

Alfresco One 5.0

Administering Alfresco



Contents

Administering.....	5
System paths.....	5
Starting and stopping Alfresco.....	6
Starting the Alfresco server.....	6
Stopping the Alfresco server.....	6
Starting Alfresco Share.....	6
Using the Admin Console.....	7
About the Alfresco Admin Console.....	7
Launching the Admin Console.....	7
Admin Console: System Summary.....	8
Admin Console: Consoles.....	9
Admin Console: Email services.....	13
Admin Console: General settings.....	16
Admin Console: Repository Services.....	20
Admin Console: Support Tools.....	27
Admin Console: Authentication directories.....	28
Admin Console: Virtual File Systems.....	40
Starting the Activiti workflow console.....	42
Customizing the Alfresco Admin Console.....	42
Using the Share Admin Tools.....	44
Opening the Share Admin Tools.....	44
Application settings.....	44
Managing categories.....	45
Using the Node Browser.....	46
Managing tags.....	47
Sites Manager.....	47
Managing replication jobs.....	48
Managing users.....	50
Managing groups.....	54
Configuring Alfresco.....	56
Configuration overview.....	57
Configuring databases.....	65
Configuring OpenOffice subsystem.....	74
Configuring synchronization.....	77
Configuring email.....	79
Configuring file servers.....	85
Configuring IMAP protocol support.....	94
Configuring system properties.....	98
Encrypting configuration properties.....	99
Configuring the repository.....	102
Configuring Alfresco subsystems.....	120
Setting up Alfresco authentication and security.....	124
Alfresco security.....	125
Authentication subsystems.....	126
Configuring authentication.....	129
Authorities.....	158
Defining permissions.....	160
Access Control Lists.....	163
Modifying access control.....	166
Public services.....	167
Implementation and services.....	170
Admin password in default authentication.....	173

Setting up clustering.....	175
Clustering prerequisites when upgrading to Alfresco One 5.0.....	175
Setting up Share cluster.....	176
Setting up repository server cluster.....	180
Tracking clustering issues.....	183
Configuring search.....	184
Configuring search in Alfresco Share.....	184
Solr overview.....	185
Configure Solr 4 search service.....	187
Solr 4 security.....	198
Solr 4 monitoring and troubleshooting.....	202
Solr 4 backup and restore.....	207
Full text search configuration properties for Solr 4 index.....	208
Using Filtered search.....	210
Setting Solr 4 log4j values.....	213
Calculate the memory needed for Solr 4 nodes.....	213
Transactional metadata query.....	215
Setting up Enterprise to Cloud Sync.....	219
Enterprise to Cloud Sync overview.....	219
Configuring Enterprise to Cloud Sync.....	220
Troubleshooting Enterprise to Cloud Sync.....	221
Managing transformations.....	224
Standard transform options.....	224
Additional transform options.....	284
File types that support preview and thumbnail generation.....	290
Creating and managing workflows.....	293
What is a workflow?.....	293
Setting up Hybrid workflow.....	295
Workflow architecture.....	296
Workflow tools.....	299
Process definitions.....	299
Task model.....	305
Setting up Activiti Designer.....	308
Deploying the task model.....	309
Deploying a process definition.....	309
Backing up and restoring.....	309
Backing up and restoring the repository.....	309
Migrating.....	310
Migrating servers.....	310
Using the Bulk Import tool.....	311
In-place bulk import.....	311
Streaming Bulk Import.....	312
Preparing the file system.....	312
Importing with the Bulk Import tool.....	314
Using content stores.....	320
Content stores overview.....	320
Content store types.....	321
Content store selector.....	335
Managing content stores.....	338
Configuring templated nodes and space templates.....	339
Setting files as templates.....	340
Setting folders as templates.....	340
Setting up Alfresco multi-tenancy.....	340
Enabling multi-tenancy.....	340

Multi-tenancy administration.....	341
Features not supported in a multi-tenant environment.....	342
Setting up content replication.....	342
Configuring content replication.....	342
Creating a new transfer target for content replication.....	343
Opening locked content in the source repository.....	343
Configuring the File System Transfer Receiver.....	344
Setting up the File System Transfer Receiver.....	344
Start File System Transfer Receiver.....	344
File System Transfer Receiver launch properties.....	345
File System Transfer Receiver custom properties.....	345
File System Transfer Receiver log file properties.....	346
Monitoring Alfresco.....	346
JMX monitoring and management extensions.....	346
Scheduled jobs.....	347
Auditing Alfresco.....	347
Audit configuration and environment.....	349
Audit filters.....	349
Content auditing.....	351
Audit sample files.....	356
Enabling auditing.....	356
Auditing examples.....	358
Audit configuration files.....	359
Built-in data producers.....	360
DataExtractors and DataGenerators.....	360
Locating the audit code.....	361
Defining the audit application.....	362
Simple audit query.....	363
Advanced audit query.....	365
Understanding PathMappings.....	365
Audit recording values.....	367
Using values that have changed in a post method call.....	369

Administering

This section describes the processes and procedures for maintaining and administering an Alfresco production environment.

System paths



- Explicit Windows paths use back slashes
C:\Adirectory
- Explicit Linux paths use forward slashes
/srv/adirectory
- Back slashes also indicate the same path can apply in both Windows or Linux environments
\adirectory\

Alfresco installation location

The Alfresco installation directory is referenced throughout this guide as `<installLocation>`.

`<classpathRoot>` directory (Windows)

The `<classpathRoot>` is a directory whose contents are automatically added to the start of your application server classpath. The location of this directory varies depending on your application server. For example:

- (Tomcat) C:\Alfresco\tomcat\shared\classes

`<classpathRoot>` directory (Linux)

The `<classpathRoot>` is a directory whose contents are automatically added to the start of your application server classpath. The location of this directory varies depending on your application server. For example:

- (Tomcat) tomcat/shared/classes/

alfresco-global.properties file

The `alfresco-global.properties` file is where you store all the configuration settings for your environment. The file is in Java properties format, so backslashes must be escaped. The file should be placed in `<classpathRoot>`. When you install Alfresco using the setup wizard, an `alfresco-global.properties` file is created, which contains the settings that you specified in the wizard. An `alfresco-global.properties.sample` file is supplied with the setup wizard and also with the WAR zip file. This `.sample` file contains examples of common settings that you can copy into your `alfresco-global.properties` file.

`<extension>` directory

The `<extension>` directory is where you store Spring configuration files that extend and override the system configuration. This directory can be found at `<classpathRoot>\alfresco\extension`.

<web-extension>

The <web-extension> directory is where you store Spring configurations that extend and override the system Share configuration. This directory can be found at <classpathRoot>\alfresco\web-extension.

