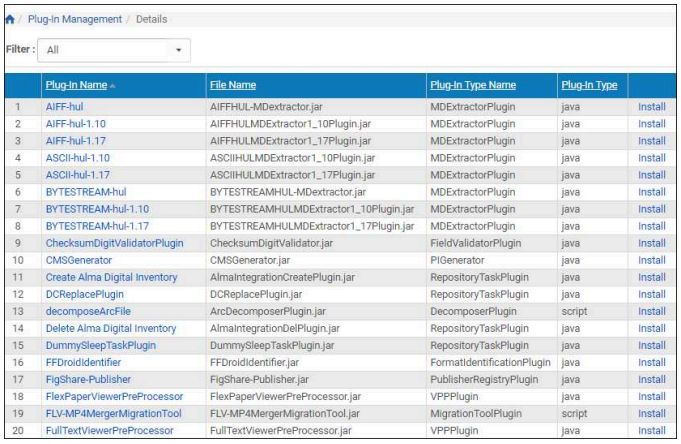
To install and configure the plug-in:

1. Upload the JAR file to the application custom plug-in input directory.
   The location of the input directory is relative to the operational_shared directory (can be found in the HfrGeneralParameters table's operational_shared parameter). The full path is .../operational_shared/plugins/custom/.
2. Install the plug-in using the application plug-in management UI.
   a. On the custom Plugin List page (Administration > System Configuration > General > Plug- In Management), click the Add Plug-In instance button.



Plug-In Management Page

A list of available plug-ins opens. The plug-in you installed should appear in the list. Scroll or use the Filter or Find feature to help you find it.



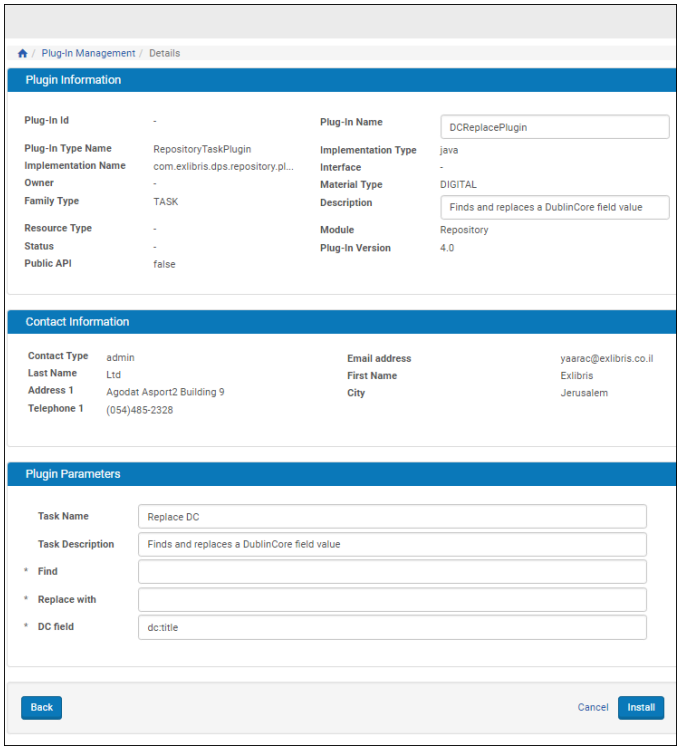
Plug-In Management / Details

Filter : All

	Plug-In Name	File Name	Plug-In Type Name	Plug-In Type	
1	AIFF-hul	AIFFHUL-MDextractor.jar	MDExtractorPlugin	java	Install
2	AIFF-hul-1.10	AIFFHULMDExtractor1_10Plugin.jar	MDExtractorPlugin	java	Install
3	AIFF-hul-1.17	AIFFHULMDExtractor1_17Plugin.jar	MDExtractorPlugin	java	Install
4	ASCII-hul-1.10	ASCIIHULMDExtractor1_10Plugin.jar	MDExtractorPlugin	java	Install
5	ASCII-hul-1.17	ASCIIHULMDExtractor1_17Plugin.jar	MDExtractorPlugin	java	Install
6	BYTESTREAM-hul	BYTESTREAMHUL-MDextractor.jar	MDExtractorPlugin	java	Install
7	BYTESTREAM-hul-1.10	BYTESTREAMHULMDExtractor1_10Plugin.jar	MDExtractorPlugin	java	Install
8	BYTESTREAM-hul-1.17	BYTESTREAMHULMDExtractor1_17Plugin.jar	MDExtractorPlugin	java	Install
9	ChecksumDigitValidatorPlugin	ChecksumDigitValidator.jar	FieldValidatorPlugin	java	Install
10	CMSGenerator	CMSGenerator.jar	PIGenerator	java	Install
11	Create Alma Digital Inventory	AlmaIntegrationCreatePlugin.jar	RepositoryTaskPlugin	java	Install
12	DCReplacePlugin	DCReplacePlugin.jar	RepositoryTaskPlugin	java	Install
13	decomposeArcFile	ArcDecomposerPlugin.jar	DecomposerPlugin	script	Install
14	Delete Alma Digital Inventory	AlmaIntegrationDelPlugin.jar	RepositoryTaskPlugin	java	Install
15	DummySleepTaskPlugin	DummySleepTaskPlugin.jar	RepositoryTaskPlugin	java	Install
16	FFDroidIdentifier	FFDroidIdentifier.jar	FormatIdentificationPlugin	java	Install
17	FigShare-Publisher	FigShare-Publisher.jar	PublisherRegistryPlugin	java	Install
18	FlexPaperViewerPreProcessor	FlexPaperViewerPreProcessor.jar	VPPPlugin	java	Install
19	FLV-MP4MergerMigrationTool	FLV-MP4MergerMigrationTool.jar	MigrationToolPlugin	script	Install
20	FullTextViewerPreProcessor	FullTextViewerPreProcessor.jar	VPPPlugin	java	Install

List of Available Plug-Ins

b. When you find the plug-in you want to install, click the corresponding Install link.
c. The Plug-In Information page for that plug-in opens.



