

RSD GLASS

3.4.4

Multi-Tenant Edition

Operations Guide

English

## **Preface**

RSD GLASS® is a software package property of RSD - Geneva, Switzerland that cannot be used without license.

RSD reserves the right to make any modifications to this product and to the corresponding documentation without prior notice or advice.

## **Trademarks and abbreviations**

All brand and product names quoted in this publication are trademarks or registered trademarks of their respective holders.

**Manual: RSD GLASS® Operations Guide version 3.4.4**

Copyright © RSD, May 2015. All rights reserved.

For all countries, copies or abstracts of this documentation cannot be made without written approval of RSD.

# Contents

Introduction	5
RSD GLASS® Architecture	6
RSD GLASS® Components	6
RSD GLASS® applications and protocols	7
RSD GLASS® servers	10
RSD GLASS® configuration	11
Configuration files	11
Configuring the interface via GLASS Administrator GUI	11
Generalities	11
Navigation	12
Logging out	12
Configuration options	12
General settings	12
Email settings	14
Theme customization	14
Connecting RSD GLASS® to RSD Content Repositories	14
RSD Admin Center®	15
Home screen	15
Managed applications screen	15
Role management screen	16
Subject associations screen	17
Integrating the platform	18
Introduction	18
Define scenario layout and available data	19
Create or check content repository definitions	19
Create classification rules	20
Create mapping to RSD GLASS	21
Simulate a bulk import process	23
Run the bulk import	23
RSD GLASS® Operations	24
Starting and stopping RSD GLASS®	24
Backing up and restoring RSD GLASS®	24
RSD GLASS® batch process: Bulk Import	25
Bulk Import architecture	25
Generate an XML import file	26
Run RSDBatchTools Bulk Import process	26
Check results	26
Prerequisites	26
Installation steps	26
Return codes	26
Bulk Import Usage	27
Examples	27
XML output file	27
Structure	28
Example	28

RSD GLASS® batch process: Bulk Indexing	29
Introduction	29
Prerequisites	29
Installation steps	29
Return codes	29
Bulk Indexing Usage	30
Examples	30
Configuring Bulk Indexing and Bulk Import processes	30
Batch administration	33
RSD GLASS® logging	33
Application security configuration	33
RSD GLASS Policy Manager:	33
RSD GLASS Governance Manager/Client:	35
RSD GLASS Physical_Records:	37
ACL evaluation details	38
Governance Manager subject attributes	39
Securing RSD GLASS® communications	40
Introduction	40
Communications in RSD GLASS®	40
Schema	40
Matrix	41
Secure AMF (BlazeDS) communication	41
Securing the HTTP Server with HTTPS	42
Apache HTTP Server SSL installation	42
Enabling SSL on Apache Tomcat	43
Configuration Examples	43
Secure Database communication	43
Client Configuration	43
Server Configuration	44
SSL Replication Configuration	44
Secure web services communication	45
Secure LDAP/SAML communication	45
LDAP	45
Server Configuration	45
Client Configuration	46
SAML	46
Secure SOLR communication	47
Secure Batch communication	47
Secure Content Repository communication	48
Appendices	49
Appendix 1: sample log4j configuration files	49
Appendix 2 : HTTP Server configuration examples	50
Apache HTTP Server in front of Tomcat	50

# Introduction

This manual describes the task of operating a RSD GLASS® system. It is written for operators, system programmers, and administrators who are responsible for application configuration, operation, and recovery procedures in their installation.

Operations refers to the day-to-day activities involved in keeping the applications running smoothly, and recovery refers to the operations required to bring a failed system back online.

Technicians operating a RSD GLASS® system are not supposed to have an in-depth knowledge of the product beyond a limited number of routine operation tasks. In order to obtain additional information, if need be, the other RSD GLASS® manuals should be consulted:

- installation procedures are described in the **RSD GLASS® Installation Guide**;
- multi-tenant-specific installation handling is covered by the **RSD GLASS SaaS Guide**,
- end-user's handling and administration of the product and its components are described in the dedicated **User Guides** for RSD GLASS® Policy Manager, Governance Manager, RSD GLASS® Client.

# RSD GLASS® Architecture

## RSD GLASS® Components

RSD GLASS® **Policy Manager** is one of the main modules of RSD GLASS®. It is intended to be a corporate tool for creating, maintaining, approving and publishing Information Governance policies, as well as deploying those policies in the Business Units. Policy Manager is the main working tool for the enterprise Record Manager who is responsible for the design, implementation and administration of Record management within the organization.

RSD GLASS® **Governance Manager** is another one of the main modules of RSD GLASS®. It enables the enterprise architecture team to enforce the corporate policies across geographies and jurisdictions, platforms and applications, repositories and data warehouses. Once the Master Classification has been defined and deployed in the Business Units (BU's) using RSD GLASS® Policy Manager, File Plans can be built through RSD GLASS® Governance Manager. Governance Manager is primarily used by the Business Unit Records Administrator. It also offers content access.

While **Policy Manager** *centrally defines* Information Governance policies, **Governance Manager** *locally implements* the resulting Master Classification in the Business Units.

RSD GLASS® **Client** is one of the modules of RSD GLASS®. It is the corporate tool that provides authorized users with transparent role-based access to Records contained in the RSD GLASS® environment.

RSD GLASS® **Governance Services** provides Information Governance connectors to enterprise Content Repositories.

RSD **Admin Center®** is a technical central administration tool for configuring and monitoring RSD GLASS® applications/services and Resources through a single interface to manage services and modules.

Other non-RSD GLASS® components of interest are these:

- Lucene **Solr**: a full text search engine, used by RSD GLASS® content search facilities;
- **Content Repositories** are the external sources of Records for RSD GLASS® (RSD GLASS Repository may be one of them);
- **LDAP** Directory services are required for authentication and role-based access control (LDAP is the only authentication protocol that RSD GLASS® accepts);
- **SAML/Oauth** Identity Providers (OpenAM, ADFS, Shibboleth...) provide an alternative to LDAP;
- RSD **Activisor®** is an optional tool to aggregate audit trails issued from all Audit server applications running on all platforms.

## RSD GLASS® applications and protocols

RSD GLASS® consists of 3 standalone applications, the first 2 of which include their own DB schema:

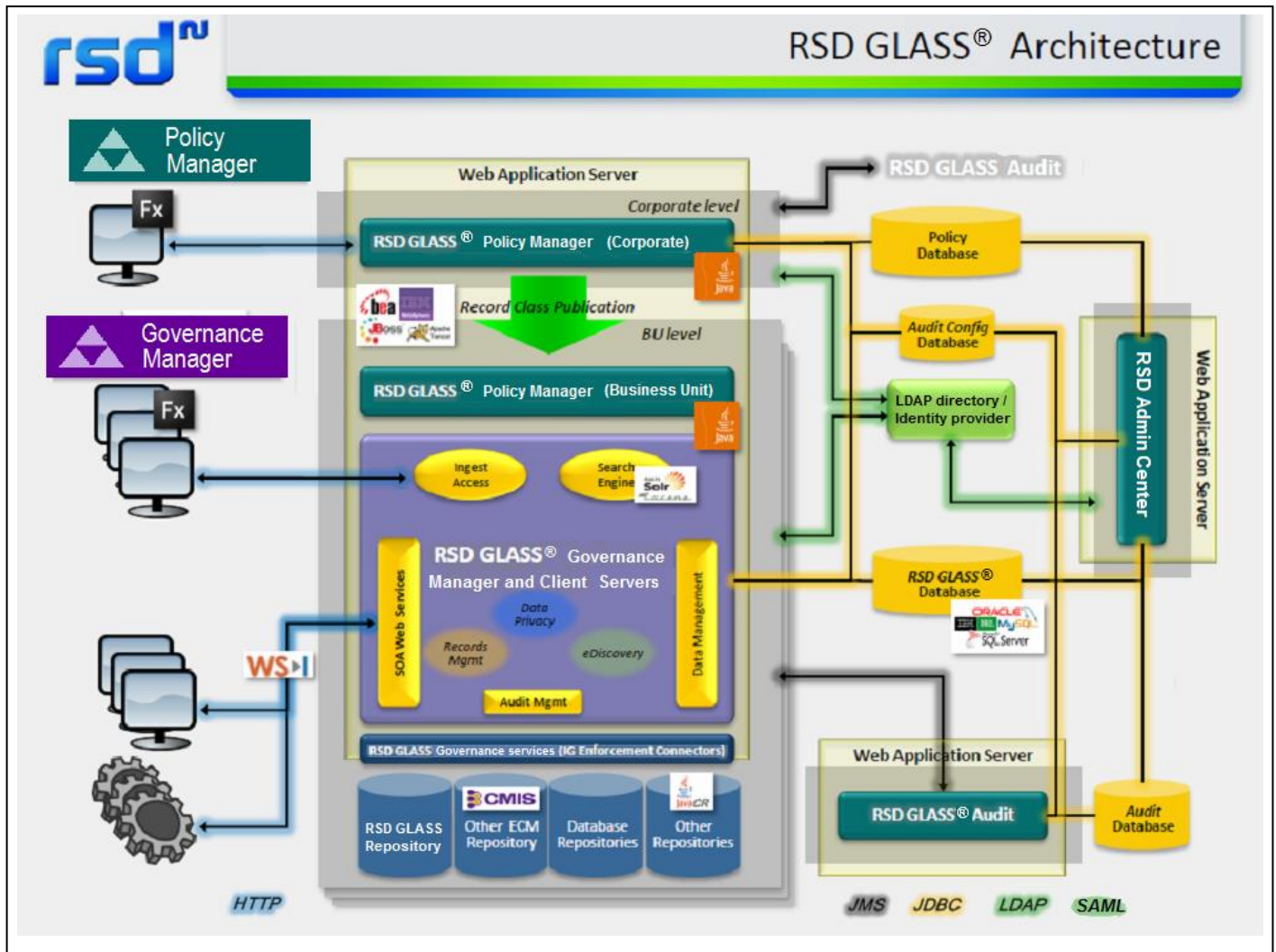
application	components	deployed using
RSD GLASS® Policy Manager	Policy Manager	RSDGlassPolicyManager.war
RSD GLASS®	Governance Manager, Glass Client, Governance Services	RSDGlass.war
RSD Admin Center®		RSDAdminCenter.war  (see manual "RSD Admin Center - Technical Reference")

The following table lists the interfaces and protocols used by the different RSD GLASS® components.

FROM:	TO:	TYPE:	PROTOCOL:
<b>Policy Manager</b>	Policy Manager Database	SQL requests	JDBC
Policy Manager	AuditServer	JMS Communication	JMS
LDAP	Policy Manager	LDAP requests	LDAP
<b>Governance Manager</b>	RSD GLASS® Database	SQL requests	JDBC
Governance Manager	Policy Manager Database ( read only)	SQL requests	JDBC
Governance Manager	RSD GLASS® Repository	API requests	RSD API
Governance Manager	AuditServer Governance Manager	JMS Communication	JMS
LDAP	Governance Manager	LDAP requests	LDAP
Governance Manager	Lucene Solr	HTTP requests	HTTP
RSD GLASS Admin Center	Policy Manager, Governance Manager, Physical Records	HTTP requests	HTTP



The architecture of RSD GLASS® is represented below, as well as the protocols involved between the different elements.