<solrRootDir>

The <solrRootDir> directory is the Solr home directory which contains the Solr core directories and configuration files. This directory can be found at <ALFRESCO_HOME>\solr4.

Starting and stopping Alfresco

This section describes how to run the Alfresco server and Share.

Starting the Alfresco server

- If you installed Alfresco as a service, from the **Start** menu, select **All Programs > Alfresco One > alfresco manager tool**, and start the `Tomcat Server` and `Postgres` services.
- Alternatively, from a command prompt, navigate to the Alfresco installation directory (`C:/Alfresco`) and run `servicerun START`.
You need administrator rights to run this command.
These services are also available from the **Start** menu under **Control Panel > System and Security > Administrative Tools > Services**.
- If you installed Alfresco as a service, double click the **Application Manager** tool in the Alfresco root directory and start the `PostgreSQL Database` and `Tomcat Server` services.
- Alternatively, browse to `/opt/alfresco/` and run `./alfresco.sh start` as an administrator.
 - 🚫 If you installed Alfresco using the setup wizard, the `alfresco.sh` script included in the installation disables the Security-Enhanced Linux (SELinux) feature across the system.
 - ✏️ The default shell for this script is `sh`. You can edit the `alfresco.sh` file to change to your preferred shell. For example, change the `#!/bin/sh` line to `#!/bin/bash`.

Stopping the Alfresco server

- (Windows)
 - Open the Control Panel **Services** window and stop the following services:
 - `alfrescoPostgreSQL`
 - `alfrescoTomcat`
 - Click the **Start** menu, and select **All Programs > Alfresco Enterprise > Alfresco Enterprise Service > Stop Alfresco Enterprise service**.

The command prompt that opened during startup closes. Alfresco has now stopped.

- (Linux) Browse to `/opt/alfresco/`, and run `./alfresco.sh stop`.

Starting Alfresco Share

Once you have installed Alfresco, you can start Alfresco Share using a browser.

1. Browse to the location of your Alfresco installation.
For example, `http://<your-host>:8080/share`.
In Windows, alternatively, you can click the **Start** menu, and select **All Programs > Alfresco Enterprise > Alfresco Share**.
Alfresco Share opens in a browser.
2. Log in using a user name and password.

Using the Admin Console

The Admin Console is an Enterprise only application that gives you control over the management and settings of the Alfresco environment.

You'll find help text on the Admin Console pages to assist you with setting up your Alfresco repository.

About the Alfresco Admin Console

The Alfresco Admin Console is a standalone console for managing the administration of the Alfresco repository.

The Admin Console is a tool comprising separate pages that identify a particular administrative activity or feature. The pages in the Admin Console are:

- System Summary
- Email Services
- General
- Repository Services
- Support Tools
- Authentication directories
- Virtual File Systems

You can use the Admin Console as your main tool to help you manage your Alfresco production environment. It is a simple alternative to using a JMX console, or manually setting properties in the `alfresco.global.properties` file.

The settings that you choose in the Admin Console will take precedence over any setting that you add in the `alfresco.global.properties` file.

Launching the Admin Console

Ensure that the Alfresco server is running.

1. Enter the following URL in a browser window:

```
http://<your-host-name>:<alfresco-port>/alfresco/service/enterprise/admin
```

Where `<your-host-name>` is the host name where you are running the Alfresco server and `<alfresco-port>` is the port number on which the Alfresco server is running (by default, the port number is 8080).

An **Authentication Required** prompt displays, showing the IP address or name and the port number of the Alfresco server.

2. Enter your Alfresco user name and password.

Your user name and password must be for an account with administrator permissions.

The Admin Console displays in a browser window. The title bar shows the host name and its IP address.

You will remain logged into the Admin Console for the duration of the browser session. If you close the browser window completely and then connect to the Admin Console using the URL, you will be prompted to enter your Alfresco account details again.

A useful starting point in the Admin Console is the [System Summary](#) page, which gives an overview of the which settings are enabled or disabled.

Admin Console: System Summary

The System Summary page shows an overview of the status of the Alfresco repository, including the general system information, subsystem status, clustering settings, the current authentication chain, and details of which AMPs are applied to the system.

There are no actions or entry fields on the System Summary page. This page is a high-level overview of the setting you have chosen or are set as default on the repository.

The overview is divided into the following sections:

- System information
- File Systems
- Transformation Services
- Indexing Subsystem
- Repository Clustering
- Activities Feed
- Authentication
- Email
- Auditing Services
- Content Stores
- Alfresco Module Packages
- Users and Groups

System Information

The System Information summary shows the general details of the Alfresco installation. This information is useful for confirming the Alfresco installation details, Java installation details, the host operating system specification and memory details.

File Systems

The File Systems summary shows the settings from the File Servers page. See [Enabling File Servers](#) for more information.

Transformation Services

The Transformation Services summary shows the settings from the Transformation Services page. See [Working with the Transformation Services](#) for more information.

Indexing Subsystem

The Indexing Subsystem summary shows the settings from the Search Service page. See [Working with the Search Service](#) for more information.

Repository Clustering

The Repository Clustering summary shows the settings from the Repository Server Clustering page. See [Repository Server Clustering](#) for more information.

Activities Feed

The Activities Feed summary shows the settings from the Activities Feed page. See [Setting the Activities Feed](#) for more information.

Authentication

The Authentication summary shows the settings from the Directory Management page, in particular, the current authentication chain. See [Managing authentication directories](#) for more information.

Email

The Email summary shows the settings from the Inbound Email and Outbound Email pages. See [Managing inbound emails](#) and [Managing outbound emails](#) for more information.

Auditing Services

The Auditing Services summary indicates the status of auditing in Alfresco.

Content Stores

The Content Stores summary lists the location of the default content stores.

Alfresco Module Packages

The Alfresco Module Packages summary identifies which modules have been applied against this instance of Alfresco.

Users and Groups

The Users and Groups summary shows the number of individual users and groups within the system.

Admin Console: Consoles

The Consoles section on the Admin Console contains pages for administering models, tenants, and workflow definitions and property bundles at runtime in the repository.

Admin Console: Model and Messages Console

The administrator can manage repository models and message resource bundles using the Model and Messages Console.