Information for Existing Plug-In

d. Enter a name in the Plug-In Name field and a Description to reflect it.

Because you are installing and not editing, Rosetta will create a new entry in the plug-in list.

e. Add or change values for all required parameter fields.

If you are installing a variation on an existing plug-in, you must rename the plug-in and enter values in the required parameter fields before Rosetta will allow the installation to be completed.

f. Click install.
   The new plug-in is installed in the system and appears on the Plug-In List page, available for use.

To create another plugin instance with different initialization parameters values, repeat step Step d, making sure the plug-in name is unique.
To change the initialization parameters of an installed plug-in:

1. From the Plug-In Management page (Administration > System Configuration > General > Plug- In Management), find the name of the plug-in whose parameters you want to change and click its corresponding Edit link.
2. Change the initialization parameter values and click Save.

To stop using a given plug-in:
From the Plug-In Management page (Administration > System Configuration > General > Plug- In Management), select the Delete link corresponding to the row of the plug-in you want to disable.
To upgrade a (custom) plug-in:
From the Custom tab of the Plug-in List page (Administration > System Configuration > General > Plug- In Management), click the Upgrade button.
An upgraded version of the plug-in is deployed.

Disk Space Management

Disk Space Management

Administrators can configure how disk space is used on the Storage Server, as described in the following sections:

- **Checking Disk Space**
- **Freeing Disk Space**

Checking Disk Space

Administrators can view the following disk space parameters in the directory:

- Total disk space
- Used disk space3
- Available disk space

To check disk space in the directory:
Type the following commands:

```
cd /exlibris

df -kh
```

The Rosetta system provides information about disk space, as shown in the following example:

```
dps.corp.exlibrisgroup.com-p1(1) >>cd /exlibris

dps.corp.exlibrisgroup.com-p1(1) >>df -kh

Filesystem            size   used avail capacity Mounted on

filer01:/vol/data8/il-dtldev07a

                      30G    26G   3.7G    88%    /exlibris
```

Administrators can run the df -kh command for each deposit area in order to see the total amount of disk space used by deposited activities. The location of each deposit area is configured in the HFrGeneralParameter table, under the parameter name logic_deposit_area.

Freeing Disk Space

Administrators can free an additional amount of disk space by deleting old log files from the log directory (whose alias is dps_log).

Rosetta Storage Management

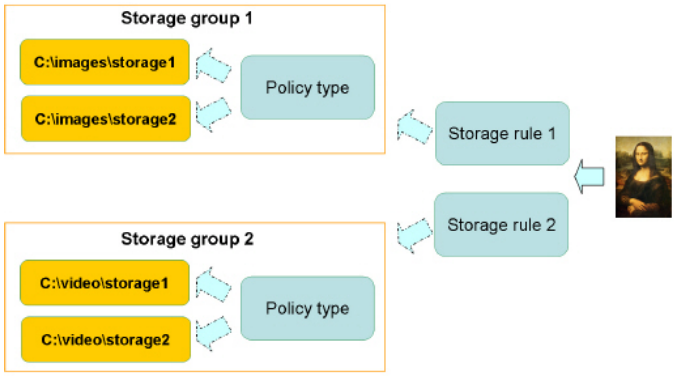
Understanding Storage Architecture

Understanding Storage Architecture

Administrators can define how Rosetta stores content on the various servers: Deposit, Staging, and Permanent. Administrators configure the following components separately for the Staging Server and the Permanent Repository:

- Storages, which are the storage areas. A storage can be a Network File System (NFS) or another type of data storage solution that Rosetta can access through HTTP requests (such as Amazon S3).
- Storage groups, which aggregate individual storages that contain similar types of data, such as metadata or intellectual entities (IEs).
- Because a storage group can contain multiple storages, Administrators configure storage group policies that define the storage to be used for each content object.
- Storage rules, which determine the storage group to be used. Each storage rule consists of the following parameters:
  - Input parameters (such as content object type), with which the Rosetta system compares actual parameters of a content object
  - Output parameters, which define the storage group to be used, if actual parameters of a content object match input parameters of the storage rule

The diagram below shows the organization of the storage components in the Rosetta system:



Storage Components
The information flow consists of the following stages:

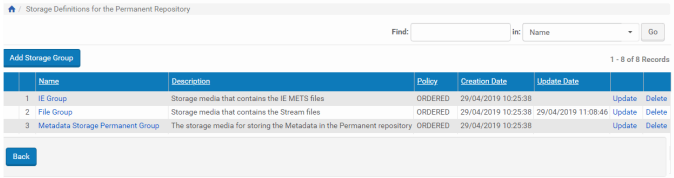
1. After a content object is moved from the Deposit Server to the Staging Server, the Rosetta system sequentially compares the actual content object parameters with the input parameters of the storage rules.
2. When the content object parameters match the input parameters of a storage rule, the Rosetta system moves the object to the storage group, as defined in the output parameters of the storage rule.

3. Within the storage group, the Rosetta system moves the object to a storage, as defined in the policy type.

Configuring Storage Groups

Configuring Storage Groups

Administrators work with storage groups using the Storage Groups List page.



Storage Groups List page

To access the Storage Groups List page, on the Administration page go to Repository Configuration > Storage Rules and Definitions > Storage Definitions for the Permanent Repository.