## RSD GLASS® servers

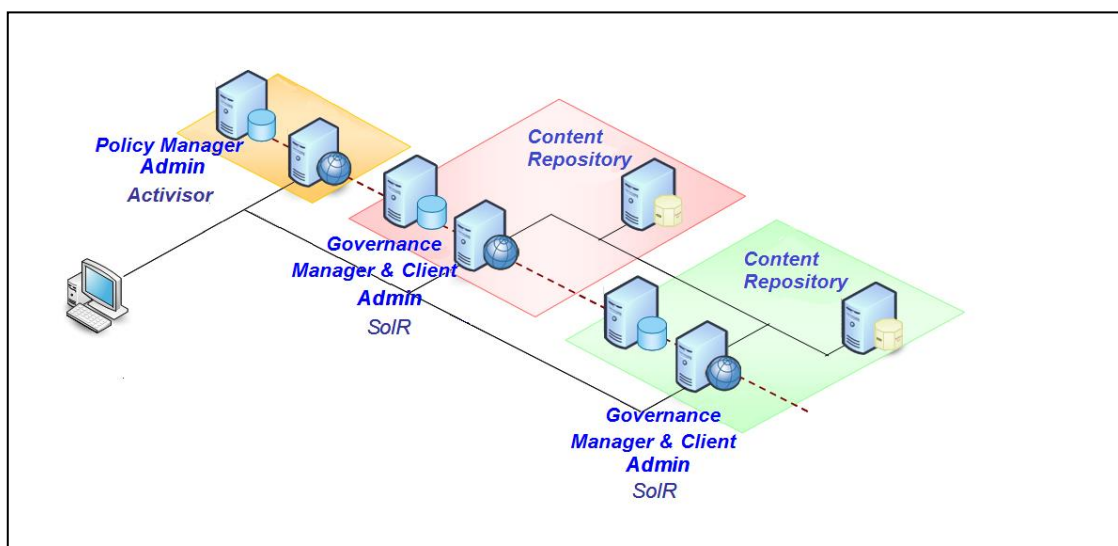
One RSD GLASS® **Policy Manager** server is required, and as many **RSD GLASS® servers** may be installed as necessitated by your environment, the number and types of Content Repositories, the localization of your enterprise Business Units, your network configuration and performance, etc.

RSD GLASS® can easily be scaled-out horizontally to support a large number of users or a large number of requests. Many instances of RSD GLASS® servers can be started, benefiting of the JEE framework scalability.

One straightforward configuration would be to have one RSD GLASS® Policy Manager central server and as many RSD GLASS® servers as there are Business Units. All servers would run the Administration and Audit applications; **Solr** would run on the RSD GLASS® servers and **RSD Activisor** on the Policy Manager server.

Note that Solr and RSD Activisor may also be installed on separate dedicated servers, for performance reasons and because these are not RSD GLASS®-dedicated products; both have their own proprietary databases and their file systems.

A configuration with one Policy Manager server and 2 RSD GLASS® servers is hereunder represented.



# RSD GLASS® configuration

## Configuration files

RSD GLASS® configuration files are located in the orm configuration directory. This directory is defined at installation time by the **glass.config** variable.

RSD GLASS® needs a specific license code to run properly. License codes are delivered in a "GLASS.lic" file to be deployed in the RSD GLASS® configuration directory.

The Administration server parameters (port, server) are stored in a dedicated configuration file, see manual "RSD Admin Center - Technical Reference".

**Log4j** configuration files (log4j.xml) are defined by default in RSD GLASS®, Policy Manager and Admin. The alternative would be to define a central log4j configuration file in the Web application server (~properties directory for Websphere). This log4j configuration file would apply to all applications running under the same application server. An example can be found in Appendix 1.

A **batch.properties** configuration file is required for Bulk Import (described in the related section).

## Configuring the interface via GLASS Administrator GUI

### Generalities

The Administrator Interface allows you to set defaults for all GLASS GUI users. It displays screens to modify settings. These application settings are saved into the GLASS database.

On an individual level, each GLASS GUI user is able to customize some of the application settings. These personal settings are saved into the GLASS database in order that users can log on to the application from any computer and work in their preferred environment.

To access the Administrator Interface:

- execute the following URL in your browser.

***http://address:port/application\_context?application=admin#***

Where:

- *address* is the Web Application Server address.
- *port* is the Web Application Server port number.
- The *address* and *port* must be separated by a colon and contain no spaces.
- *application\_context* is the path under which the GLASS Interface is installed. For example:  
RSDGlassPolicyManager/RSDGlassPolicyManagerGUI/RSDGlassPolicyManagerGUI.html
- There are no spaces in the internet address line.
- The login panel for the application is displayed.  
Initially, no password is defined. Therefore, choose the **Login** button.
- The Administrator interface is displayed.

## Navigation

The sub-branches of the Site contents tree correspond to Module entries.

### Navigation methods

- Click on a name in the Site contents tree. The Attributes panels available at that level are displayed on the right.
- Click on ► next to an element of the Site contents tree to display the Module it contains.
- Click on ▼ next to an element of the Site contents tree to hide the Module it contains.
- Click on an Attribute panel on the right to display the elements it contains.

## Logging out

Selecting Logout from the Menu bar logs the current user out of the application and returns to the login screen.

## Configuration options

When at the initial Administrator interface screen, the following configuration functionality is available.

These options facilitate the saving (export) of all Administrator interface settings in an XML formatted configuration file and the use of these settings (import) at another site or when a new version of the GLASS GUI component is installed.

When changes are made to the configuration, you must press the Save button to save them.

### Import

From the File button select the Import option to import the specified configuration file. Choose the path and file name. Click the Import button and press the Save button to apply the settings.

### Export

From the File button select the Export option to export the Administrator interface settings in an XML formatted file. Default name: application\_settings.xml

## General settings

### User settings

These parameters permit the saving of user settings (customized list displays, format) in order to restore the settings each time the user logs on. If the functionality is activated, the user's personal settings are saved into the database.

To activate this functionality:

- Tick the 'Enable save and restore of user settings' box.
- The 'User settings path' field may be ignored.

**Custom files path**

This field may be ignored.

**Temporary files path**

Indicates the location of a directory where temporary files should be stored.

For Unix platforms, it is strongly recommended that this attribute be set. For Windows platforms, this attribute can be set, but it is optional.

Normally, files in this directory are automatically deleted upon successful completion of the functionality requested.

- In the field 'Temporary files path' indicate the full path. This must be a directory, which exists, on the same machine where the Web application server is running. Default: directory, on the system, where temporary files are normally stored.

**Date format**

You can choose one of two different date formats for display of date information. Click on either:

US (mm/dd/yyyy)	Dates are displayed showing the month, the day, and the year in four digits. Standard format for dates in the United States.
EU (dd/mm/yyyy)	Dates are displayed showing the day, the month, and the year in four digits. Standard format for dates outside the United States.

**Display list on full page by default**

By selecting this option it is possible to specify that the Report list is displayed on the full page.

When using this option, the 'Enable save and restore of user settings' box must not be ticked.

Click the 'Validate' button. You may then press the Save button to save the settings.

## Email settings

These parameters set the defaults for e-mailing a log.

- For the 'Email Server Address', enter the address for the machine on which the mail server resides.
- For the 'Email Server Port', enter the port number where the mail server can be accessed (for example: 25, well-known port for SMTP).
- For the 'From address', enter a valid email address.
- For the 'Display Name', enter the sender name that you wish to be displayed.
- For 'Character Encoding' enter the required character encoding option. Default: ISO-8859-1

Click the 'Validate' button. You may then press the Save button to save the settings.

## Theme customization

The "**Theme customization**" tab lets you change the logo that is displayed on top of the screens of the interface. The logo may be any 20 pixels high image, in any of the usual types (.png, .jpg, .gif, .tif, .bmp).

## Connecting RSD GLASS® to RSD Content Repositories

This paragraph gives some guidelines in the case where one of your content repositories is RSD EOS or RSD GLASS® Repository.

To connect the EOS and RSD GLASS® Repository repositories to RSD GLASS® and have RSD GLASS® manage those repositories, some actions must be carried out directly in EOS and RSD GLASS® Repository:

- **EOS for z/OS** prerequisites are described:
  - in the manual "RSD EOS for z/OS Management Guide and Technical Reference", section C ("Report processing logic"), chapter "GLASS archiving";
  - in the specific "EOS-GLASS interface" manual;
- **RSD GLASS® Repository for z/OS** prerequisites are described in the manual "RSD GLASS Repository for z/OS Management Guide and Technical Reference", section C ("Document processing logic"), chapter "ILD-RSD GLASS interaction";
- **RSD GLASS® Repository for Unix or Windows** prerequisites are described in the manual "RSD GLASS Repository for Unix/Windows Management Guide and Technical Reference", section 3 (Implementation), "RSD GLASS® Repository as an RSD GLASS® repository".

# RSD Admin Center®

RSD Admin Center® is the central administration tool to configure and monitor RSD GLASS® applications, services and resources. You should refer to the "RSD Admin Center - Technical Reference" manual to review the installation process as a whole.

RSD Admin Center® can be accessed using an URL in the form of:

**http://servername:port/RSDAdminCenter/**

**Note:** for configuring a multi-tenancy environment, you should refer to the "RSD GLASS® SaaS Guide".

## Home screen

Once a valid User login and password have been entered, the "Home" screen is displayed and the Managed Applications are listed.

The "Add application" button enables you to define a new managed application: you must enter:

- a name for the application;
- the type of server to be managed;
- the URL to the context root of the application
- the URL of the SAML ECP (enhanced client or proxy) profile, if applicable.
- OAuth check box to be selected if the administered application requires OAuth2 authentication.

A remote admin user + password must also be defined.

Click on the managed application of your choice to further administrate it, through the screens that follow.

Note: the version of RSD Admin Center® that you use should match the version of the managed applications. No upward compatibility is guaranteed when the versions do not match.

## Managed applications screen

After a managed application has been chosen, the home screen for this application is displayed.

For example, with a Policy Manager application:

- a "Summary" frame displays the characteristics of the application: name, type, URL, version;
- the "Role management" frame displays the roles, and if allowed role schemas, that you may configure for this application;
- the "Security" frame can be used to set the general security parameters and the subject associations.

With a Governance Manager application:

- a "Summary" frame displays the characteristics of the application;
- the "Security" frame can be used to set the general security parameters and the subject associations;

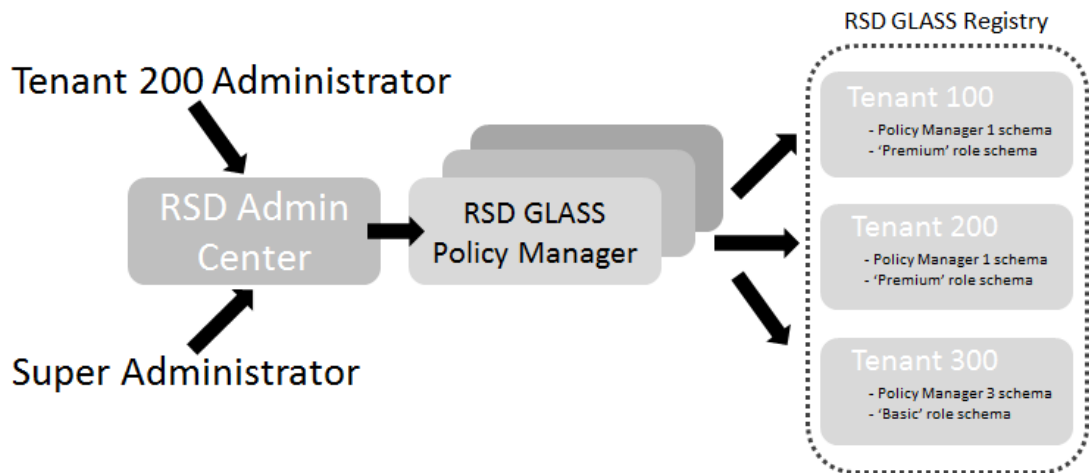
- the "Security" frame can be used to set the general security parameters and the subject associations.
- the "Miscellaneous" frame can be used to set Content search parameters and Converters.

## Role management screen

Features available in the Role Management screen are dependent on if you are logged into RSD Admin Center as a user with Super Administrator rights.

Super Administrator users can manage all configurations whereas a standard user can only configure a specific role schema.

The following example illustrates how an Administrator of a tenant called Tenant 200 accesses and configures only the 'Premium' role schema that has been defined in the RSD GLASS Registry for his tenant:



The Super Administrator user needs to initially select the 'Configuration' he wishes to manage from the dropdown box. This is a list of all the role schemas that have been associated to a tenant in the RSD GLASS Registry:

Note that the name of each configuration is automatically generated using the description given when the schemas is created followed by the name of the schema given in the Registry.