1. Open the Admin Console.
2. In the **Consoles** section, click **Model and Messages Console**. You see the **Model and Messages Console** page.
3. Perform the following as required for administering models:
 - a. To list all deployed models that are stored in the repository data dictionary, type `show models`.
 - b. To upload model to repository and load into runtime data dictionary, type `deploy model`.
This command also sets the repository model as active. If a model is already deployed, then it will be updated and re-deployed.
`deploy model alfresco/extension/exampleModel.xml`
 - c. To permanently delete model from repository (all versions) and unload from runtime data dictionary, type `undeploy model`.
`undeploy model exampleModel.xml`
 - d. To set repository model to active and load into runtime data dictionary, type `activate model`.
`activate model exampleModel.xml`

- e. To set repository model to inactive and unload from runtime data dictionary, type `deactivate model`.

```
deactivate model exampleModel.xml
```
4. Perform the following as required for administering message resource bundles:
 - a. To list all deployed message resource bundles that are stored in the repository data dictionary, type `show messages`.
 - b. To upload message resource bundle to repository and runtime message service, type `deploy messages`.

```
deploy messages alfresco/extension/lifecycle-messages
```
 - c. To remove message resource bundle from repository and from runtime message service, type `undeploy messages`.

```
undeploy messages lifecycle-messages
```
 - d. To reload message resource bundle from repository into runtime message service, type `reload messages`.

```
reload messages lifecycle-messages
```

Managing tenants

1. Open the Admin Console.
2. In the **Consoles** section, click **Tenant Console**. You see the **Tenant Console** page.
3. Perform the following as required:
 - a. To list all tenants and show their details, type `show tenants`.
 - b. To show details for a single tenant, type `show tenant <tenant domain>`.
 This shows the status (for example, whether it is enabled or disabled) and the root content store directory.
 - c. To create a tenant, type `create <tenant domain> <tenant admin password> [<root contentstore dir>]`.
 For example, `create zzz.com l3tm31n /usr/tenantstores/zzz`
 This creates an empty tenant. By default the tenant will be enabled. It will have an administrator user called `admin@<tenant domain>` with the supplied password. All users that the administrator creates can log in using `<username>@<tenant domain>`. The root of the `contentstore` directory can be optionally specified. If it is not specified, or does not exist, the repository default root content store will be used (as specified by the `dir.contentstore` property). Specifying a unique content store root for each tenant is recommended to keep the tenants properly separated, for example, to allow the backup and restore of individual tenants.
 - d. To enable a tenant, type `enable <tenant domain>`.
 This enables the tenant so that it is active and available for new logins.
 - e. To disable a tenant, type `disable <tenant domain>`.
 This disables the tenant so that it is inactive and prevents tenant login.

Admin Console: Workflow Console

The administrator can manage workflows, including testing of newly developed workflows using the Workflow Console. It also supports the debugging/diagnosis of current in-flight workflows.



The Workflow Console must not be used to terminate in-flight WCM workflows. Doing so does not clean up the workflow sandboxes or locked content, leaving the Web Project in an inconsistent and unrecoverable state.

1. Open the Admin Console.

2. In the **Consoles** section, click **Workflow Console**. You see the **Workflow Console** page.
3. Perform the following commands as required for managing workflows:
 - a. To output the contents of the file located at `<definitionClassPath>`, type `show file <definitionClassPath>`.
where `<definitionClassPath>` is the class path to workflow definition file.
 - b. To deploy workflow definition to Alfresco server, type `deploy <workflowEngine> <definitionClassPath>`.
where `<workflowEngine>` is the name of workflow engine (jbpm or activiti) and `<definitionClassPath>` is the class path to workflow definition.
 - c. To redeploy the last workflow definition, type `redploy`.
 - d. To list the latest deployed workflow definitions or display all workflow definitions (including previous versions) with the additional keyword `all`, type `show definitions [all]`.
 - e. To use the workflow definition identified by `<workflowDefId>`, type `use definition [<workflowDefId>]`.
If you use `use definition []` instead, the currently selected workflow definition is shown.
 - f. To undeploy the latest version of the workflow definition identified by `<workflowDefId>`, type `undeploy definition <workflowDefId>`.
This will also terminate and remove all in-flight workflows associated with the definition. Do not use this function with WCM workflows unless there are no in-flight workflows for this definition.
If multiple versions of the definition exist, you will need to undeploy each version in turn to remove the definition completely or issue the `undeploy definition name` command.
 - g. To undeploy all versions of a workflow definition, type `undeploy definition name <workflowName>`.
Just like `undeploy definition`, all in-flight workflows associated with each version are terminated. Remember not to use this function with WCM workflows unless there are no in-flight workflows for this definition.
4. Perform the following commands as required for managing variables:

The following variables are defined automatically when the console starts. They may be deleted or modified.

 - `var bpm:package package 1` (test package of one document)
 - `var bpm:assignee person admin` (test assignee who is admin)
 - a. To show all defined variables, type `var`.
 - b. To define or update a variable, type `var <varName>[*]=<varValue>`.
where:
 - `<varName>` is the variable name
 - `[*]` defines a collection (if specified)
 - `<varValue>` is the variable value (comma-separated list of values)

```
var bpm:assignee*=admin,fred
var wf:notifyMe=true
```
 - c. To define or update a `(cm:person)` node ref variable, type `var <varName>[*] person <varValue>`.

where:

- `<varName>` is the variable name
- `[*]` defines a collection (if specified)
- `<varValue>` is the variable value (comma-separated list of values)

```
var bpm:assignee* person admin,fred
```

- d. To define or update a (`usr:authorityContainer`) node ref variable, type `var <varName>[*] group <varValue>`.

where:

- `<varName>` is the variable name
- `[*]` defines a collection (if specified)
- `<varValue>` is the variable value (comma-separated list of values)

```
var bpm:groupAssignee group GROUP_Engineering
```

- e. To define or update a (`bpm:workflowPackage`) node ref variable, type `var <varName> package <itemCount>`.

```
var bpm:package package 4
```

A new workflow package is created containing `<itemCount>` content items.

- f. To delete an existing variable, type `var <varName>=`.

5. Perform the following commands as required for managing workflows:

- a. To start a new workflow using the currently selected workflow definition, type `start [<varName[=varValue]]*`.

```
start bpm:assignee=david wf:predefined
```

- b. To display a list of active workflows for the currently selected workflow definition, type `show workflows [all]`.

This command display a list of all workflows (latest and previous versions of process definitions) when used with the additional keyword `all`.

- c. To use the specified `<workflowId>`, type `use workflow <workflowId>`.

- d. To describe the specified `<workflowId>`, type `desc workflow <workflowId>`.

- e. To display the workflow paths for the specified `<workflowId>`, type `show paths [<workflowId>]`.

If `<workflowId>` is omitted, the paths for the currently started workflow are shown.

- f. To describe the specified `<pathId>`, type `desc path <pathId>`.

This command includes the list of properties associated with the path.

- g. To display all available transitions for the specified `<workflowId>`, type `show transitions [<workflowId>]`.

If `<workflowId>` is omitted, the transitions for the currently started workflow are shown.

- h. To signal transition on specified `<pathId>`, type `signal <pathId> [<transitionName>]`.

If `<transitionName>` is omitted, the default transition is taken.

- i. To fire an event of custom `eventtype` against the specified path, type `event <pathId> <eventtype>`.