The following actions (on the following pages) can be performed on the Storage Group List page:

- **Adding a Storage Group**
- **Re-Ordering the List of Storages in a Storage Group**
- **Updating a Storage Group**
- **Deleting a Storage in a Storage Group**

Adding a Storage Group

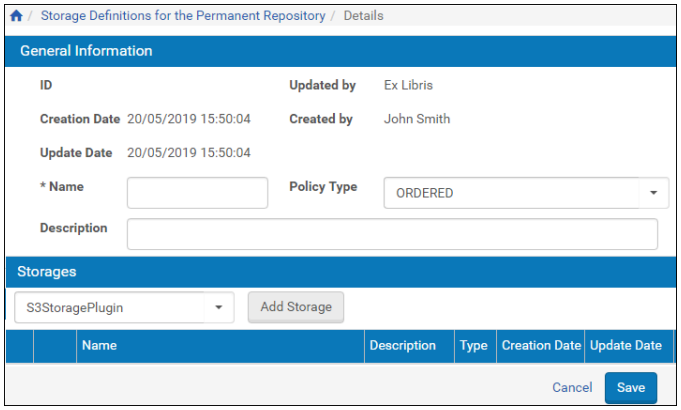
Administrators can add a new storage group in order to aggregate similar types of data, such as metadata or intellectual entities (IEs). When creating a storage group, Administrators must add at least one storage to save the storage group in the Rosetta system.

Administrators can add as many storage groups as are needed. When adding a storage group, Administrators perform the following actions:

1. Define general information about a storage group
2. Add at least one storage to the storage group

To add a new storage group:

1. On the Storage Group List page (see **Configuring Storage Groups**), click Add Storage Group. The Storage Group Edit page opens.

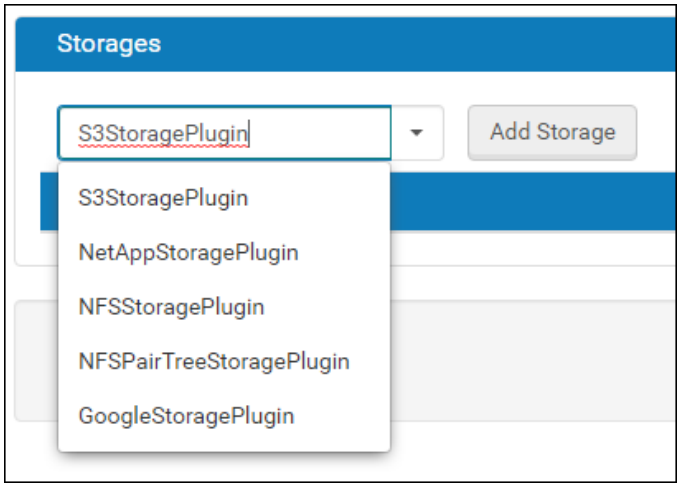


Storage Group Edit Page

2. In the General Information pane, complete the fields as described in the following table:
Storage Group General Information Fields

Field	Description
Name	The name of the storage group.
Description	The description of the storage group.
Policy Type	The criteria that determine which storage is used. The following options are available: • Random policy - Randomly select a storage that has adequate space available. • Orderly policy - Use the first storage (based on the storage order) that has adequate space available.

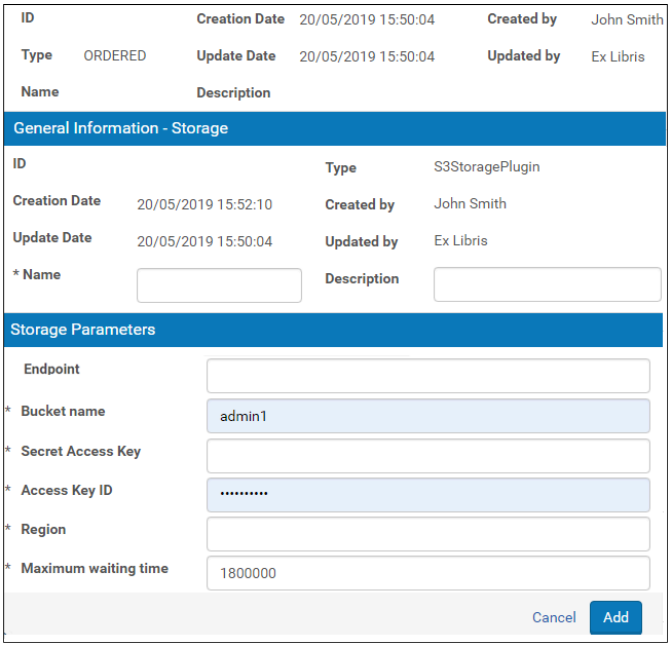
3. In the Storages pane, select your storage plug-in.



Storage Plug-ins

The following options are available:

a. S3 Storage Plugin
A storage solution based on the Amazon S3 cloud service. In this situation, Rosetta can be connected to the cloud and the Permanent repository is located on Amazon servers.