### Role Configuration

The "Configure roles" option in the Role management frame allows you to configure Roles in Policy Manager.

Type a search term in the top field and click on the Search button. Based on that search term, the corresponding roles are listed for the managed application.

There may be the possibility to add and edit roles depending on the authentication mode and if roles are retrieved from LDAP or not. If the Add button is available add any missing roles.

The bottom frame displays the related authorized applications. To allow login to one of those applications, you tick the box in the list.

The roles defined in Policy Manager are used for all applications listed.



For each application that you select in the bottom frame, the resources are displayed in the frame on the right side. You can tick the relevant options to define the authority that you grant on the resource:

- Full control;
- Traverse;
- Create;
- Read;
- Update;
- Delete.

Before leaving, you have to save the updated configuration using the Save button.

Further information on how configure access to the RSD GLASS® applications, see ["Application security configuration"](#) on page 33.

## Subject associations screen

The "Configure subject associations" option in the Security frame allows you to configure associations between Group, Users and Roles for Policy Manager, Governance Manager, Physical Records. This option is only relevant in the case where LDAP is used and user and role associations are performed in RSD GLASS.

Select the desired tab (Groups, Users or Roles), and type a search term in the top field and click on the Search button. Based on that search term, the corresponding Groups, Users or Roles are listed for the managed application.

Depending on the authentication mode and if the objects and their associations come from LDAP or not, you may be able to associate one object to another or not.

For each Group, the associated Users and Roles are displayed. New Roles may be assigned to (or removed from) a Group by dragging and dropping them into the frame on the right, or using the left/right arrows.

For each User, the associated User attributes, Groups and Roles are displayed. New user attributes (key/value) and new Roles may be assigned to a User or removed from a User.

For each Role, the associated User and Groups are displayed. Similarly, you can assign a Role to new Users or new Groups or remove a Role from a User or a Group.

Before leaving, you have to save the parameters using the Save button.

# Integrating the platform

## Introduction

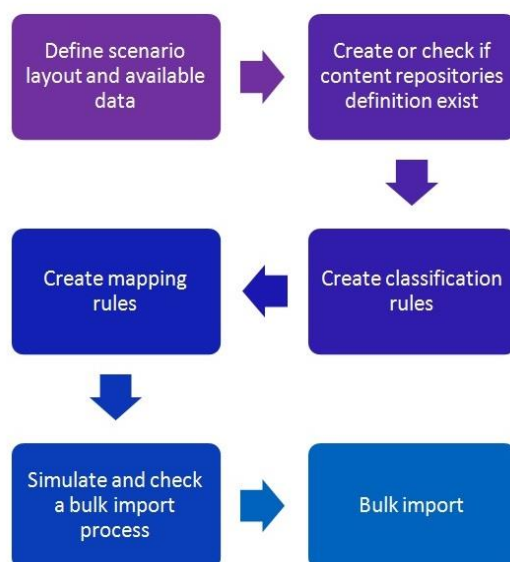
This chapter gives guidance for integrating, configuring and setting up the platform once deployed.

RSD GLASS Governance Services provides connectors for each type of repository (as an alternative, you can use the RSD GLASS® Integration SDK toolkit to create applications and application repository connectors).

Installing a new connector (the driver in the Repository/Virtual Repository section in Governance Manager) provides interoperability between RSD GLASS® and the remote Enterprise Content Management or Archiving system. The connector enables the repository to implement RSD GLASS® services for Information Governance enforcement. Depending on the repository type, you should refer to the RSD GLASS® Governance Services **"Management Guide and Technical Reference"** dealing with the related connector.

To integrate the solution, you need to define all the content schemas, metadata mapping and classification rules. Once defined, you can then start testing your XML import files. You will need afterwards to monitor the different batches and correct errors should a batch be rejected.

The integration methodology is synthesized in the figure below.



Topics covered:

- [Define scenario layout and available data](#) page 19
- [Create or check content repository definitions](#) page 19
- [Create classification rules](#) page 20
- [Create mapping to RSD GLASS](#) page 21
- [Simulate a bulk import process](#) page 23
- [Run the bulk import](#) page 23

## Define scenario layout and available data

From a functional standpoint, you should analyze and identify the content in the repository, identify the documents to process and create a selection of candidates. Those candidates will be exported to RSD GLASS® to be referenced and managed. The target node is defined through classification rules, and the mapping will apply to create the node to which the content will be associated. The correct Record class may be assigned depending on relevant attributes or values from the extracted content. An XML file should be retrieved from the repository to be used to reference content into RSD GLASS.

## Create or check content repository definitions

In RSD GLASS® Governance Manager, you will use the **Settings** menu, **Repositories** option.

### *1. Content repository definition*

The content repository definition enables you to setup a connector for communication between RSD GLASS® and the remote system. In the "connection" tab, you define the "Chroot file path" (case sensitive) as the path where you have data to analyze and import into RSD GLASS. Check the "Capabilities" tab (by default you may set all capabilities checked).

### *2. Virtual repository*

Create a virtual repository. You may set mandatory fields with default values; integrity check, compliance level and encryption may be initially disabled.

### *3. Associate an XSD schema to the Content Repository*

You have to download an XSD schema which is compliant with the current version of XML generated by the content repository.

In RSD GLASS® Governance Manager, you will use the **Integration Tools** menu, **Schema Manager** option to create the content source. In the content source creation form, you associate the content repository definition with the content source.

You also create a new schema instance for the content source, in order to enable the content source to be attached to one or more XSD schema definitions. The schema may be uploaded using the "from file" button or pasted in the edit box using the "Edit XSD" button.

The "document path" indicates at which level you will find the definition of the documents to import: this defines the loop scope for processing multiple documents in a single XML source. It may be selected by double clicking on one of the metadata displayed from the XSD schema.

## Create classification rules

Classification rules enable definition of different scenarios and target nodes for the bulk import process.

In RSD GLASS® Governance Manager, you will use the **Integration Tools** menu, **Classification Rules** option to create the rules.

To create a new rule, you first select a content source to attach the rule(s) you need, then click **Add new rule** to create a new entry. A rule is associated with **conditions** and with **results** that are expected when a condition is fulfilled.

You must determine what information to use from the XML. For example what information to find in the XML source to identify the type of document you are dealing with.

You must also determine which condition you want to test. Conditions are tests of XML node properties in order to import that particular node to a given location in a File plan. You define the conditions logic to test the value and decide if the rule applies or not.

Example: based on the following XML structure:

```
<Batch>
  <Documents>
    <Document DocID="001.001.001" DocType="File" MimeType="">
      <Tags>
        <Tag TagName="Name" TagDataType="Text" TagValue="2009-01-Pay Slip.pdf"/>
        ...
      </Tags>
    </Document>
  </Documents>
</Batch>
```

the condition might be:

- Criteria: Tags/Tag[@TagName='Name']/@TagValue
- Operator: Contains
- Value: Pay Slip

You must finally define a result for the rule, i.e. which result is expected if conditions are fulfilled. A result for a rule is a particular node you will use as reference to do your import in a File plan: it may be a Record class, or a specific folder. That particular node will be the **parent of the item you will further reference** in the File plan.

Example:

/0000000001/COR/300/50

### Note:

XSLT criteria and values support XSLT functions. You should learn about some basic **XSLT** (*Extensible Stylesheet Language Transformations*) functions and mechanisms before using mapping tools in GLASS. The same goes for **XPath** (*XML Path Language*) functions and concepts, because criteria are mostly based on XPath referencing.

## Create mapping to RSD GLASS

A mapping in class is a set of conditions and XSLT functions used to extract information from an XML source and generate a target XML for an import on a target node of a File plan

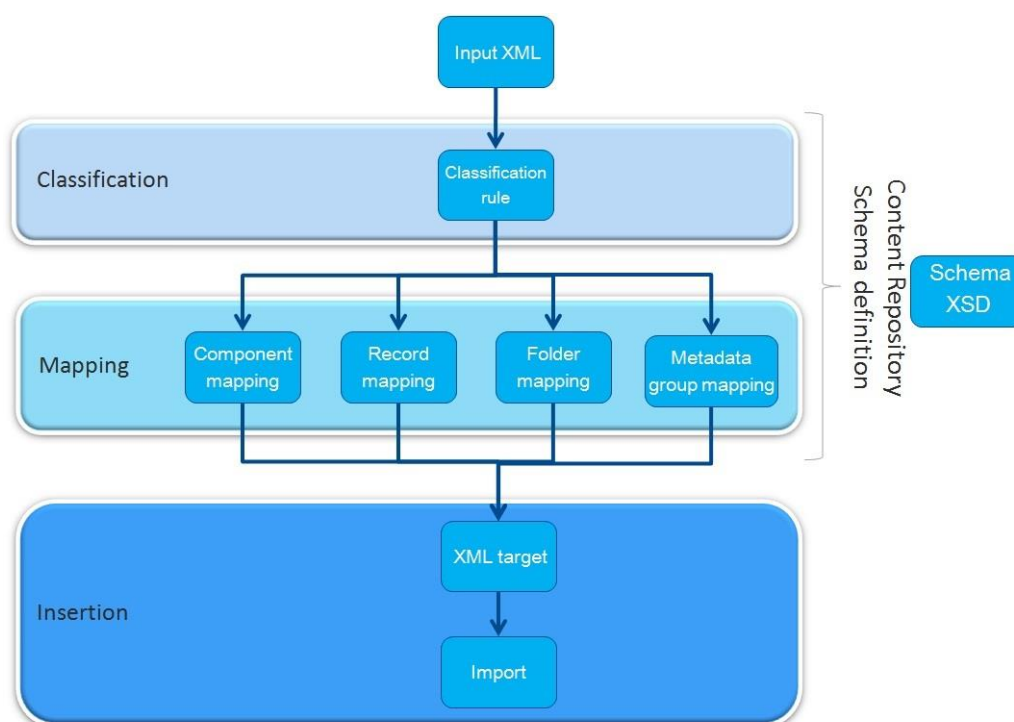
A mapping is associated to a schema. It splits relationships definitions around:

- Component level
- Record level
- Folder level
- Additional metadata group related to target parent record class

The mapping is conducted as follows:

- the input is an XML source format based on a specific content repository
- the content repository provides access to a set of mapping definitions
- these mapping definitions are used to generate an XSLT transformation sheet
- the transformation sheet and the source are processed by a transformation processor
- the resulting XML is then used to carry out the import to the target node selected by the classification rules into the RSD GLASS File plan

The import process is synthesized in the figure below.



In RSD GLASS® Governance Manager, you will use the **Integration Tools** menu, **Mapping to RSD GLASS** option to create the mapping. The screen consists of four different parts:

1. Source selector panel: you select the content repository source you want to create mappings for.

2. RSD GLASS schema structure (destination schema): display details of all node levels managed by RSD GLASS where a mapping has to be defined. Each level is attached to a particular Content Repository.

3. Mapping rules detail and edit form: displays mapping rules attached to a selected field for a given node level. For each mapping rule, the list shows a set of conditions to fulfill in order to apply the rule, and a result associated with the rule. The different rules are:

- component-level rules
- record-level rules
- folder-level rules
- metadata group-level rules (if a metadata is "distinctive", its value is used as folder name when the node is created). Metadata groups may be attached at record level or folder level.

4. Source structure schema: displays a graphical representation of a source schema, thus enabling you to build a mapping with different fields of a node level in RSD GLASS. The list matches the different schemas you have attached so far to a content repository source.

Example (component-level rule):

If the value extracted from the tag exists, it is reused as the name for the title of the component. If the value does not exist or is empty, then a fixed name «**No File name**» is used for the item. The conditions are as follows:

ID	Criteria	Operator	Value	Result
1	Tags/Tag[@TagName='Name']/@TagValue	!=	''	Tags/Tag[@TagName='Name']/@TagValue
2	Tags/Tag[@TagName='Name']/@TagValue	none	none	«No File Name»

If the "Default and last Condition result" box is ticked, the rule will be applied if no other condition has been fulfilled. If the "Static result" box is ticked, the content of the field is considered to be a string and is not evaluated by the XSLT compiler.