- j. To fire an event of custom `eventtype` against the specified path, type `event <pathId> <eventtype>`.

- k. To end (cancel) the specified `<workflowId>`, type `end workflow <workflowId>`.
 - l. To force deletion of the specified `<workflowId>`, type `delete workflow <workflowId>`.
Do not use this function with WCM workflows.
 - m. To force deletion of all in-flight workflows, type `delete all workflows`.
Do not use this function with WCM workflows.
6. Perform the following commands as required for managing workflow timers:
- a. To display a list of active timers for the currently selected workflow definition, type `show timers [all]`.
This command displays a list of all timers when used with the additional keyword `all`.
7. Perform the following commands as required for managing tasks:
- a. To list tasks assigned to the currently selected user, type `show my tasks`.
 - b. To list tasks completed by the currently selected user, type `show my completed`.
 - c. To list tasks in a pool for the currently selected user, type `show my pooled`.
 - d. To list the tasks associated with the specified workflow `<pathId>`, type `show tasks [<pathId>]`.
If `<pathId>` is omitted, the tasks associated with the currently selected workflow path are shown.
 - e. To describe the task identified by `<taskId>` user, type `desc task <taskId>`.
 - f. To update the state of the specified `<taskId>`, type `update task <taskId> [<varName[=varValue]>]*`.
Task properties are provided as name/value pairs or references to pre-defined variables.

```
update task jbpm$122 bpm:assignee=fred wf:notifyMe=false
```
 - g. To end the task identified by `<taskId>`, type `end task <taskId> [<transitionName>]`.
If `<transitionName>` is omitted, the default transition is taken.
 - h. To query for tasks, type `query task [predicate]*`.
If no predicates are provided, all in-progress tasks are returned (across all active workflows).

Admin Console: Email services

The Email section on the Admin Console contains pages for configuring email servers.

Admin Console: Managing inbound emails

Set these inbound email properties to activate sending and receiving site invites, and also for receiving activity notification emails.

1. Open the Admin Console.
2. In the **Email Services** section, click **Inbound Email**.
You see the **Inbound Email** page.
3. Set the email properties:

Inbound Email property	Example setting	What is it?
Enabled	No	Use check box to enable or disable the inbound email service. By default, it is not enabled.
Unknown User	anonymous	This is the user name to authenticate as when the sender address is not recognized.
Allowed Senders	.*	To allow senders, enter a comma-separated list of email REGEX patterns of allowed senders. If there are any values in the list, then all sender email addresses must match. For example:.*\@alfresco\.com, .*\@alfresco\.org.
Overwrite Duplicates	Yes	By default, duplicate messages to a folder will overwrite each other. Deselect this check box to keep duplicate messages and apply a unique number.
Maximum Server Connections	3	This provides the maximum number of connections allowed in order to control the performance of the system. To prioritize the email subsystem higher, increase this number. The default setting is 3.
SMTP Authentication Enabled	No	Use this check box to enable or disable the authentication of inbound email against the repository.
Email Server Port	25	This is the default port number for the email server.
Email Server Domain	alfresco.com	This is the default domain for the email server.
Blocked Senders		To block senders, enter a comma-separated list of email REGEX patterns, for example: .*\@hotmail\.com, .*\@googlemail\.com. If the sender email address matches a listed value, then the message will be rejected.
Email Authentication Group	EMAIL_CONTRIBUTORS	This is the name of the group in which users must be a member to add content to the repository by email. The default group is EMAIL_CONTRIBUTORS.

Inbound Email property	Example setting	What is it?
Transport Layer Security (TLS)	Enabled	<p>This enables the TLS protocol, which upgrades a plain text connection to an encrypted TLS or SSL connection instead of using a separate port for encrypted communication. Select the TLS support setting:</p> <ul style="list-style-type: none"> • Disabled: TLS support is disabled • Hidden: On the EHLO command, server support for TLS is hidden, though TLS will still be accepted if the client uses it • Enabled: On the EHLO command, server support for TLS is announced • Required: TLS authentication is required

4. Click **Save** to apply the changes you have made to the properties.
If you do not want to save the changes, click **Cancel**.

Admin Console: Managing outbound emails

1. Open the Admin Console.
2. In the **Email Services** section, click **Outbound Email**.
You see the **Outbound Email** page.
3. Set the email properties:

Outbound Email property	Example setting	What is it?
Hostname	smtp.example.com	This is the name of the SMTP(S) host server.
Encoding	UTF-8	This is the email encoding type. The default is UTF-8.
Editable Sender Address		This check box enables the From field in outbound emails to be edited to differ from the Default Sender's Address. When you deselect this check box, the Default Sender's Address is always used. You should deselect this option if your email server security settings require the From field to match the user name used for email server authentication.
Email Server Port	25	This is the default port number for the email server.

Outbound Email property	Example setting	What is it?
Default Sender's Address	alfresco@demo.alfresco.org	The default address that is used in the From field of outbound emails if no alternative is available.
Email Protocol	SMTP	Select a protocol from the list. This is the protocol that will be used when sending email.
Username	anonymous	The account user name that connects to the SMTP server. The user name and password are only required if your server requires them for authentication.
Password		The account user password.

4. Click **Save** to apply the changes you have made to the properties.
If you do not want to save the changes, click **Cancel**.

Admin Console: General settings

The General section on the Admin Console contains pages for managing your Alfresco license, viewing information about the repository, and the system settings.

Uploading a new license

The access and use of Alfresco is managed by your license. The license is a file that you upload into Alfresco, which sets limits on the maximum number of users and a maximum number of content objects that you can use. Your limitations are set when you purchase the license from Alfresco. To increase the limitations, contact Alfresco to obtain a new license.

You will receive an email confirming the purchase of your license, and a license file is attached to the email. The license file has a filename of <license-name>.lic. You use this license file to upload the license restrictions into your system.

Before you upload a new license, ensure that Alfresco is running and that you can access the Admin Console. When you first run Alfresco, it defaults to using a 30-day trial license. You must upload your purchased license to run the Alfresco server before the trial period has expired.

1. Copy the license file to the directory in which Alfresco is installed.
For example, on Windows, copy the file to the C:\Alfresco directory; on Linux, copy the file to /opt/alfresco-x.x.x.
2. Open the Admin Console.
3. In the General section, click **License**.
4. In the **License Management** section, choose from where you want to upload the license file.

There are two options for storing the Alfresco license:

Upload License which allows you to locate a license file anywhere on your system.

- a. Click **Upload License**.
You can then locate and select the license file from the directory structure.
- b. Select the file, and then click **Upload**.

The new license will be applied to the repository. This will take precedence over license files on the file system. You might also need to restart the server to enable any features added in the new license.

Apply New License which automatically applies a license file that is stored in the Alfresco install directory.

a. Click **Apply New License**.