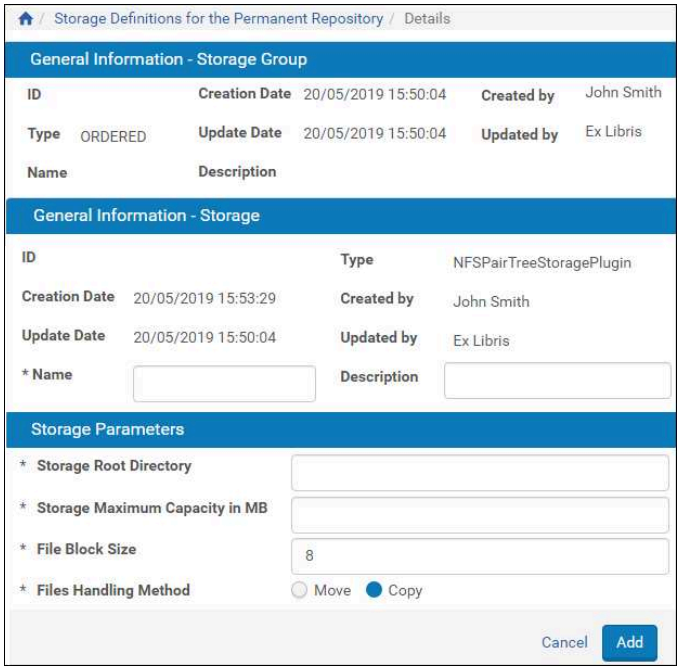


S3StoragePlugin

With this storage plug-in, the following parameters are used:

- Endpoint – the URL of the entry point of the AWS Web service
- Bucket name – This is the term used by Amazon for describing the location of the files in their cloud.
- Secret Access Key – The access key for the bucket.
- Access Key ID – The key ID.
- Maximum waiting time – The time Rosetta will wait for response from Amazon when retrieving files. When the time elapses, Rosetta displays a message to the user that the file is not accessible.

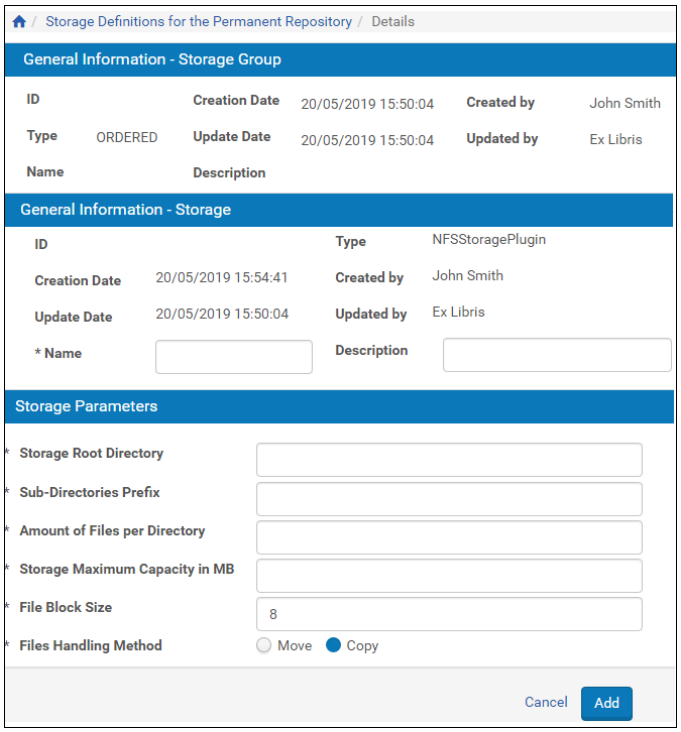
b. NFS Pair Tree Storage Plug-in
An NFS plug-in in which the directory structure inside the file system is based on the files' fixity value and not dates.



NFSPairTreeStoragePlug-in

In the pair tree storage, files are located in a folder that is created based on an MD5 hash of their PID. This increases the speed and efficiency when search algorithms are used to retrieve files.
This plug-in can be used to implement different logic for folders' naming conventions.

c. NFS Storage Plugin
   The original out-of-the-box plug-in provided by Rosetta. Files are stored conventionally, based on their date of upload.

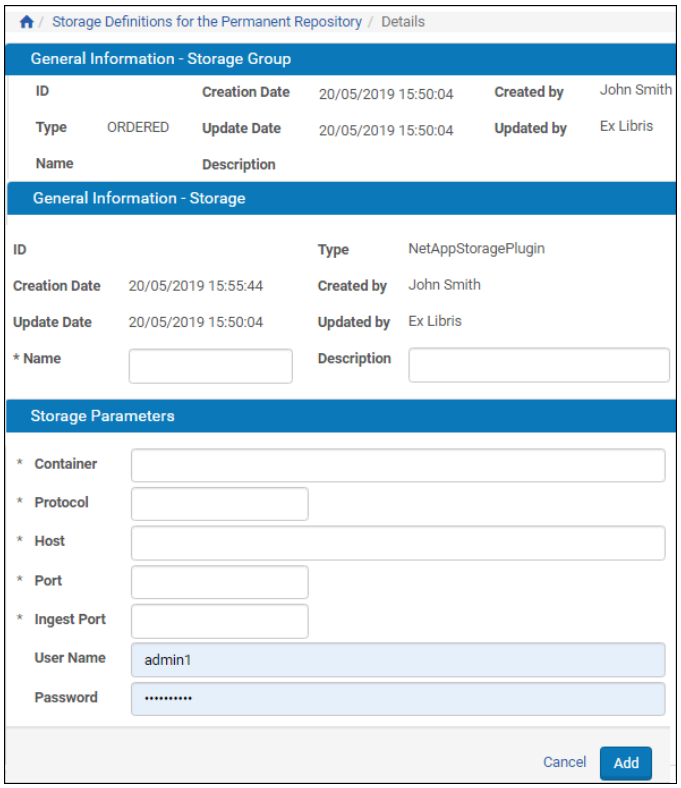


NFSStoragePlugin

Parameters for the NFS storage plug-in are described below.
Storage Parameters for NFS Storage