## Simulate a bulk import process

RSD GLASS Governance Manager includes a tool to simulate and validate a bulk import process with the different rules created so far. This tool uses a sample source file and then applies both classification rules and mapping to simulate a bulk import. Nodes created are then viewable inside a dedicated form to check the result. Different XML content at an intermediate level can be accessed and verified during the simulated bulk import process:

- XML source Input
- classification rules result
- mapping rules result
- import data view

In RSD GLASS® Governance Manager, you use the **Integration Tools** menu, **Import simulation** option to simulate the bulk import process:

- first select a content source
- click «**Upload**» to select a source sample file generated on the repository platform to do the simulation
- all nodes are then processed in the simulation and after a few seconds (depending on the size of the sample file), the different result windows are populated
- a validated step is marked with a green check, a step that failed is marked with a warning icon (in that case, access the panel of the step to obtain more information on the error)
- a correct simulation is characterized by nodes with OK status and all steps of a node import with OK status. Make sure the simulation is OK before going any further.

## Run the bulk import

See relevant section below ("RSD GLASS® batch process: Bulk Import").

# RSD GLASS® Operations

## Starting and stopping RSD GLASS®

The order in which RSD GLASS® components or applications should be started is as follows:

- Database systems
- Content Repositories (such as RSD GLASS® Repository)
- Solr
- RSD GLASS® and RSD GLASS® Policy Manager (in no specific order)

Conversely, the order in which GLASS components or applications should be stopped is as follows:

- RSD GLASS® and RSD GLASS® Policy Manager (in no specific order)
- Solr
- Content Repositories (such as RSD GLASS® Repository)
- Database systems

You may use whatever tool is at your disposal to carry on the operation (e.g. the Websphere administrative console if Websphere is your web application server).

Some administrative modifications on RSD GLASS® require that RSD GLASS® be restarted to take into account the new objects or the new functions, for example:

- when a new File Plan is defined;
- when a physical or virtual Content Repository is created;
- when the audit has been activated;
- when configuration files (such as the log4j configuration files) are modified;
- when new license keys must be installed.

## Backing up and restoring RSD GLASS®

It is recommended that the application servers be stopped before backing up RSD GLASS®.

The order in which RSD GLASS® elements should be backed up is as follows:

- Content Repositories (such as RSD GLASS® Repository) ; RSD GLASS® databases (Policy Manager, RSD GLASS®, Audit) ; Solr directory
- RSD GLASS® Policy Manager

The order in which RSD GLASS® elements should be restored, if need be, is the reverse order.

You may also want to archive the .csv files generated by the Audit servers.



## RSD GLASS® batch process: Bulk Import

The main batch process in RSD GLASS® is the **Bulk Import** of document information in .xml format from the content repositories into the RSD GLASS® database.

Bulk Import is the only available process to import a large volume of records into RSD GLASS®.

Bulk Import is a process to import references to external records/content into RSD GLASS®. It creates Records into RSD GLASS® reflecting content stored in an external Content Repository and it uploads descriptive information related to these Records.

The following components are used by the Bulk Import:

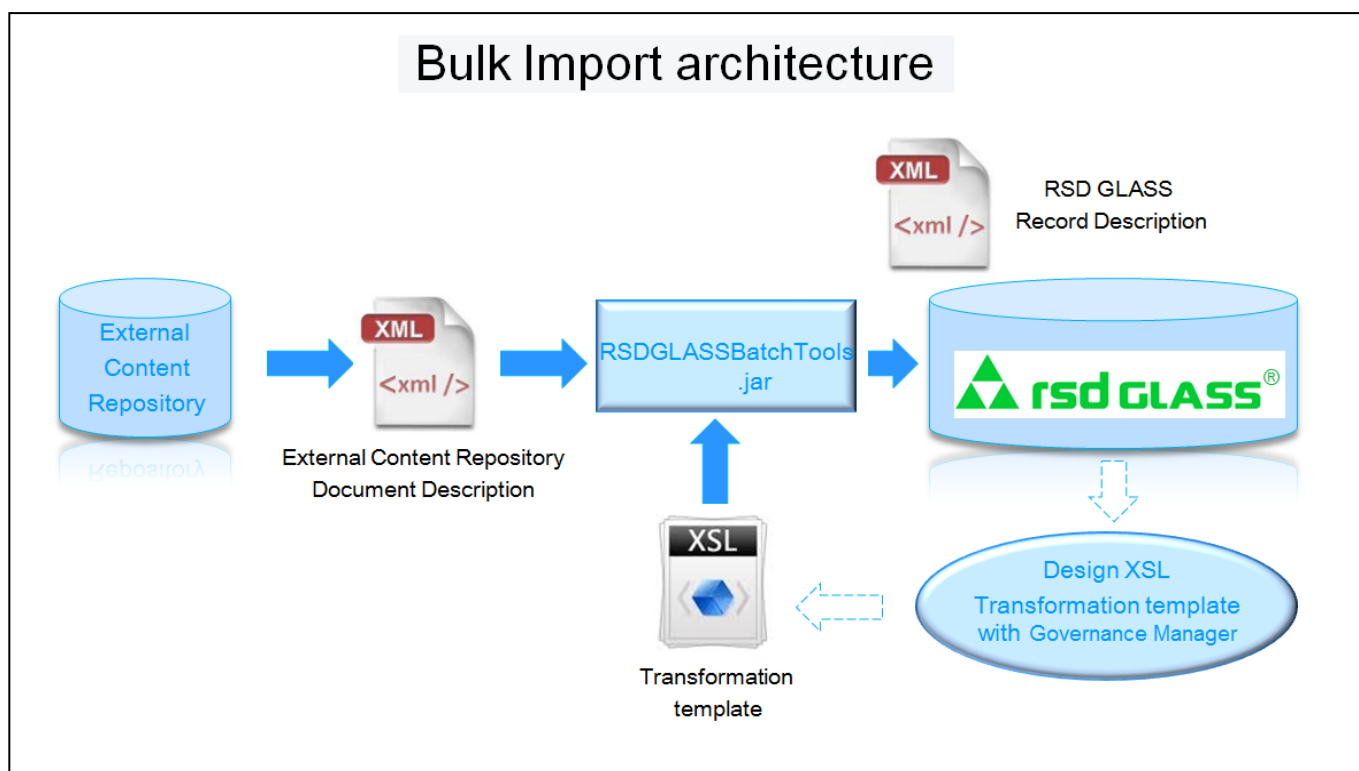
- the Governance Manager application (using Web Services)
- an external Content Repository (such as RSD GLASS® Repository)
- the RSD GLASS® database
- a **batch.properties** configuration file; the path to XML files; the XSL mapping file...

The **batch.properties** configuration file is used to define:

- the JDBC driver to use
- the URL for connecting to the RSD GLASS® database
- the database user and password
- the RSD GLASS® instance user and password
- the RSD GLASS® instance node management URL
- the RSD GLASS® instance mapping URL
- the Oauth properties

### Bulk Import architecture

The Bulk Import process is implemented with RSD GLASS® Governance Manager.



The Bulk Import process consists of the following steps:

### Generate an XML import file

This preparatory step creates a XML description file for a group of document stored in an external content repository. This information file must include a Unique ID to access each document and metadata information related to each document entry.

### Run RSDBatchTools Bulk Import process

You must invoke the "RSDGLASSBatch.jar" application.

This application requires a Batch.properties configuration file to run (see below).

### Check results

Check the result of the Bulk Import process by browsing the File plan.

## Prerequisites

1. Java Runtime Environment (JRE) must be installed. Note that this is included in the Java 2 Platform Standard Edition version 7 necessary for the Application server environment.  
The **Sun JVM** should be preferred. If you want to use instead the **IBM JVM**, then you must download **jaxp-ri-1.4.jar** from the relevant place.
2. When using information stored on **RSD GLASS® Repository Open System servers**:  
"GLASS Rollback Export" functionality must be activated and performed on your RSD GLASS® Repository server installation. This functionality is available as from Version 5.3 of the RSD GLASS® Repository server and is described in the corresponding version of the documentation "*RSD GLASS® Repository for Windows and Unix Servers - Management Guide and Technical Reference*".  
When using information stored on **RSD GLASS® Repository z/OS servers**:  
ILM - RSD GLASS® interaction must be activated. This functionality is available as from Version 5.0 with the RSD GLASS®-related EPF. ILM functionality is described in "*RSD GLASS® Repository for z/OS Server - Management Guide and Technical Reference*" as well as the ILM mini documentation. RSD GLASS® functionality is described in the RSD GLASS® mini documentation.

## Installation steps

1. Copy and uncompress the **RSDGlassBatch.zip** according to your environment in a working directory. This will create a libs directory containing all the necessary librairies, and the jar file to use, called **RSDGlassBatch.jar**.
2. Create the **batch.properties** file in the execution directory in order to define the global parameters of the batch services. See example batch.properties in the section "[Configuring Bulk Indexing and Bulk Import processes](#)" on page 30.
3. Execute the command. The following is an example of the command and it should be modified according to the example given below and using the options detailed in "[Bulk Indexing Usage](#)" on page 30:

```
java -cp RSDGlassBatch.jar com.rsd.glass.batch.bulkimport.BulkImport
```

## Return codes

Return codes from the Bulk Import process are as follows:

- 0: Bulk Import successfully completed;

- 1: Bulk Import failed, see the processing log.
- 2: WEB SERVICE ERROR
- 3: CONTENT STORAGE ERROR
- 4: METADATA ERROR
- 5: CONTENT REPOSITORY ERROR (~)
- 6: JOB ERROR (on restart)

In case of abnormal end, the process must be restarted from the beginning.

## Bulk Import Usage

-t,--temp-folder	The path of the folder that will contain temporary files for indexing [Mandatory]
-c,--glass-fqc	The full qualified code from which to index
-T,--tenant-id	When OAUTH is the authenticationMode, TeanantID must be set
-j,--job-id	Used to identify a job: could be any sequence of characters
-i,--input-path	The path of the input xml export file from the external repository.
-h,--chunk-size	Number of items to process at once while reading input.
-g, --ingestion-size	Number of Records to be included in an ingestion request. The number specified will depend on the infrastructure used.
-s, --source-label	Optional. The content source (identified by its label) to be used by the Import module.
-S, --source-crlid	Optional. The content source (identified by its contentRepositoryId) to be used by the Import module.
-r, --errorRecovery	Optional. Activation of recovery mode.
-f, --force-restart	Optional. Force to restart a failed batch whatever its state.
-d, --date-of-the-day	Optional. Use the current day date for ingestion instead of the date specified in the mapping file.
-z, --date-of-the-day	Optional. Delete given job (with all its associated nodes).
-Djava.endorsed.dirs=	Optional. The path of the folder that will contain only the jaxp-ri-1.4.jar if you use the IBM JVM instead of the Sun JVM.
-Dproperties.path=	Optional. The path of the batch.properties file. This enables you to use the same Bulk Import implementation for two different RSD GLASS Governance Managers.

## Examples

### Bulk Import Usage example

```
# Multi-tenant mode (only TENANT_B is bulk indexed)
java -cp "/etc/lib/mysql-connector-java-5.1.34-bin.jar:RSDGlassBatch.jar"
com.rsd.glass.batch.bulkimport.BulkImport -t temp -T TENANT_B -j 123abc -i file:export1.xml -h100 -g10 -S fld -d
```

## XML output file

The Bulk Import process creates a XML output file that documents which records have been imported, gives execution details, and specifies whether the import was successful or not.

The output file is named **jobid\_results.xml**.

An `errorCode.properties` file is produced by the process, it contains a list of data management error messages and the corresponding codes. For example, if importing a document with an undefined parent, an error code 1030 is returned (`datamgmt.err.NoParent=1030`).