This applies a new license that is stored on the file system. This option will not apply the license if the server has a license uploaded to the repository.

When you have uploaded your license, the `.lic` file is automatically renamed to `<license-name>.lic.installed`.

When your license is about to expire, you must purchase a new license and upload it to your system. When you purchase further licenses, repeat the same steps using the new license file.

 An Alfresco license is unique to a specific version of Alfresco. When you upgrade to a new version of Alfresco, you will need to install a new license.

Repository information

1. Open the Admin Console.
2. In the **General** section, click **Repository Information**.

You see the **Repository Information** page showing the details of your Alfresco installation.

System settings

1. Open the Admin Console.
2. In the **General** section, click **System Settings**.
3. Set the Alfresco Repository Settings properties:

These properties are read-only and are set in the `alfresco-global.properties` file only. See the properties starting with `alfresco.` in [sysAdmin subsystem properties](#) on page 98.

Alfresco Repository Settings property	Example setting	What is it?
Repository Context	alfresco	This property specifies the context path of the Alfresco web application URL. The default value is <i>alfresco</i> . The context path is the path that is used by applications (for example, IMAP, SharePoint, and email) to access Alfresco. If you change this value, it must be defined with the same name as the Alfresco directory name specified by your application server. For example, if you are using Tomcat, this is the <code>/webapps/alfresco</code> directory in Tomcat, where <i>alfresco</i> is the name of the proxy server or specific server that you are using.
Repository Hostname	<code>\${localname}</code>	This property is the host name of the Alfresco web application that is used by external applications. Alfresco attempts to auto-detect the host name in place of <code>\${localname}</code> . If auto-detection fails, <code>\${localname}</code> is replaced with the IP address.
Server Allow Writes	true	Write access is permitted to the repository, as long as the Alfresco license is valid.
Protocol	http	This property is the protocol component of the Alfresco web application. The default is <i>http</i> . If you require HTTPS support you will need to configure this in the host application server.
Port	8080	This property is the port number of the Alfresco web application URL that is resolved by external applications. The default is <i>8080</i> .

4. Set the Server Settings properties:

Server Settings property	Example setting	What is it?
Allowed Users		This property allows you to specify which users can log in. By default, all users can log in. Enter a comma-separated list of users to allow only those users to log in. If you do not include the administrator user setting up this list (that is, the current user), then this will be added automatically.
Maximum Users	-1	The maximum number of simultaneous users allowed to log in. The default value -1 allows an unlimited number of users.

5. Set the Share Application Settings properties:

Share Application Settings property	Example setting	What is it?
Share Context	share	This property sets the context path of the Share web application URL. The default is share. You can set this context to a name that is appropriate for your instance of Alfresco.
Protocol	http	This property sets the protocol for the Share web application. The default is http. HTTPS support requires additional configuration within the host application server.
Share Hostname	127.0.0.1	This property sets the externally resolvable host name of the Share web application URL. The default value is <code>localhost</code> .
Port	8080	This property sets the externally resolvable port number of the Alfresco web application URL. The default is 8080.
Site Public Group	GROUP_EVERYONE	This property is the name of the group that controls user access to Public sites. The default is GROUP_EVERYONE, which contains all users.

6. Click **Save** to apply the changes you have made to the properties. If you do not want to save the changes, click **Cancel**.

Admin Console: Repository Services

The Repository Services section on the Admin Console contains pages for setting the Activities feed, the clustering tool, setting which workflow engine is in use, enabling replication, enabling and setting the search service, and controlling the Subscription and Transformation services.

 The repository Admin Console is for managing individual repository servers and must not be accessed through a load balancer.

Activities Feed

1. Open the Admin Console.
2. In the **Repository Services** section, click **Activities Feed**.

You see the **Activities Feed** page.

3. Set the activities properties:

Activities Feed property	Example setting	What is it?
Activity Feed Enabled	Yes	This enables or disables activity notifications to users using email.
Frequency CROM Expression	0 0 0 * * ?	This specifies a cron expression which defines the frequency with which users will receive Activities Feed emails. Emails are only sent if there are new activities since the last email. By default this is every 24 hours at midnight.
Maximum Number	100	The maximum number of activities that are reported on in the Activities dashlets and Activities Feed emails.
Maximum Age (mins)	44640	This is the maximum age of the activities shown in the Activities Feed emails. Activities older than the maximum age are not shown. The default setting is 44640 (a 31-day month).

4. Click **Save** to apply the changes you have made to the properties.
If you do not want to save the changes, click **Cancel**.

Admin Console: Repository server clustering

Servers connected to the same database instance are usually clustered automatically. In most cases no additional configuration is necessary.

1. Open the Admin Console.
2. In the **Repository Services** section, click **Repository Server Clustering**.

You see the **Repository Server Clustering** page.

3. Set the clustering properties:

For Host Server:

Clustering property	Example setting	What is it?
Server Name	ip-x-x-x-x	This specifies the name of the host server that you are currently connected to.
Cluster	Yes	This shows if clustering is enabled or disabled. You need to have a correct license to enable clustering.
IP Address	x.x.x.x	This specifies the IP address of the server.
Cluster ID	Yes	This specifies the unique id of the server.

For Cluster Members: Server Details

Clustering property	Example setting	What is it?
Server	ip-x-x-x-x	This specifies the server name of the cluster member.
IP	x.x.x.x	This specifies the IP address of the server.
Port	5701	This specifies the port number of the server.
Last Registered	02-Oct-2013 12:48:37	This specifies the date and time when the cluster member was last started.
Number of Members	1	This specifies the total number of members in the cluster.

For Offline Cluster Members: Server Details

Clustering property	Example setting	What is it?
Server	ip-x-x-x-x	This specifies the server name of a previously clustered server member that is no longer a member of the cluster.
IP	x.x.x.x	This specifies the IP address of the offline server.
Port	5701	This specifies the port number of the offline server.
Last Registered	02-Oct-2013 12:48:37	This specifies the date and time when the offline cluster server was last started.

- Click **Remove from list** to decommission a particular cluster member.
The offline cluster member no longer appears in the **Offline Cluster Members** list.
- Set the clustering properties for **Connected Non-Clustered Server(s)**:

In exceptional cases, an Alfresco server may be connected to the same database as other cluster members, and yet it may not be a member of the repository cluster. In other words, it will have clustering disabled. Such a server is called connected non-clustered server.

Clustering property	Example setting	What is it?
Server	ip-x-x-x-x	This specifies the name of the server.
IP	x.x.x.x	This specifies the IP address of the server.

6. To check if clustering is working properly, click **Validate Cluster**.
You see the **Cluster Validation** page. This page shows the validation results for a cluster.

Cluster validation performs a check to ensure that communication between the cluster members is working correctly. For a cluster to be considered validated, all cluster members should show success status. If one server fails in a two-server cluster, then both the servers will be marked as failed.