Field	Description
Storage Parameter Name	The name of the storage parameter.
Root Directory	The root directory of the storage. e:\storage1\data\2008\06\10.
Directory Prefix	The directory prefix that is part of the directory name. For example, if the directory prefix is file$, the system creates the following directories: • e:\storage1\data\2008\06\10\file$1 • e:\storage1\data\2008\06\10\file$2 • e:\storage1\data\2008\06\10\file$3 The directory prefix is suffixed with a sequential counter.
Files Per Directory	The maximum number of files that can be saved in the storage. When the maximum number of files that can be saved in the directory is reached, the system creates a new directory.
Storage Max Size (MB)	The total amount of data that can be saved in the storage.
Storage Block Size	The storage block size is used for an exact calculation of the available disk space. It is strongly recommended that you leave this field at its default value.
Files Handling Method	Select one method from the three available: • Copy the files from their previous location (Deposit or Operational) to this storage (Operational or Permanent) so that two copies of the processed files exist in the system. • Move the files from their previous location (Deposit or Operational) to this storage (Operational or Permanent) so that only one copy of the file exists in the system. • Soft Link for Operational storage definitions, useful for handling large files, thus avoiding copying and moving them twice. Because derivative copy files are not moved and copied to the Permanent repository, they must be moved and copied from the Deposit file system to the Operational repository. Otherwise, if soft-copied, they might get lost when the Deposit file system is cleaned. Therefore, an out-of-the-box rule has been added that makes sure that these files are always copied. Move policy should only be used where operational and permanent storage are on the same mount; otherwise, there is no performance advantage (since streams are always removed from operational after SIP reaches permanent).

d. Net App Storage Plugin
The NetApp plug-in is a storage solution by NetApp, Inc., that integrates with the Rosetta storage system.

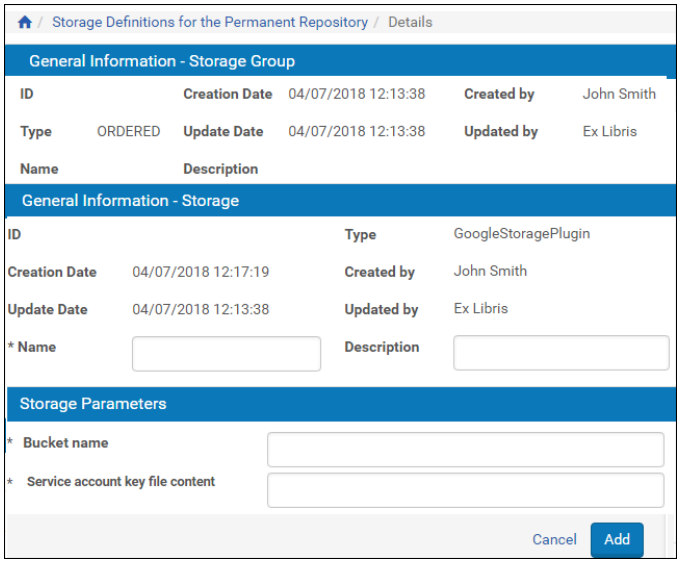


NetApp Plug-in

The NetApp application manages the storage of files in a combination of tapes and disks. Rosetta interacts only with the management layer, which stores and retrieves files for Rosetta.

e. Google Storage
The Google plug-in is a storage solution by Google, that integrates with the Rosetta storage system.

Google storage is supported for permanent only.



Google Storage Plug-In

The following are the parameters of the Google Storage Plug-in:
Storage Parameters for NFS Storage

Field	Description
Bucket name	The name of the directory in Google in which the files are stored.
Service account key file content	The authentication key

4. Click Add Storage.
5. The Storage Edit page for the storage plug-in you chose opens.
6. In the General Information - Storage pane, enter a name and description for the storage.
7. In the Storage Parameters pane, complete the fields as described in the relevant section above. (For NetApp parameters, see the documentation for the NetApp storage plug-in.)
8. Click Save. The new storage is displayed on the Storage Group Edit page.

The storage is not added to the Rosetta system until you click Save on the Storage Group Edit page (see Step 8).

9. To add more storages, repeat Step 3 - Step 6.
10. On the Storage Group Edit page, click Save.

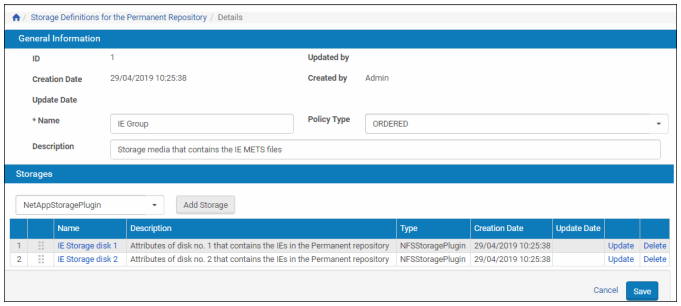
The storage group and its related storages are saved in the Rosetta system.

Re-Ordering the List of Storages in a Storage Group

To define a storage that must be used for a specific content object, the Rosetta system compares the parameters of the storage with the content object parameters (such as file size), using the policy type.
Storages are analyzed in the same order that they are displayed on the List of Delivery Rules page. The Rosetta system uses the first storage found that matches the parameters of the content object.
An Administrator can re-order storages to change their priority.

To re-order the list of storages:

1. On the Storage Groups List page



Storage Group Edit page

2. In the Storages pane, use the up and down arrows to change a storage's priority.
3. Click Save.

The Rosetta system now analyzes the storages in the updated order.

Updating a Storage Group

Administrators can update a storage group to add more storages, modify the existing storages, or change the descriptive information of the storage group.
To update a storage group:

1. On the Storage Group List page (see **Configuring Storage Groups**), locate the group you want to update and click Update. The Storage Group Edit page opens.
2. Modify the fields you want to update and click Save.