## Structure

<code>&lt;results jobId=" " success=" " failure=" "&gt;</code>	number of imported documents (success=) and not imported (failure=), job identifier
<code>&lt;document&gt;</code>	Start of document information
<code>&lt;docid&gt;&lt;docid/&gt;</code>	docid of imported document
<code>&lt;status&gt; &lt;/status&gt;</code>	value: CREATED or FAILED
<code>&lt;uid&gt; &lt;/uid&gt;</code>	UID of the imported document (if document was successfully imported)
<code>&lt;fullQualifiedcode&gt; &lt;/fullQualifiedcode&gt;</code>	full qualified code of the imported document (if document was successfully imported)
<code>&lt;seq&gt; &lt;/seq&gt;</code>	sequence order in the creation
<code>&lt;errorCode&gt; &lt;/errorCode&gt;</code>	error code if the import fails
<code>&lt;errorLog&gt; &lt;/errorLog&gt;</code>	error log if the import fails
<code>&lt;/document&gt;</code>	End of document information
<code>&lt;document&gt;</code>	
.....	
<code>&lt;/document&gt;</code>	
<code>&lt;/results&gt;</code>	

## Example

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<results failure="5" success="45" jobId="57696">
  <document>
    <docid>015585CA1F37900B11E4A18308</docid>
    <status>CREATED</status>
    <uid>1f09a9e5-2064-1766-8e65-014aba9dc4c7</uid>
    <fullQualifiedcode>/0000000004/0001/0001-000/0001-000-
000/0000000001/</fullQualifiedcode>
    <seq>1</seq>
  </document>
  .....
  <document>
    <docid>015585CA1F37900B11E4A18308002700E</docid>
    <status>FAILED</status>
    <seq>50</seq>
    <errorCode>datamgmt.err.NoParent</errorCode>

    <errorLog>com.rsd.glass.core.datamgmt.facade.exception.DataMgmtExcept
ion: [datamgmt.err.NoParent] Cannot access the parent node. No Node
matches the following criteria([recordClassRef c4272ea1-2064-1766-
8e65-014aba9d3e6c]:nodeType:folder AND
classificationPath:/0000000004/0001/0001-000/0001-000-000* AND
FTGroup.FOLDERNAME:&quot;FOLDER0000100050&quot;)</errorLog>
  </document>
</results>
```

# RSD GLASS® batch process: Bulk Indexing

## Introduction

The Bulk Indexing batch process in RSD GLASS® allows you to re-index the RSD GLASS® database necessary for the full text search.

This process should not be run on a regular basis. It is only needed when changing the Search Engine configuration files (Solr), or when indexes are corrupted.

The following components are used by the Bulk Indexing:

- the RSD GLASS® Governance Manager application (using Web Services)
- the RSD GLASS® database
- the RSD GLASS® indexes (e.g. Solr)
- a batch.properties configuration file

## Prerequisites

Java Runtime Environment (JRE) must be installed. Note that this is included in the Java 2 Platform Standard Edition version 7 necessary for the Application server environment.

The Sun JVM should be preferred. If you want to use instead the IBM JVM, then you must download jaxp-ri-1.4.jar from the relevant place.

## Installation steps

1. Copy and uncompress the **RSDGlassBatch.zip** according to your environment in a working directory. This will create a libs directory containing all the necessary librairies, and the jar file to use, called **RSDGlassBatch.jar**.

2. Create the **batch.properties** file in the execution directory in order to define the global parameters of the batch services. See example batch.properties in the section "[Configuring Bulk Indexing and Bulk Import processes](#)" on page 30.

3. Execute the command. The following is an example of the command and it should be modified according to the example given below and using the options detailed in "[Bulk Indexing Usage](#)" on page 30:

```
java -cp RSDGlassBatch.jar com.rsd.glass.batch.indexing.BulkIndexing
```

## Return codes

Return codes from the Bulk Indexing process are as follows:

- 0: Bulk Indexing successfully completed;
- 1: Bulk Indexing failed, see the processing log.
- 2: WEB SERVICE ERROR
- 3: CONTENT STORAGE ERROR
- 4: METADATA ERROR
- 5: CONTENT REPOSITORY ERROR
- 6: JOB ERROR

In case of abnormal end, the process must be restarted from the beginning.

## Bulk Indexing Usage

-t,--temp-folder	The path of the folder that will contain temporary files for indexing [Mandatory]
-c,--glass-fqc	The full qualified code from which to index
-T,--tenant-id	When OAUTH is the authenticationMode, TeanantID must be set
-d,--content-index	Allow or disallow document content indexing[Mandatory]
-s,--reset-index	Reset index or just update it [Mandatory]
-i,--job-input	CSV file which contains the fqc codes to re-index
-f,--force-restart	Force to restart a failed batch whenever its state
-h,--chunk-size	Number of nodes to process at once while reading input [Mandatory]
-M,--multitenant-mode	Batch executed in a multi-tenant environment
-o,--only-component	May be added in order to index only components. [Optional].

## Examples

### Bulk Index Usage example

```
# Multi-tenant mode (only TENANT_B is bulk indexed)
java -cp "/etc/lib/mysql-connector-java-5.1.25-bin.jar:RSDGlassBatch.jar"
com.rsd.glass.batch.indexing.BulkIndexing -t temp -c /0000000001 -h 1000 -s true -d false -T
TENANT_A
```

## Configuring Bulk Indexing and Bulk Import processes

Configuration for both the Bulk Indexing and Bulk Import processes is done through the **batch.properties** configuration file.

### Common configuration

#### Authentication

The parameter **input.glassInstance.authenticationMode** sets Authentication Mode. Possible values are OAUTH or SIMPLE.

- Related parameters are: **input.glassInstance.username** (User name for authentication to the RSD GLASS application)
- **input.glassInstance.password** (User password for authentication to the RSD GLASS application)

These parameters are mandatory but may be left empty if SIMPLE mode is set.

#### RSD GLASS web service

The BulkIndex batch needs to request the RSD GLASS service. This is configured using the following parameters:

- **input.glassInstance.rootURL** (Webservices URL of the RSD GLASS application)
- **input.glassInstance.samlIdpEcpUrl** (if RSD GLASS is using SAML authentication this should be the IDP's ECP url, otherwise it should be left empty)

## Multi-tenant configuration

### Zookeeper

In multi-tenant mode, the Bulk Indexing batch needs to access Zookeeper to retrieve configuration information. This is configured using the following parameters:

- **input.zk.host** (Zookeeper host url)
- **input.zk.root** (Zookeeper root configuration)

### Datasource

In multi-tenant mode, the datasource configuration depends on the tenant and is transparently retrieved from Zookeeper. The only parameter needed is:

- **input.db.type** (Database type, for example: mysql)

### Example

batch.properties - - Multi-tenant mode

```
#-----  
# RSDGlassBatch.jar configuration file  
#  
# This configuration file is used to define Bulk Import and Bulk ReIndexing behavior.  
#-----  
# Import thread pool Configuration  
#  
# The following parameters are permitted :  
# - bulkImport.threads.maxPoolSize Maximum number of concurrent threads for the bulk import  
bulkImport.threads.maxPoolSize=10  
# Indexing thread pool Configuration  
#  
# The following parameters are permitted :  
# - indexing.threads.maxPoolSize Maximum number of concurrent threads for the bulk indexing  
indexing.threads.maxPoolSize=10  
#-----  
# Multi Tenant configuration  
#-----  
# Database Configuration  
#  
# The following parameters are permitted :  
# - input.db.type Engine type of the batch database  
input.db.type=mysql  
# Zookeeper Configuration  
#  
# The following parameters are permitted :  
# - input.zk.host Zookeeper host url  
# - input.zk.root Zookeeper root configuration  
input.zk.host=myserver:2181  
input.zk.root=/rsd-conf  
# Glass WebServices Configuration  
#  
# The following parameters are permitted :
```