7. Click **Close**.

Admin Console: Enabling workflow process engines

In previous versions of Alfresco, a jBPM workflow engine was available. Although this process engine is still shipped with the installation, Alfresco recommends that you use the Activiti workflow engine for all new workflows.

In a new Alfresco installation, jBPM is disabled by default. If you have existing, migrated jBPM workflows that you wish to continue using, you must enable the jBPM workflow engine.

1. Open the Admin Console.
2. In the **Repository Services** section, click **Process Engines**.

You see the **Process Engines** page.

3. View the Activiti Workflow Engine properties:

Activiti Workflow Engine property	Example setting	What is it?
Activiti Workflow Enabled	enabled	Enables or disables the Activiti workflow engine. This workflow engine is enabled by default. When using only new workflows, you do not need to change any of the settings on this page.
Process Definitions Visible	enabled	Specifies whether the Activiti process definitions are available to users.

The other items in this section show the Activiti engine status details:

Activiti Workflow Engine status	Example setting	What is it?
Currently Running Process Instances	0	Specifies the number of Activiti process definitions running in the system.
Currently Running Task Instances	0	Specifies the number of Activiti-defined tasks running in the system.
Process Definitions Deployed	1	Specifies the number of Activiti process definitions deployed.

4. Enable the jBPM Workflow Engine for migrated workflows.

jBPM Workflow Engine property	Example setting	What is it?
jBPM Workflow Enabled	enabled	Enables or disables the jBPM workflow engine. This workflow engine is disabled by default. Set to enabled to continue using migrated jBPM workflows.
Workflow Definitions Visible	enabled	Specifies whether the jBPM workflow definitions are available to users.

5. For creating your own, more complex workflow definitions, click the **Activiti Workflow Console** link.

For more information on creating workflow definitions, see [Creating and managing workflows](#).

6. Click **Save** to apply the changes you have made to the properties.

If you do not want to save the changes, click **Cancel**.

Admin Console: Working with the replication service

The **Replication Service** page in Admin Console displays the settings to enable or disable the replication service and to control permissions.

The replication service allows content to be replicated (transferred) between distinct Alfresco repositories. For more information, see [Managing replication jobs](#).

Replication service property	Example setting	What is it?
Replication Enabled	disabled	Enables or disables the ability to replicate content from this repository.
Replicate Read Only	enabled	Enables or disables the permission settings for replicas in the target repository. The default setting is enabled, which sets the replicas as read-only. Replicas are normally read-only to enforce integrity. This option should only be disabled for specific use cases.

Working with the Search Service

The **Search Service** page in Admin Console enables you to manage and monitor the search service you want to use in Alfresco.

The Admin Console enables you to configure the Solr 4 search service using configuration properties.

Configuring the Solr 4 search service

1. Open the Admin Console.
2. In the **Repository Services** section, click **Search Service**.
You see the **Search Service** page.
3. In the **Search Service** section, select **Solr 4** from the **Search Service In Use** list.

4. Set the Solr 4 search service properties:

Solr search property	Example setting	What is it?
Content Tracking Enabled	Yes	This specifies that Solr 4 can still track with the No Index search enabled. This setting can be used to disable Solr 4 tracking by separate Solr instance(s) configured to track this server.
Solr Port (Non-SSL)	8080	This specifies the application server's http port (non-secure) on which Solr 4 is running. This is only used if Solr 4 is configured to run without secure communications.
Solr base URL	/solr4	This specifies the base URL for the Solr 4 web application.
Solr Hostname	localhost	This specifies the hostname on which the Solr 4 server is running. Use localhost if running on the same machine.
Solr SSL Port	8443	This specifies the application server's https port on which Solr 4 is running.
Auto Suggest Enabled	0	This specifies that the Solr 4 auto-suggest feature is enabled. This feature presents suggestions of popular queries as a user types their query into the search box or text box.
Indexing in Progress	No	This specifies if Solr 4 is currently indexing outstanding transactions.
Last Indexed Transaction	17	This specifies the transaction ID most recently indexed by Solr 4.
Approx Index Time Remaining	0 Seconds	This specifies the estimated time that Solr 4 will take to complete indexing the current outstanding transactions.
Disk Usage (GB)	0.001748	This specifies the disk space used by the latest version of the Solr 4 index. Allow at least double this value for background indexing management.
Index Lag	0 s	This specifies the time that indexing is currently behind the repository updates.
Approx Transactions to Index	0	This specifies the estimated number of outstanding transactions that require indexing.

Solr search property	Example setting	What is it?
Memory Usage (GB)	0	This specifies the current memory usage. The value may vary due to transient memory used by background processing.
Indexing in Progress	No	This specifies if Solr 4 is currently indexing outstanding transactions.
Last Indexed Transaction	17	This specifies the transaction ID most recently indexed by Solr 4.
Approx Index Time Remaining	0 Seconds	This specifies the estimated time that Solr 4 will take to complete indexing the current outstanding transactions.
Disk Usage (GB)	0.000034	This specifies the disk space used by the latest version of the Solr 4 index. Allow at least double this value for background indexing management.
Index Lag	0 s	This specifies the time that indexing is currently behind the repository updates.
Approx Transactions to Index	0	This specifies the estimated number of outstanding transactions that require indexing.
Memory Usage (GB)	0	This specifies the current memory usage. The value may vary due to transient memory used by background processing. The value does not include Lucene related caches.
Backup Location (Main Store)	<code>\${dir.root}/solr4Backup/alfresco</code>	This specifies the location where the index backup for the main WorkspaceStore is stored on the Solr 4 server.
Backup Cron Expression (Main Store)	0 0 2 * * ?	This specifies a unix-like expression, using the same syntax as the cron command, that defines when backups occur. The default value is 0 0 2 * * ? meaning the backup is performed daily at 02.00.
Backups To Keep (Main Store)	3	This specifies the number of backups to keep (including the latest backup).
Backup Location (Archive Store properties)	<code>\${dir.root}/solr4Backup/archive</code>	This specifies the location where the index backup for ArchiveStore is stored on the Solr 4 server.

Solr search property	Example setting	What is it?
Backup Cron Expression (Archive Store properties)	0 0 4 * * ?	This specifies a unix-like expression, using the same syntax as the cron command, that defines when backups occur. The default value is 0 0 4 * * ? meaning the backup is performed daily at 04.00.
Backups To Keep (Archive Store properties)	3	This specifies the number of backups to keep (including the latest backup).
CMIS Query	Use database if possible	This specifies the default mode which defines if and when the database should be used to support a subset of the CMIS Query Language.
Alfresco Full Text Search	Use database if possible	This specifies the default mode which defines if and when the database should be used to support a subset of the Alfresco Full Text Search.