The storage group is updated in the Rosetta system.

Deleting a Storage in a Storage Group

Administrators can delete a storage in a storage group.
The Rosetta system does not enable Administrators to delete a storage that contains data.
To delete a storage in a storage group:

1. On the Storage Group List page (see **Configuring Storage Groups**), locate the storage group with which you want to work and click Update. The Storage Group Edit page opens.
2. In the Storages pane, locate the storage you want to delete and click Delete. The confirmation page opens.
3. Click OK.

The storage is deleted from the Rosetta system.

Deleting a Storage Group

Administrators can delete a storage group.
The Rosetta system does not enable Administrators to delete a storage group that contains at least one storage item.
To delete a storage group:

1. On the Storage Group List page (see **Configuring Storage Groups**), locate the group you want to delete and

click Delete. The confirmation page opens.

2. Click OK.

The storage group is deleted from the Rosetta system.

Storage Space Checker

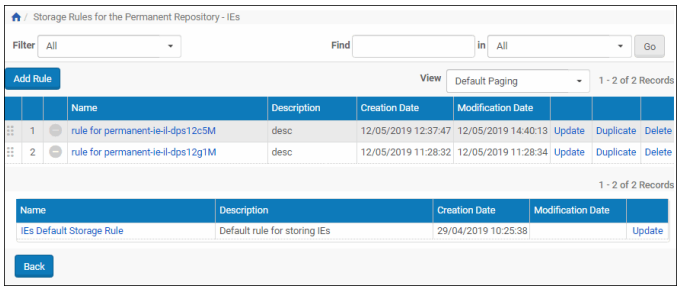
Storage Space Checker

A Storage Space Checker (implemented as a Startup Check plug-in) checks each storage group for available space, as defined by the aggregation of the Storage Max Size values for all storages (with a Storage Max Size) in a given group. Storage groups comprised of one or more storages with no Storage Max Size (for example, Amazon s3 storage) are not checked.
The default thresholds are 5% for a warning and 1% for an error, and can be defined in the UI of the Storage Space Checker plug-in instance (under Plugin Management). Threshold values are global for all (relevant) storage groups. Warnings and errors appear in the System Check UI with the details of the relevant storage group.
Rosetta storage space definitions are logical and presume the existence of available physical disk space, which must be monitored separately.

Configuring Storage Rules

Configuring Storage Rules

Administrators configure storage rules to determine which storage group the Rosetta system must use for specific content objects.
Administrators define storage rules from the Rule List page and the Rule Details page.



Rule List page
To access this page from the Administration page, go to Repository Configuration > Storage Rules and Definitions, and select one of the options for configuring storage rules (for example, Permanent Repository - IEs, or Operational Repository).
The following actions (on the following pages) can be performed from the Rule List page:

- Adding a Storage Rule
- Updating a Storage Rule
- Defining the Default Storage Rule
- Re-ordering the List of Storage Rules
- Deleting a Storage Rule
- Activating and Deactivating a Storage Rule

Rule Details Page

Rule Details Page

If you are adding or editing a specific rule, you will work on the Rule Details page to define the parameters of the rule. See the Operators Used in Rule Parameters section for details on how to use operators in parameters.

Operators Used in Rule Parameters

The following operators are used for specific types of parameter data.

String Values
String values are words that are not separated by a comma (,), for example, one Producer name (John Smith), one MIME type (audio/mp3), one error code, one Format ID). String values use the following operators:

- Equal – The string and the input value must match exactly.
- Contains – The string and the input value must match partially with the '*' character.

List of Strings
A list of strings is a list of string values separated by a comma (,) sometimes populated by a widget. Lists of strings use the following operators:

- List Contains – used when each error returned should match exactly a single given error in the rule.
- List Equals - Used when the order of the items in the list and the list itself should match exactly. For example, a rule defined as "Invalid page dictionary object, Invalid object number in cross-reference stream" will match to the actual output from JHOVE – "Invalid page dictionary object, Invalid object number in cross-reference stream."

Numeric Fields
Numeric fields (for example, file size) use numbers as matching and comparison values.

- Greater Than (>) – The input value should be greater than the parameter value.
- Less Than (<) - The input value should be less than the parameter value.
- Equal (=) - The input value should be equal to the parameter value.
- Not Equal (! =) - The input value should be not equal to the parameter value.

Date Fields
Date fields (such as Creation Date) compare date values with time operators.

- After – The input date should be later than the parameter date value.
- Before - The input date should be earlier than the parameter date value.

- Equal (=) - The input date should be the same as the parameter date value.
- Not Equal (! =) - The input date should not be the same as the date parameter value.

Any
All fields can use this operator for indicating that any input value will be accepted by the rule. For example, if the ‘Any’ operator is used in the Producer Name field, the rule can match all Producers.
The following table summarizes the possibilities for matching between the rule parameter values and the run-time values:

Possible Matches Between Rule Parameter and Run-Time Values

Run-time Value	Operator	Possible Rule Values	Result
Demo Producer	Equal	Demo Producer	Match
Demo Producer	Contains	Demo*	Match
image/tiff or image/bmp	In List	Image/tiff, image/bmp	Match
image/tiff, image/bmp	List Equals	Image/tiff, image/bmp	Match
grey or gray	In List with Regular Expression	gr[ea]y	Match
12345	<, >, =, !=	10000	< - No match > - Match = - No match != - Match
23/11/2011	Before, After, =, !=	23/11/2011	Before - No match After - No match = - Match != - No match