```

# - input.glassInstance.username User name for authentication to the Glass application
# - input.glassInstance.password User password for authentication to the Glass application
# - input.glassInstance.rootURL Webservices URL of the Glass application
# - input.glassInstance.samlIdpEcpUrl if Glass is using SAML authentication this should be the IDP's
ECP url, otherwise it should be left empty.
# -- Example IDP ECP url: http://{host}:{port}/idp/profile/saml2/soap/ecp.
# - input.glassInstance.authenticationMode OAUTH in case of multi tenant configuration
input.glassInstance.username=
input.glassInstance.password=
input.glassInstance.rootURL=http://myserver:8080/RSDGlass
input.glassInstance.samlIdpEcpUrl=
input.glassInstance.authenticationMode=OAUTH
# OAuth2 Configuration
#
# The following parameters are permitted :
# - input.glassInstance.access.oauth2.accessTokenUri url of openam for accessToken
# - input.glassInstance.access.oauth2.clientID clientID of OAuth client agent
# - input.glassInstance.access.oauth2.secretID secretID of the current clientID
# - input.glassInstance.access.oauth2.tenantIDKey tenant id key
input.glassInstance.access.oauth2.accessTokenUri=http://myserver:8080/OpenAM/oauth2/access_token
input.glassInstance.access.oauth2.clientID=GlassRoot
input.glassInstance.access.oauth2.secretID=GlassRoot
input.glassInstance.access.oauth2.tenantIDKey=TenantID

```



## Batch administration

There are two ways of controlling batch jobs:

1. Develop a Java program using Spring Batch JobOperator's class<sup>1</sup>;
2. Use Spring Batch web admin<sup>2</sup>.

In both cases, you will have to configure a "batch.properties" configuration file as mentioned in this guide and Spring Batch ones.

## RSD GLASS® logging

RSD GLASS® takes advantage of the log4j Java-based logging utility. It also makes use of the Web application server native log system to display application runtime messages.

More information about log4j is available at <http://logging.apache.org/log4j/1.2/manual.html>.

A sample log4j configuration file can be found in Appendix 1.

This sample can be used to get 4 log files corresponding to each one of the 4 applications.

## Application security configuration

This section gives information about configuring access to RSD GLASS® Policy Manager and other RSD GLASS® applications.

Policy Manager is the central point of control for which roles may perform which actions on which resource types.

Initial access to Policy Manager and other RSD GLASS® modules functionality is configured in RSD Admin Center® at Home > PolicyManager > Role Management.

If you have a role as a Super Administrator, and RSD GLASS Policy Manager is in a cloud deployment with multiple tenants, then a list of all tenants' role permission schema will be proposed.

Before a user is granted access to a functionality of RSD GLASS® an initial access control is made with the action that the functionality performs on the corresponding resource type of the RSD GLASS® application. For example the Policy Manager Laws menu is shown only if the user's role has the Policy Manager application checked under Authorized Applications and "Full control" is checked for "Law" under "Functions for {role}" and RSD GLASS Policy Manager Application".

Note additional controls may be made to further limit access in the specific application with ACLs and Security Metadata rules on specific resource instances.

Here is a list of each application's Function names with the actions they are controlled against.

### RSD GLASS Policy Manager:

resource type name	actions the resource is controlled against	notes
Access Control List	CREATE, READ, UPDATE, DELETE	Management of Access Control List on objects

---

<sup>1</sup> Spring Batch JobOperator: "<http://static.springsource.org/spring-batch/reference/html/configureJob.html#JobOperator>"

<sup>2</sup> Spring Batch Admin User Guide: "<http://static.springsource.org/spring-batch-admin/reference.html>"

resource type name	actions the resource is controlled against	notes
Action	CREATE, READ, UPDATE, DELETE	Management of Lifecycle Actions objects
Action Workflow	CREATE, READ, UPDATE, DELETE	Management of Lifecycle Action workflow objects
Administration	CREATE, READ, UPDATE	Management of Administration of Policy Manager
Audit settings	CREATE, READ, UPDATE, DELETE	Management of Policy Manager audit parameters
Close Review	CREATE, READ, UPDATE, DELETE	Management of Record Class Close Review workflow
Code Template	CREATE, READ, UPDATE, DELETE	Management of Codification templates
Export	CREATE	Access to Policy Manager Export feature
Folder Template	CREATE, READ, UPDATE, DELETE	Management of Folder template objects
Jurisdiction	CREATE, READ, UPDATE, DELETE	Management of Jurisdiction objects
Language	CREATE, READ, UPDATE, DELETE	Management of Language objects
Law	CREATE, READ, UPDATE, DELETE	Management of Law objects
Legal Case	CREATE, READ, UPDATE, DELETE	Management of Legal Case Type objects
Legal Content	CREATE, READ, UPDATE, DELETE	Management of Legal Content Provider objects
Metadata	CREATE, READ, UPDATE, DELETE	Management of Metadata / Metadata Group / Metadata Group Types object
Policy Action Event	CREATE, READ, UPDATE, DELETE	Management of Lifecycle Events object
Public View	CREATE, READ, UPDATE, DELETE	Management of Public View objects
Record Class	CREATE, READ, UPDATE, DELETE, TRAVERSE	Management of Record Class objects
Record Class review	CREATE, READ, UPDATE, DELETE	Review of Record Class objects
Record Type	CREATE, READ, UPDATE, DELETE	Management of Record Type objects
Retroactive	CREATE, READ, UPDATE, DELETE	Access to Record Class Retroactivity feature (entering a retroactive date for a new version of a Record Class)
Open/Close Action	CREATE (open), DELETE (close)	Access to Open/Close Record Class action
Search	CREATE, READ, UPDATE, DELETE	Access to Search Record Class panel

resource type name	actions the resource is controlled against	notes
Security properties	CREATE, READ, DELETE	Configure general security parameters via RSDAdminCenter
Survey	CREATE, READ, UPDATE, DELETE	Management of Classification Requirements objects
Survey Management	CREATE, READ, UPDATE, DELETE	Access to Classification Requirements Management panel
Version	CREATE, READ, UPDATE, DELETE	Management of Record Class Versions
View Template	CREATE, READ, UPDATE, DELETE	Management of Public View objects
Workflow	CREATE, READ, UPDATE, DELETE	Access to Record Class Validation Workflow panel

## RSD GLASS Governance Manager/Client:

resource type name	actions the resource is controlled against	notes
Access Control List	CREATE, READ, UPDATE, DELETE	Management of Access Control List on objects
Administration	CREATE, READ, UPDATE, DELETE	Management of Administration of Governance Manager
Audit settings	CREATE, READ, UPDATE, DELETE	Management of Governance Manager audit parameters
Batch Job	CREATE, READ, UPDATE, DELETE	Access to Batch job panel (to use GovernanceManager Batch Management in settings menu)
Box	CREATE, READ, UPDATE, DELETE	Management of Box objects
Box withdrawal	CREATE	Management of Box withdrawal
Business item	READ, UPDATE, DELETE	Management of Business items
Business View	CREATE, READ, UPDATE, DELETE	Management of Business Views
Component	CREATE, READ, UPDATE, DELETE	Management of Component objects
Declared Record	DELETE	Management of Declared Records
Entity Owner	CREATE, READ, UPDATE, DELETE	Ownership Management
Execute Action	CREATE, READ, UPDATE, DELETE	Access to Execute action panel
Folder	CREATE, READ, UPDATE, DELETE, TRAVERSE	Management of Folder objects
GLASS Client Settings	CREATE, READ, UPDATE, DELETE	to administrate GovernanceManager Converter parameters in RSDGlassAdmin
Integrity Data	CREATE, READ, UPDATE, DELETE	Management of Integrity properties on objects (required to modify component integrity data)

resource type name	actions the resource is controlled against	notes
Legal Case	CREATE, READ, UPDATE, DELETE	Access to Legal case panel
Legal Close	CREATE	Management of Legal close action
Legal Hold	CREATE	Management of Legal hold action on objects
Manual Enforcement	CREATE, READ, UPDATE, DELETE	Access to Manual enforcement panel
Mapping Tool	CREATE, READ, UPDATE, DELETE	Access to Mapping tools panel
Metadata	CREATE	Search on metadata
Metadata	CREATE, READ, UPDATE	Manage metadata values
Owner Metadata	UPDATE	Controls update owner
Reclassification	CREATE	Management of Reclassification of a Record or a component
Record	CREATE, READ, UPDATE, DELETE, TRAVERSE	Management of Record objects
Record Class	CREATE, READ, UPDATE, DELETE, TRAVERSE	Management of Record Class objects
Record Date	CREATE, READ, UPDATE, DELETE	Management of Record Date values
Repositories	CREATE, READ, UPDATE, DELETE	Management of Repositories objects
Declare Action	CREATE	Access to Declare action on objects
Declared Record	DELETE	Access to Delete a declared Record
Hold Action	CREATE, DELETE (un-hold)	Access to Hold action on objects
Move Action	CREATE	Access to Move action on objects (controls if the move facility can be performed on the contents of a given file plan)
Open/Close Action	CREATE	Access to Open/Close action on objects
RM File Plan View	CREATE, READ, UPDATE, DELETE, TRAVERSE	Access to RM view panel (required to view file plans. Added so that some or all file plans can be hidden except for their business views)
Schedule Action	CREATE, READ, UPDATE, DELETE	Access to Schedule action panel
Search	CREATE, READ, UPDATE, DELETE	Access to Full Text Search
Search Index	CREATE, READ, UPDATE, DELETE	Access to Index Search (advanced search)
Search Settings	CREATE, READ, UPDATE, DELETE	Management of Search queries objects (to administrate GovernanceManager Content Search parameters in RSD Admin Center®)

resource type name	actions the resource is controlled against	notes
Security properties	CREATE, READ, DELETE	Configure general security parameters via RSDAdminCenter
Trigger Action	CREATE, READ, UPDATE, DELETE	Access to Trigger Actions panel
Workspace (File Plan or Business View)	CREATE, READ, UPDATE, DELETE, TRAVERSE	Access to Governance Manager Workspace

For RSD GLASS® for Physical Records, READ BOX is required for those roles that need to be able to recall an archived physical component's box, CREATE BOX is required for those roles that need to create boxes and archive them for the first time, and UPDATE BOX / DELETE BOX for those roles that can create boxes.

## RSD GLASS Physical\_Records:

resource type name	actions the resource is controlled against	notes
Access Control List	CREATE, READ, UPDATE, DELETE	Management of Access Control List on objects
Administration	CREATE, READ, UPDATE, DELETE	Management of Administration of Physical Records
Audit settings	CREATE, READ, UPDATE, DELETE	Management of Physical Records audit parameters
Box	CREATE, READ, UPDATE, DELETE, TRAVERSE	Management of Box objects
Box Type	CREATE, READ, UPDATE, DELETE	Access to Box type panel
Container	CREATE, READ, UPDATE, DELETE, TRAVERSE	Management of Container objects
Container Type	CREATE, READ, UPDATE, DELETE	Access to Container type panel
Contract	CREATE, READ, UPDATE, DELETE, TRAVERSE	Management of Contract objects
Global Physical Records Parameters	CREATE, READ, UPDATE, DELETE	Access to Global parameters panel
Logistics Operations	CREATE, READ, UPDATE, DELETE	Access to Logistics operations panel
Physical Component	CREATE, READ, UPDATE, DELETE, TRAVERSE	Management of Physical properties on objects
Physical Records Settings	CREATE, READ, UPDATE, DELETE	Management of Physical records settings objects
Recall Request	CREATE, READ, UPDATE, DELETE	Management of Recall properties on objects
Security properties	CREATE, READ, DELETE	Configure general security parameters via RSDAdminCenter
Site	CREATE, READ, UPDATE, DELETE	Management of Site objects
Storage Provider	CREATE, READ, UPDATE, DELETE	Management of Storage properties on objects

resource type name	actions the resource is controlled against	notes
Store Request	CREATE, READ, UPDATE, DELETE	Management of Store properties on objects
Warehouse Operations	CREATE, READ, UPDATE, DELETE	Management of Warehouse properties on objects

**Note:** users with the 'GlassRoot' role will never be denied access. After initial configuration the use of other roles should be preferred.

## ACL evaluation details

For an access control request of an action to be permitted on a particular node or node capability, for a particular user, the following controls are evaluated (not necessarily in this order).

- RSD Admin Center policy rules are evaluated ensuring that the users role may perform the given action on the: capability, or type of the node; for the module GLASS. If this fails, access is denied.
- ACL rule evaluation:

The rules are automatically split into deny/permit rules for: users, groups, roles, any subject; and evaluated in the following order;

Rules for:

- deny glass client node owner, permit glass client node owner,
- deny users, permit users,
- deny groups, permit groups,
- deny roles, permit roles,
- deny any subject, permit any subject

If any matching rule is found the first one is used.

If no particular rule matches for a given ACL, the ACLs of the parent nodes will be evaluated in order from child to parent until a matching rule is found.

The result (Permit or Deny) of the matching rule is used only if the other controls do not evaluate to deny. If no matching rule is found in any ACL, access is denied.

- Parent ACL traverse action rules: The equivalent of an access control request of the action TRAVERSE on each parent node is evaluated. If any of those evaluates to deny, access is denied.
- Metadata rules (Record class security metadata) are evaluated for the given node and if any activation rules evaluate to true against the nodes security metadata, and any deny rules evaluate to true, access is denied.

## Governance Manager subject attributes

For the security metadata defined on imported Policy Manager record classes to be applied properly, default values for subject attribute values corresponding to these metadata must be defined.

For each imported metadata name there should be a default value defined with the RSDAdminCenter operation `com.rsd.glass/Manager/GovernanceManager/Security/MetaDataPolicyManager/addDefaultSubjectAttribute`.

Set the value equal to 0 for integer attributes, and the desired value for the string attributes.

If the security property `security.default ldap.useSubjectAttributes` is false or `authenticationMode` is SAML (subject attributes not configured to come from LDAP), then use `com.rsd.glass/Manager/GovernanceManager/Security/UserManager/UserAttribute` operations of RSD Admin Center® set any users attribute values that should be different than the defaults for access.

The keys used to set the values for subject attributes and their defaults are in the form `"metadataName#integer"` or `"metadataName#string"` depending on the type of the metadata.

There may be RSD GLASS® Physical Records ACL rule resource target for each site associated with the contract. The resources are in the form `Site/site name`. This allows the sites to be restricted for specific subjects. For the site restrictions to be in effect the subject must not have the right to update the contract (which requires access to all sites).

# Securing RSD GLASS® communications

## Introduction

This chapter describes the communications flow that takes place in the RSD GLASS® Solution. It will suggest ways to secure those communications to avoid sharing information unencrypted over the network. It will mostly explain how to encrypt the communication between two parts and ensure nobody can intercept any data in between.

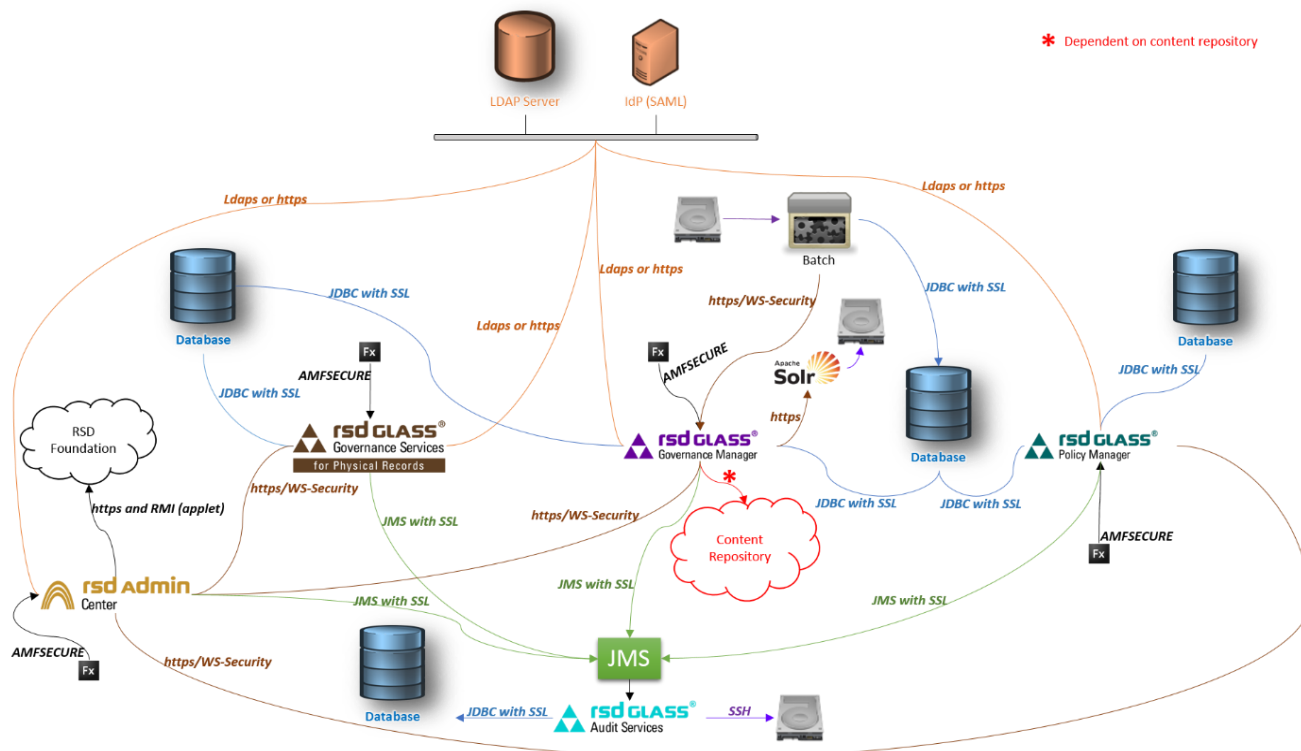
However, be aware that securing some communications can have a significant impact on application performance like, for example, between an application and its database.

Encryption could be avoided if the communication takes place in a secured private network, but as soon as the communication goes over a public network, encryption must be considered.

## Communications in RSD GLASS®

### Schema












You can find below a full picture of communication between the applications. Each link suggests how communication security could be achieved:



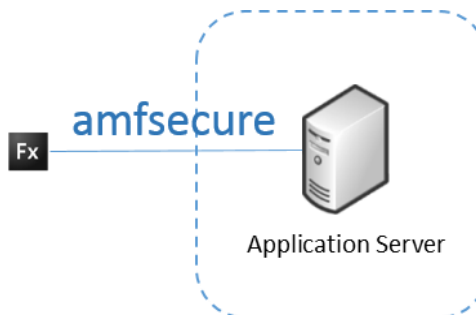


## Matrix

Here is a matrix of all the communications for a given application:

	AMF SECURE https (RSD Glass Apps)	WS https WS- Securit y 	WS https WS- Securit y 	WS https WS- Securit y 	Database JDBC / SSL	LDAP Idaps	IdP (SAML) https	SOLR https 	JMS JMS / SSL	Content Repositor ies (Depende nt on underlyin g content repositor y)	Hard Drive or Network Drive SSH	WS RMI https WS- Security RSD Glass Foundatio n
	X											
		X	X	X	X	X	X		X			X
					X	X	X		X			
					X	X	X	X	X	X		
					X	X	X		X			
											X	
											X	
<b>Batch</b>					X						X	

## Secure AMF (BlazeDS) communication



RSD GLASS® Governance Manager, RSD GLASS® Policy Manager, RSD GLASS® for Physical Records and RSD Admin Center have their front-end client made in Adobe Flex. The Flash application uses Action Message Format (AMF) protocol for the back-end communication.

BlazeDS is used as a server-based Java remoting and web messaging technology and allows you to connect to back-end distributed data and push data to Adobe Flex. BlazeDS applications consist of client-side code and server-side code. Client-side code is typically a Flex application written in MXML and ActionScript and deployed as a SWF file. Server-side code is written in Java.

When you need to secure communication between a Flex Application and the back-end you have to define a secured communication channel in order to encrypt AMF messages.

The RSD GLASS® Solution front-ends have already enabled the secure communication configuration. There are two channels defined in the *services-config.xml* file located in each RSD GLASS® Application war file. One channel uses the default broker amf url

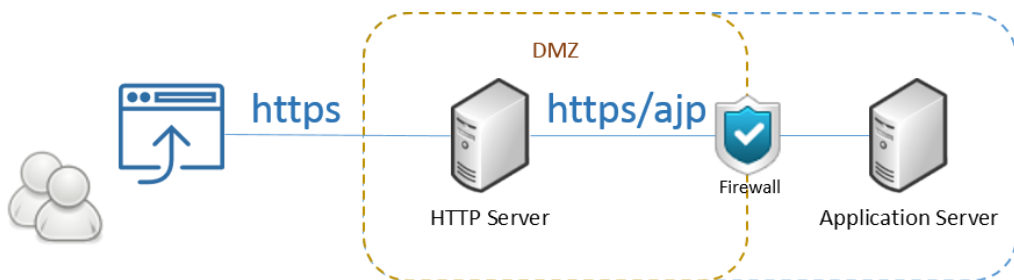
“/messagebroker/amf” and the second channel uses a secured broker amf url:  
“/messagebroker/amfsecure”

The only thing needed is having the HTTP server (or Application server if set in front) allowing SSL to communicate over HTTPS.

## Securing the HTTP Server with HTTPS

Usually an HTTP server (i.e.: Apache HTTP server) is put between the Flex UI and the Application Server (i.e.: Tomcat, JBoss, IBM Websphere...). It adds more security and performance.

If the application is accessed through the Internet, communication with the HTTP server and the Flex UI should be encrypted:



Usually, an HTTP server lives in the DMZ and receives HTTPS requests. HTTPS is normally only necessary over the internet, and the HTTPS payload should be removed from the Apache server before forwarding the (de-crypted) request to the application server servlet engine.

In order to implement SSL you will need to get a valid certificate. There are multiple certification authorities available like Thawte, VeriSign, etc. that will certify the ownership of your public key. After having chosen a certificate of authority that best suits your needs, you can install it on your HTTP server.

## Apache HTTP Server SSL installation

1. Save primary and intermediate certificates to a folder on the server with the private key. Important: the private key must be kept away from unauthorized people.
2. Open httpd.conf file and find the “<VirtualHost>” element block. This block is usually at the bottom of the file, or can be found in a separate file like ssl.conf.
3. If you want to keep both secure (https) and non-secure (http) communications available, you will need a VirtualHost for each, you can copy the current “<VirtualHost>” element block and change port from 80 to 443.
4. Add the lines in bold below (adapt to your configuration and file names):