- Click **Save** to apply the changes you have made to the properties.
If you do not want to save the changes, click **Cancel**.

Configuring No Index search service

- Open the Admin Console.
- In the **Repository Services** section, click **Search Service**.
You see the **Search Service** page.
- In the **Search Service** section, select **No Index** from the **Search Service In Use** list.
- Set the No Index search service properties:

Inbound Email property	Example setting	What is it?
Content Tracking Enabled	Yes	This specifies that Solr can still track with No Index search enabled. This setting can be used to disable Solr tracking by separate Solr instance(s) configured to track this server.
CMIS Query	Use database if possible	This specifies the default mode which defines if and when the database should be used to support a subset of the CMIS Query Language.
Alfresco Full Text Search	Use database if possible	This specifies the default mode which defines if and when the database should be used to support a subset of the Alfresco Full Text Search.

- Click **Save** to apply the changes you have made to the properties.
If you do not want to save the changes, click **Cancel**.

Admin Console: Enabling the subscription service

1. Open the Admin Console.
2. In the **Repository Services** section, click **Subscription Service**.
You see the **Subscription Service** page.
3. Use the **Enabled** check box to choose whether to enable or disable the Follow feature for all users:
 - Select the check box to enable subscriptions
 - Deselect the check box to disable subscriptions

The **Enabled** check box is selected by default. This allows users to follow other users and then filter activities according to who they are following. If you disable subscriptions, users will not be able to follow users and they will not see the activities. For example, on the **My Profile** page, the **I'm Following** and **Following Me** options are not visible.

4. Click **Save** to apply the changes you have made to the properties.
If you do not want to save the changes, click **Cancel**.

Admin Console: Transformation services

1. Open the Admin Console.
2. In the **Repository Services** section, click **Transformation Services**.
You see the **Transformation Services** page.
3. Set the Office Transform - JODConverter properties.

Property	Example setting	What is it?
JODConverter Enabled	No	This enables or disables the JODConverter for transformations.
Max Tasks per Process	200	This is the maximum number of tasks that can be performed concurrently.
Office Suite Location	/Applications/ alfresco-5.0.0/ libreoffice.app/Contents	This shows the directory path locations of OpenOffice.org or LibreOffice.
Port Numbers	8100	This is the port number that JODConverter uses. To enable multiple process instances, enter a comma-separated list of port numbers, all of which must be available.
Task Execution Timeout	120000	This is the duration in milliseconds after which a task will timeout.
Task Queue Timeout	30000	This is the duration in milliseconds after which a the task queue will timeout.

4. Click **Save** to apply the changes you have made to the properties.
If you do not want to save the changes, click **Cancel**.

Admin Console: Support Tools

The Support Tools section on the Admin Console contains the page for exporting the system information to a zip file.

Admin Console: Exporting system settings

1. Open the Admin Console.
2. In the **Support Tools** section, click **Download JMX Dump**.
You see the **Download JMX Dump** page.
3. Click **Export** and then click OK.
This will export the system information (JMX dump) and then download the zip file to your local machine.

Admin Console: Authentication directories

The Directories section on the Admin Console contains the page for defining authentication of Alfresco users and groups, including access to external directory services and setting up authentication chains.

Managing authentication directories

The Directory Management page provides an interface for you to:

- create, configure and manage internal Alfresco directories, OpenLDAP and Active Directory
- configure authentication chain options for services, such as CIFS and browser SSO
- test connections to various services before activating them in the authentication chain
- manage common user synchronization settings
- easily set up directory services for Alfresco without using property files

Managing the authentication chain

1. Open the Admin Console.
2. In the **Directories** section, click **Directory Management**.
You see the **Directory Management** page.
3. In the **Authentication Chain** section, specify the name of the new directory in the **Name:** field.
4. Specify the authentication subsystem type from the **Type:** menu.



If you have an **External** authentication type, the relevant directory will always appear as the first item in the chain.

5. Click **Add**.
The new authentication chain appears in the table.

The Authentication Chain table has the following fields:

- **Order:** Use the up and down arrows to reorder the authentication chain.
- **Name:** Specifies the name of the authentication chain.
- **Type:** Specifies the authentication subsystem type, such as OpenLDAP, Active Directory, Passthru, Kerberos, and External.
- **Enabled:** Specifies if authentication is enabled or not.
- **Synchronized:** Specifies if the authentication chain is synchronized or not.

- **Action:** Enables you to perform specific actions on the selected authentication chain, such as:
 - **Edit:** Enables you to configure the authentication directories. See [Managing authentication directories](#) for more information.
 - **Test:** Enables you to run an authentication test. To process the test request, you need a valid user name and password.
 - **Reset:** Enables you to reset the directory to its initial settings or default values. You will lose all changes you have made to this directory since it was created.
 - **Remove:** Removes the directory from the authentication chain list.
 - **Test synchronize:** Enables you to check if synchronization is configured correctly.
6. To manage the synchronization of Alfresco with all the user registries (LDAP servers) in the authentication chain, click **Synchronization Settings**.
You see the **Synchronization Settings** page. See [Synchronization Settings](#) for more information.
 7. To start the user directory sync of all users and groups, click **Run Synchronize**.
 8. Click **Save** to apply the changes you have made to the authentication chain.
If you do not want to save the changes, click **Cancel**.

Managing authentication directories

The authentication subsystem support certain properties that can be configured to integrate the subsystem with Alfresco. This topic describes how to manage the various subsystems using their configuration properties.

Click on the relevant authentication directory for more information.

Configuring OpenLDAP

This topic describes the instructions for configuring OpenLDAP using the configuration properties in the Admin Console.

1. Open the Admin Console.
2. In the **Directories** section, click **Directory Management**.
You see the **Directory Management** page.
3. In the **Authentication Chain** section, under **Actions**, click **Edit** corresponding to the OpenLDAP directory.
You see the **Edit LDAP Directory** page.
4. Set the configuration properties.

Synchronization property	Example setting	What is it?
Authentication Enabled	Yes	This specifies that the directory will be used to authenticate users.
User Name Format	-	This specifies how to map the user identifier entered by the user to that passed through to LDAP.