To define Boolean logic when using multiple conditions, select one of the following options between conditions:

- OR
- AND (default)

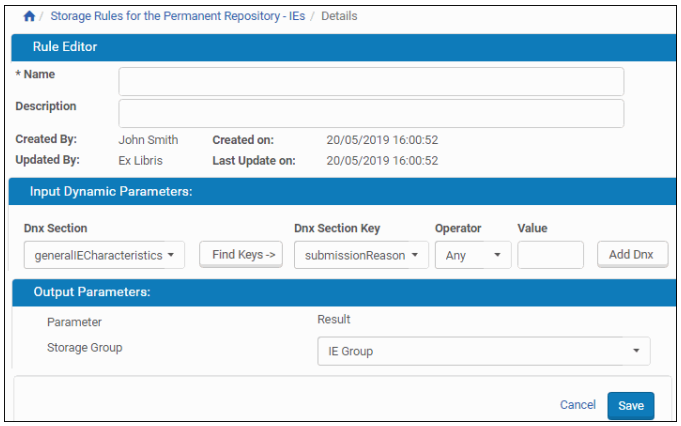
The Boolean connector between different types of attributes (for example, IE Attributes and File Attributes) is always AND.

Adding a Storage Rule

Adding a Storage Rule

Administrators can add a new storage rule to define the automatic selection of a specific storage group. Administrators can add as many storage rules as needed.
To add a new storage rule:

1. On the Rule List page (see **Configuring Storage Rules**), click Add New Rule. The Rule Details page opens.



Rule Details Page

2. Enter a Name and a Description for the new rule.
3. Enter parameters for the rule. Use the **Operators Used in Rule Parameters** section to create parameters.
4. Click Save.

The Rosetta system uses the storage rule in order of priority, as described in **Re-ordering the List of Storage Rules**.

Updating a Storage Rule

Administrators can update a storage rule in order to modify input and output parameters.
To update a storage rule:

1. On the Rule List page (see **Configuring Storage Rules**), locate the rule you want to update and click Edit. The Rule Details page opens.
2. Modify the fields you want to update, and then click Save.

The storage rule is updated in the Rosetta system.

Defining the Default Storage Rule

Administrators must configure a default storage rule. In the event that the Rosetta system cannot find a rule that matches any of the storage rules, the system uses the default storage rule.
To define the default storage rule:

1. On the Rule List page (see **Configuring Storage Rules**), in the Default Rule pane, click Update. The Default Rule Details page opens.
2. In the Output Parameters pane, in the Result drop-down list, select the storage group to be used by default.
3. Click Save.

The Rosetta system now uses the selected storage rule as the default rule.

Re-Ordering the List of Storage Rules

To define a storage group that must be used for storing an object, the Rosetta system compares the input parameters defined in a storage rule with the parameters of the storage group.
Storage rules are analyzed in the same order in which they are displayed on the Rule List page. The Rosetta system uses the first storage rule found that matches the parameters of the storage group.
To re-order the list of storage rules:

1. On the Rule List page (see **Configuring Storage Rules**), select each relevant rule and use the up and down arrows to change the rule's priority.
2. Click Save.

The Rosetta system now processes the storage rules in the defined order.

Deleting a Storage Rule

Administrators can delete an existing storage rule. After a storage rule is deleted, it is no longer available to the Rosetta system for matching.
To delete a storage rule:

1. On the Rule List page (see **Configuring Storage Rules**), locate the storage rule you want to delete and click Delete. The confirmation page opens.
2. Click OK.

The storage rule is deleted from the Rosetta system.

Activating and Deactivating a Storage Rule

Administrators can activate or deactivate a storage rule. After a storage rule is deactivated, it is no longer available to the Rosetta system for matching.
On the Rule List page, the status of the storage rule is indicated by the check mark in the Active column:

- Yellow - The storage rule is active.
- Grey - The storage rule is inactive.

To activate or deactivate a storage rule:

1. On the Rule List page (see **Configuring Storage Rules**), locate the storage rule you want to activate or

deactivate.

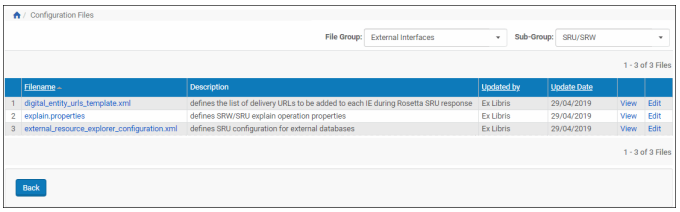
2. In the Active column, click the check mark. The check mark in the Active column indicates the new status.

The storage rule is changed from active to inactive, or from inactive to active.

Configuring External Metadata

Configuring External Metadata: SRU/SRW

External metadata about content objects in Rosetta is stored in external systems such as collection management systems. To enable Rosetta to communicate with external systems, Administrators can edit the configuration files from the Administration page by clicking Repository Configuration > External Interfaces > SRU/SRW sources and definitions.The Configuration Files page opens to the SRU/SRW sub-group (below).