```
<VirtualHost 192.168.0.1:443>  
    DocumentRoot /var/www/website  
    ServerName www.domain.com  
    SSLEngine on  
    SSLCertificateFile /etc/ssl/crt/primary.crt  
    SSLCertificateKeyFile /etc/ssl/crt/private.key  
    SSLCertificateChainFile /etc/ssl/crt/intermediate.crt  
</VirtualHost>
```

5. Save changes.
6. Restart server.

## Enabling SSL on Apache Tomcat

In order to support SSL on Tomcat, you would have to define a SSL connector in the \$CATALINA\_BASE/conf/server.xml file. Normally, the connector is commented so you can uncomment it and if not you should add a new connector configuration element like this (adapt to your configuration and file names):

```
<!-- Define a SSL Coyote HTTP/1.1 Connector on port 8443 -->
<Connector
    protocol="HTTP/1.1"
    port="8443" maxThreads="200"
    scheme="https" secure="true" SSLEnabled="true"
    SSLCertificateFile="/usr/local/ssl/server.crt"
    SSLCertificateKeyFile="/usr/local/ssl/server.pem"
    SSLVerifyClient="optional" SSLProtocol="TLSv1"/>
```

You will find more detail in Apache's documentation "SSL Configuration HOW-TO".

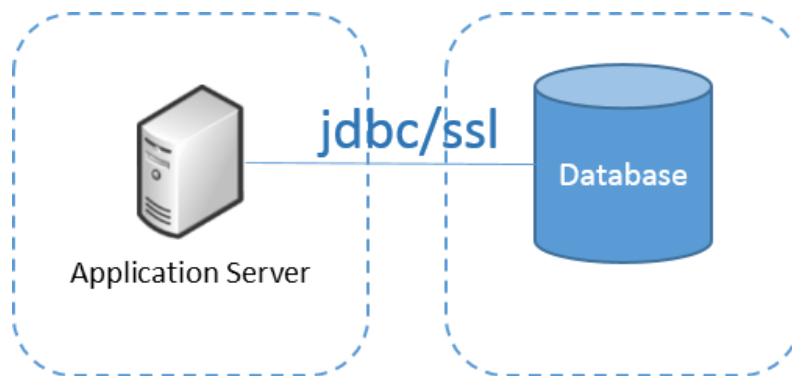
## Configuration Examples

See Appendix 2, "HTTP Server configuration examples".

## Secure Database communication

Communications from an application to the database can be secured by using SSL.

Configurations will depend on the database engine you will use, and you will have to refer to its documentation in order to enable SSL communication.



## Client Configuration

RSD GLASS® Solution uses JDBC drivers to connect to a database. The JDBC driver must be configured in order to communicate via SSL, that can be achieved by adding one or more properties to the connection url.

For example, here is how you configure your url for a MySQL jdbc driver:

```
jdbc:mysql://127.0.0.1:3306/glass?verifyServerCertificate=true&useSSL=true&requireSSL=true
```

- **verifyServerCertificate**: if this property is set to "true", it indicates if the certificate validity must be checked. For example if a self-signed certificate is used, it would fail.

- useSSL: if this property is set to “true”, it indicates that the connection will attempt to use SSL if the database is configured to support it.
- requireSSL: if this property is set to “true”, it indicates if the connection should fail if the communication cannot be done via SSL.

## Server Configuration

In order to enable SSL on MySQL server you will have to indicate where it can find the needed CA certificate, public and private keys. The simplest way is to add the entries in the “my.cnf” file located in your MySQL Server installation (adapt to your configuration and file names):

```
# SSL
ssl-ca=/etc/mysql-ssl/ca-cert.pem # CA Certificate
ssl-cert=/etc/mysql-ssl/server-cert.pem # Public Key
ssl-key=/etc/mysql-ssl/server-key.pem # Private Key
```

You can also set those parameters on the command line when starting the server:

```
mysqld --ssl-ca=ca-cert.pem --ssl-cert=server-cert.pem --ssl-key=server-key.pem
```

It is also good to create users to permit only SSL encrypted connection by adding “REQUIRE SSL” on the grant privileges statement:

```
GRANT ALL PRIVILEGES ON *.* TO 'glassuser'@'%' IDENTIFIED BY 'pass' REQUIRE
SSL;
```

You can determine whether the current connection with the server uses SSL by checking the value of the Ssl\_cipher status variable. The value is nonempty if SSL is used, and empty otherwise. For example:

```
mysql> SHOW STATUS LIKE 'Ssl_cipher';
+-----+-----+
| Variable_name | Value                |
+-----+-----+
| Ssl_cipher    | DHE-RSA-AES256-SHA |
+-----+-----+
```

## SSL Replication Configuration

If you are using MySQL replication mechanism, you can setup an SSL replication.

To use SSL for encrypting the transfer of the binary log required during replication, both the master and the slave must support SSL network connections. If either host does not support SSL connections (because it has not been compiled or configured for SSL), replication through an SSL connection is not possible.

Setting up replication using an SSL connection is similar to setting up a server and client using SSL. You must obtain (or create) a suitable security certificate that you can use on the master, and a similar certificate (from the same certificate authority) on each slave.

You must first stop the slaves, then on the MySQL server side you specify where it can find the needed CA certificate, public and private keys, by adding the entries in the “my.cnf” file located in your MySQL Server installation (adapt to your configuration and file names):

```
# SSL
ssl-ca=/etc/mysql-ssl/ca-cert.pem # CA Certificate
ssl-cert=/etc/mysql-ssl/server-cert.pem # Server Public Key
ssl-key=/etc/mysql-ssl/server-key.pem # Server Private Key
```

Then on each slave you can specify in their respective “my.cnf” file the following entries:

```
# SSL
ssl-ca=/etc/mysql-ssl/ca-cert.pem # CA Certificate
ssl-cert=/etc/mysql-ssl/client-cert.pem # Server Public Key
ssl-key=/etc/mysql-ssl/server-key.pem # Server Private Key
```

You can then start the slaves.

If you want to enforce the use of SSL connections during replication, then create a user with the REPLICATION SLAVE privilege and use the REQUIRE SSL option for that user.

For example:

```
mysql> CREATE USER 'repl'@'%.mydomain.com' IDENTIFIED BY 'slavepass';  
mysql> GRANT REPLICATION SLAVE ON *.*  
-> TO 'repl'@'%.mydomain.com' REQUIRE SSL;
```

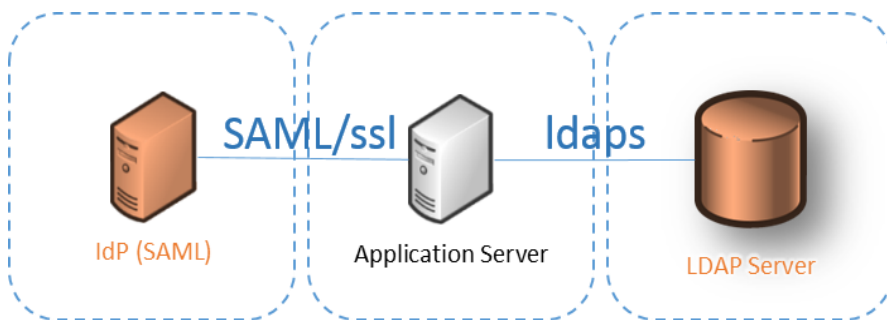
You will find more detail in the MySQL Reference Manual, chapter "Using SSL for Secure Connections".

## Secure web services communication

The RSD GLASS® solution exposes several SOAP or REST based web services.

In order to secure the communication, you need to enable SSL on the HTTP Server and the Application Servers as previously explained. This would secure the transmission of the message over the network and provide some assurance to the client about the identity of the server. This is transport level security. Transport level security secures your message only while it's on the network - as soon as it leaves the network, the message is no longer secured.

## Secure LDAP/SAML communication



## LDAP

The LDAP protocol is used to read and write in an Active Directory. By default LDAP traffic is transmitted in a non-secure way. It is possible to activate the LDAP protocol over SSL (LDAPS) by installing a certificate emitted by a certification authority.

The default non secured port is 389, when activating LDAPS, it will use default port 636.

### Server Configuration

Installing the certificate and keys will depend on your LDAP, but here is an example for OpenLDAP Server on Debian (adapt to your configuration and file names):

1. Install CA certificate:  

```
cp ~/ca/demoCA/cacert.pem /etc/ssl/certs/  
chmod go+r /etc/ssl/certs/cacert.pem
```
2. Copy ldap key and certificate files to /etc/ldap/ssl:  

```
mkdir /etc/ldap/ssl/  
cp ~/ca/ldap.dev.local-*.pem /etc/ldap/ssl/
```

3. Secure certificates:
 

```
ldap1:~# chown -R root:openldap /etc/ldap/ssl
ldap1:~# chmod -R o-rwx /etc/ldap/ssl
```
4. Enable ldaps protocol (file /etc/default/slapd):
 

```
LAPD_SERVICES="ldap://127.0.0.1:389/ ldaps:/// ldapi:///"
```
5. Create tls configuration file (tls-config.ldif):
 

```
dn: cn=config
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ssl/certs/cacert.pem
-
add: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ldap/ssl/ldap.dev.local-cert.pem
-
add: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ldap/ssl/ldap.dev.local-key.pem
```
6. Apply it:
 

```
ldapmodify -QY EXTERNAL -H ldapi:/// -f tls-config.ldif
```
7. Restart slapd:
 

```
/etc/init.d/slapd restart
```
8. Ensure it is started:
 

```
netstat -tunlp | grep slapd
tcp        0      0 0.0.0.0:636          0.0.0.0:*
LISTEN    2462/slapd
tcp        0      0 127.0.0.1:389       0.0.0.0:*
LISTEN    2462/slapd
```

## Client Configuration

On a running RSD GLASS® Solution, the ldap url must be changed from ldap://[ldaphost]:389/ to ldaps://[ldaphost]:636/. You can achieve this by using RSD Admin Center, and navigate to the “Configure general security parameters” on each application, and edit “ldapURL” entry. Be aware that modifying this entry would need a restart of the application server to make changes take effect.

## SAML

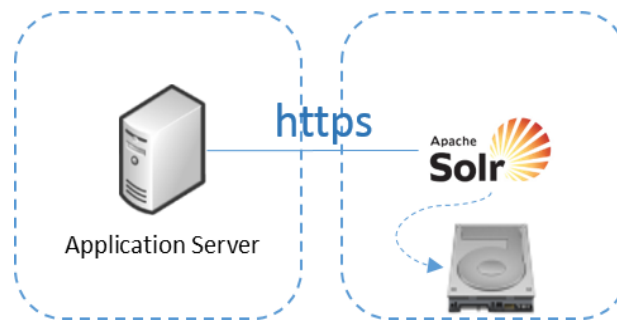
RSD GLASS® Solution is compatible with Security Assertion Markup Language (SAML).

SAML is an XML-based open standard data format for exchanging authentication and authorization data between parties, in particular, between an identity provider and a service provider.

SAML relies heavily on HTTP as its communications protocol, and specifies the use of SOAP. Knowing that, we can easily guess the use of SSL to secure the communication process.

Configuring support of SSL is dependent on your identity provider. For example, if you use Shibboleth, you would just have to install your certificate and your keys on the HTTP Apache server that Shibboleth is running on. Please refer to previous section [Apache HTTP Server SSL installation](#) on how to proceed.

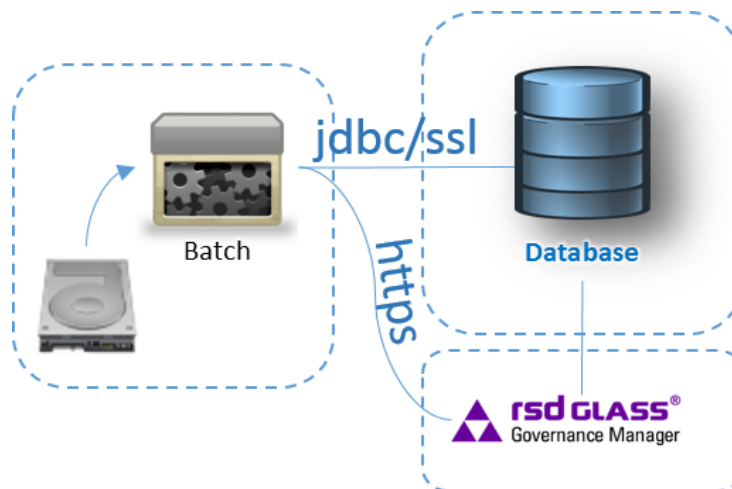
## Secure SOLR communication



Securing SOLR communications is very straightforward. The SOLR war file is deployed on a J2EE application server, and the only thing to do is to enable SSL on the application server in order to secure the communication. You will also need to update the SOLR configuration (index url and search url) in batch.properties file from http to https.

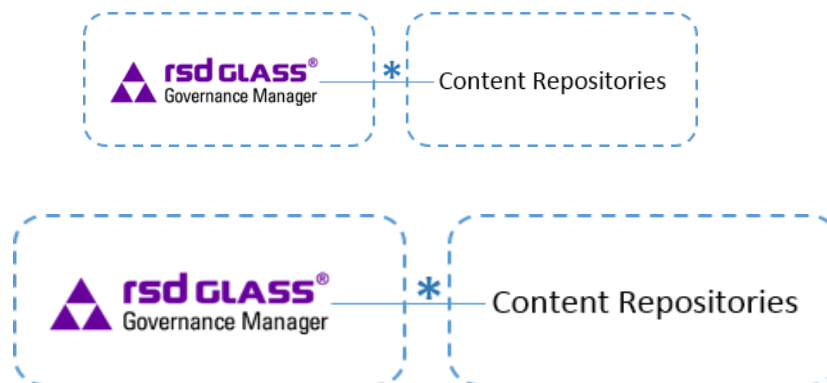
To enable SSL support on an Apache Tomcat with SOLR, please refer to the section [Enabling SSL on Apache Tomcat](#) and if you are using the “jettysolr” configuration that RSD provides, you should refer to Jetty’s Reference manual, chapter “Configuring SSL”.

## Secure Batch communication



The batch tool communicates with the database with JDBC over HTTP and RSD GLASS® with web services over HTTP. As already seen in previous sections in this document, you can apply the same mechanisms to secure communication with JDBC over HTTPS ([Secure Database communication](#)) and to secure communications with the HTTP server or Application server (Securing the HTTP Server).

## Secure Content Repository communication



Securing communications to a repository really depends on the repository itself. It must be determined how the connector interacts with the repository, and the repository encryption capabilities. However most of the connectors are using web services and therefore you should refer to the content repository documentation on how to install certificates and keys.

There are a few connectors that are using Common Internet File System (CIFS). CIFS is the standard way that computer users share files across corporate intranets and the Internet. It doesn't have any protocol level encryption options, therefore you can encapsulate the traffic in an encrypted envelope, for example by using a VPN.



# Appendices

## Appendix 1: sample log4j configuration files

### For Policy Manager:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">
<log4j:configuration debug="false" xmlns:log4j="http://jakarta.apache.org/log4j/">
  <appender name="RSDPOLICYMANAGERFILE" class="org.apache.log4j.RollingFileAppender">
    <param name="File" value="logs/rsdpolicymanager.log"/>
    <param name="Encoding" value="UTF-8"/>
    <param name="MaxBackupIndex" value="50"/>
    <param name="MaxFileSize" value="10000KB"/>
    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%d{yyyy-MM-dd HH:mm:ss} %5p {%X{TenantID}}
%X{CorrelationID} [%c{1}:%L] - %m%n"/>
    </layout>
  </appender>
  <appender name="ASYNC" class="org.apache.log4j.AsyncAppender">
    <param name="Blocking" value="true" />
    <param name="BufferSize" value="4096" />
    <param name="LocationInfo" value="true"/>
    <appender-ref ref="RSDPOLICYMANAGERFILE" />
  </appender>
  <logger name="com.rsd.glass"><level value="warn" /></logger>
  <root>
    <level value="error" />
    <appender-ref ref="ASYNC" />
  </root>
</log4j:configuration>
```

### For Governance Manager:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">
<log4j:configuration debug="false" xmlns:log4j="http://jakarta.apache.org/log4j/">

  <appender name="RSDGLASSFILE" class="org.apache.log4j.RollingFileAppender">
    <param name="File" value="logs/rsdglass.log"/>
    <param name="Encoding" value="UTF-8"/>
    <param name="MaxBackupIndex" value="50"/>
    <param name="MaxFileSize" value="10000KB"/>
    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%d{yyyy-MM-dd HH:mm:ss} %5p {%X{TenantID}}
%X{CorrelationID} [%c{1}:%L] - %m%n"/>
    </layout>
  </appender>
  <appender name="ASYNC" class="org.apache.log4j.AsyncAppender">
    <param name="Blocking" value="true" />
    <param name="BufferSize" value="4096" />
    <param name="LocationInfo" value="true"/>
    <appender-ref ref="RSDGLASSFILE" />
  </appender>
  <logger name="com.rsd.glass"><level value="warn" /></logger>
  <root>
    <level value="error" />
    <appender-ref ref="ASYNC" />
  </root>
</log4j:configuration>
```

## Appendix 2 : HTTP Server configuration examples

Below are some examples on how to configure an HTTP Server in front of some Application Servers.

### Apache HTTP Server in front of Tomcat

[http://httpd.apache.org/docs/current/ssl/ssl\\_howto.html](http://httpd.apache.org/docs/current/ssl/ssl_howto.html)

[http://www.ntu.edu.sg/home/ehchua/programming/howto/ApachePlusTomcat\\_HowTo.html](http://www.ntu.edu.sg/home/ehchua/programming/howto/ApachePlusTomcat_HowTo.html)