SRU/SRW Configuration Files

The fields are described in the following table:

External Metadata Configuration Files

File	Defines...
explain.properties	Parameters of the Rosetta system server
external_resource_explorer_configuration.xml	Parameters of an external system server

The following tasks must be performed in the configuration of SRU:

- **Configure Rosetta Parameters - External Resource File**
- **Configure Rosetta Parameters - Explain File**
- **Adaptation Needed in the SRU Server**

Configure Rosetta Parameters - External Resource File

Use the following procedure to configure Rosetta parameters for external files:

1. With the File Group field set to External Interface and the Sub-Group set to SRU/SRW, click Edit for the file named external_resource_explorer_configuration.xml.
   The file opens on the page. It contains all SRU configurations that are delivered with Rosetta.
2. Add your configuration to the file. The parameters that should be modified are:

- baseUrl - the URL of the system that acts as an SRU server (CMS)

- version - SRU version (currently 1.1 is supported by Rosetta)
- operation - since Rosetta is the client, "searchRetrieve" is the only option
- recordSchema - should be "dps"
- indexName - should be "rec.id"
- recordPacking - should be xml
- updateUrl - in cases where the server can get a response for updating the exists-in-Rosetta flag, the url should be added here (currently available only for Voyager and Aleph. For other systems - leave it empty.

The following is an example of an SRU configuration for Aleph and Voyager:

```
<?xml version="1.0" encoding="UTF-8"?>

<ExternalResourceExplorer xmlns="http://www.loc.gov/zing/srw/configuration/"xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:xml="http://www.w3.org/XML/1998/namespace">

<RepositoryName name="qa654mkdb">

<protocol type="SRW">

<parm name="baseUrl">http://10.100.2.36:17191/voyager</parm>

<parm name="version">1.1</parm>

<parm name="operation">searchRetrieve</parm>


<parm name="recordSchema">dps</parm>

<parm name="indexName">rec.id=</parm>

<parm name="recordPacking">xml</parm>

<parm name="updateUrl">http://10.100.2.36:17114/vxws/updateDPSFlag</parm>

</protocol>

</RepositoryName>


<RepositoryName name="USM01" type="ALEPH">

<protocol type="SRW">

<parm name="baseUrl">http://il-aleph07:5667/usm01</parm>
```

<parm name="version">1.1</parm>

<parm name="operation">searchRetrieve</parm>

<parm name="recordSchema">dps</parm>

<parm name="indexName">rec.id=</parm>

<parm name="recordPacking">xml</parm>

<parm name="updateUrl">http://il-aleph07:8997/X?op=ros_doc&library=USM01</parm>

<parm name="detachUrl">http://il-aleph07:8997/X?op=ros_doc_del&library=USM01</parm>

</protocol>

</RepositoryName>

</ExternalResourceExplorer>

It should be noted that RepositoryName must equal the name that is returned in the SRU/SRW response. For example, if
<identifier>DPS:10.100.2.36:ALEPH01:9611</identifier>
then the name **attribute of** RepositoryName **must be** ALEPH01, **as in**
<RepositoryName name="ALEPH01" type="ALEPH">
To get the parameters from the SRU server, call the explain URL.
The following is an example of the explain URL from the Aleph system:

<zs:explainResponse>

<zs:version>1.1</zs:version>

<zs:record>

<zs:recordSchema>http://explain.z3950.org/dtd/2.0/</zs:recordSchema>

<zs:recordPacking>xml</zs:recordPacking>

<zs:recordData>

<explain>

<serverInfo>

<host>il-aleph07</host>

<port>9997</port>

<database>usm01</database>

</serverInfo>

<indexInfo>

<set identifier="info:srw/cql-context-set/1/cql-v1.1" name="cql"/>

<set identifier="info:srw/cql-context-set/1/dc-v1.1" name="dc"/>

<set identifier="http://zing.z3950.org/cql/bath/2.0/" name="bath"/>

<index id="4">

<title>title</title>

<map>

<name set="dc">dc.title</name>

</map>

</index>

<index id="48">

<title>identifier</title>

<map>

<name set="dc">dc.identifier</name>

</map>

</index>

<index id="12">

<title>rec.id</title>

<map>

<name set="dc">rec.id</name>

</map>

</index>

</indexInfo>

</explain>

</zs:recordData>

</zs:record>

<zs:diagnostics>

<diagnostic>

<uri>info:srw/diagnostic/1/7</uri>

<message>Mandatory parameter not supplied</message>

<details>version</details>

</diagnostic>

<diagnostic>

<uri>info:srw/diagnostic/1/7</uri>

<message>Mandatory parameter not supplied</message>

<details>operation</details>

</diagnostic>

</zs:diagnostics>

</zs:explainResponse>

Configure Rosetta Parameters - Explain File

Use the following procedure to configure the dbInfo parameter in the explain file:
With the File Group field set to External Interface and the Sub-Group set to SRU/SRW, click Edit for the file named explain.properties and enter your SRU (Rosetta) details for dbInfo={DPS SRU Database}
For example,:

dbInfo=DPS SRU Database

Adaptation Needed in the SRU Server

Rosetta should store the CMS ID and metadata in the repository and link it to an IE. In order for Rosetta to perform the link, the following metadata should be returned as part of the response:

<identifier>{DPS}:{IP}:{NAME}:{CMS-ID}</identifier>

where:

- {DPS} = constant text
- {IP} = the IP of the server
- {NAME} = the name of the repository
- {CMD-ID} = the CMS ID

For example:

<identifier>DPS:10.100.2.36:qa654mkdb:9611</identifier>

Note: The response should be in DC format only.
The query:
http://10.100.2.36:17191/voyager?version=1.1&operation=searchRetrieve&query=title=boy&maximumRecords=10&recordSchema=dps

General Parameters

This page was not added to the PDF due to the following tag(s): article:topic-guide

Updating the Rosetta System

This page was not added to the PDF due to the following tag(s): article:topic-guide

Working with the Rosetta System

This page was not added to the PDF due to the following tag(s): article:topic-guide

Working with the Rosetta Database

This page was not added to the PDF due to the following tag(s): article:topic-guide

Backing Up and Recovering Data

This page was not added to the PDF due to the following tag(s): article:topic-guide