Synchronization property	Example setting	What is it?
LDAP Server URL	ldap:// openldap.domain.com:389	This specifies the URL of your LDAP server, containing its name and port. The standard ports for LDAP are 389 (and 636 for SSL)
Security	simple	This specifies the mechanism used to authenticate with the LDAP server. It should be one of the standard values provided here or one of the values supported by the LDAP provider. See LDAP configuration properties for more information.
Default Administrator User Names	-	This specifies a comma separated list of user names to be considered administrators by default. If you are using LDAP for all your users, this maps an LDAP user to be an administrator user.
Synchronization Enabled	Yes	This enables user and group synchronization. It might be that this connection should only be used for authentication, in which case this flag should be set to false.
Security Principal Name	cn=Manager,dc=company,dc=com	This specifies the LDAP user to connect for the export operation, if one is required by the <code>ldap.synchronization.java.naming</code> authentication mechanism. This should be in the same format as <code>ldap.authentication.userNameForm</code> but with a real user ID instead of %s.
Security	simple	This specifies the mechanism to use to authenticate with the LDAP Synchronization server. It should be one of the standard values provided here or one of the values supported by the LDAP provider. See LDAP configuration properties for more information.
Group query	(objectclass=groupOfNames)	This specifies the query to select all objects that represent the groups to export. This query is used in full synchronization mode, which by default is scheduled every 24 hours. The default is <code>(objectclass=groupOfNames)</code> .

Synchronization property	Example setting	What is it?
Security Principal Credentials	secret	This specifies the password for the default principal (only used for LDAP sync). Click Show Password to reveal the password. Click Hide Password to hide the password.
User Search Base	ou=People,dc=company,dc=com	This specifies the DN below which to run the user queries.
Group Search Base	ou=Groups,dc=company,dc=com	This specifies the DN below which to run the group queries.
Person Differential Query	(&(objectclass=inetOrgPerson)!(modifyTimestamp<={0}))	This specifies the query to select the objects that represent the users to export that have changed since a certain time. It should use the placeholder {0} in place of a timestamp in the format specified by <code>ldap.synchronization.timestampFormat</code> . This query is used in differential synchronization mode, which by default is triggered whenever a user, that does not yet exist in Alfresco, is successfully authenticated.
Person Query	(objectclass=inetOrgPerson)	This specifies the query to select all objects that represent the users to export. This query is used in full synchronization mode, which by default is scheduled every 24 hours.

 The **Edit LDAP Directory** page also displays certain advanced LDAP synchronization properties. It is recommended that you do not change these settings.

5. Click **Save** to apply the changes you have made to the OpenLDAP directory. If you do not want to save the changes, click **Close**.

Configuring LDAP (Active Directory)

This topic describes the instructions for configuring LDAP-AD using the configuration properties in the Admin Console.

1. Open the Admin Console.
2. In the **Directories** section, click **Directory Management**.
You see the **Directory Management** page.
3. In the **Authentication Chain** section, under **Actions**, click **Edit** corresponding to LDAP (Active Directory) directory.
You see the **Edit LDAP-AD Directory** page.
4. Set the configuration properties.

Synchronization property	Example setting	What is it?
Authentication Enabled	Yes	This specifies that the directory will be used to authenticate users.
User Name Format	-	This specifies how to map the user identifier entered by the user to that passed through to LDAP.
LDAP Server URL	ldap:// openldap.domain.com:389	This specifies the URL of your LDAP server, containing its name and port. The standard ports for LDAP are 389 (and 636 for SSL)
Security	simple	This specifies the mechanism used to authenticate with the LDAP server. It should be one of the standard values provided here or one of the values supported by the LDAP provider. See LDAP configuration properties for more information.
Default Administrator User Names	-	This specifies a comma separated list of user names to be considered administrators by default. If you are using LDAP for all your users, this maps an LDAP user to be an administrator user.
Synchronization Enabled	Yes	This enables user and group synchronization. It might be that this connection should only be used for authentication, in which case this flag should be set to false.
Security Principal Name	cn=Manager,dc=company,dc=com	This specifies the LDAP user to connect for the export operation, if one is required by the <code>ldap.synchronization.java.naming</code> authentication mechanism. This should be in the same format as <code>ldap.authentication.userNameForm</code> but with a real user ID instead of %s.
Security	simple	This specifies the mechanism to use to authenticate with the LDAP Synchronization server. It should be one of the standard values provided here or one of the values supported by the LDAP provider. See LDAP configuration properties for more information.

Synchronization property	Example setting	What is it?
Group query	(objectclass=groupOfNames)	This specifies the query to select all objects that represent the groups to export. This query is used in full synchronization mode, which by default is scheduled every 24 hours. The default is (objectclass=groupOfNames).
Security Principal Credentials	secret	This specifies the password for the default principal (only used for LDAP sync). Click Show Password to reveal the password. Click Hide Password to hide the password.
User Search Base	ou=People,dc=company,dc=com	This specifies the DN below which to run the user queries.
Group Search Base	ou=Groups,dc=company,dc=com	This specifies the DN below which to run the group queries.
Person Differential Query	(&(objectclass=inetOrgPerson)!(modifyTimestamp<={0}))	This specifies the query to select the objects that represent the users to export that have changed since a certain time. It should use the placeholder {0} in place of a timestamp in the format specified by ldap.synchronization.timestampFormat. This query is used in differential synchronization mode, which by default is triggered whenever a user, that does not yet exist in Alfresco, is successfully authenticated.
Person Query	(objectclass=inetOrgPerson)	This specifies the query to select all objects that represent the users to export. This query is used in full synchronization mode, which by default is scheduled every 24 hours.

 The **Edit LDAP Directory** page also displays certain advanced LDAP synchronization properties. It is recommended that you do not change these settings.

5. Click **Save** to apply the changes you have made to LDAP Active Directory.
If you do not want to save the changes, click **Close**.

Configuring Passthru

This topic describes the instructions for configuring Passthru using the configuration properties in the Admin Console.

1. Open the Admin Console.
2. In the **Directories** section, click **Directory Management**.
You see the **Directory Management** page.

- In the **Authentication Chain** section, under **Actions**, click **Edit** corresponding to the Passthru directory.

You see the **Edit Passthru Directory** page.

- Set the configuration properties.

Synchronization property	Example setting	What is it?
Use Local Server	No	This enables the local server to be used for passthru authentication by using loopback connections into the server.
Map Unknown User to Guest	No	This specifies whether unknown users are automatically logged in as the Alfresco guest user during SSO.
Allow Guest Login	No	This enables the guest logins to Alfresco.
Administrator User Names	-	This specifies a comma separated list of user names to be considered administrators by default.
Authenticate FTP	Yes	This enables passthru authentication for FTP access.
Authenticate Domain	DOMAIN	This specifies the Windows NetBIOS domain name to use for passthru authentication. This will attempt to find the domain controllers using a network broadcast. If the network broadcast is not successful, use the <code>passthru.authentication.servers</code> property to specify the domain controller list by name or address.
Authentication Servers	-	This specifies a comma delimited list of server names or addresses that are used for authentication. The pass through authenticator will load balance amongst the available servers, and can monitor server online/offline status.
Authentication Protocol Order	TCPIP,NetBIOS	This specifies the type of protocols and order of connection for passthru authentication sessions. The default is to use NetBIOS, and the available protocol types are NetBIOS for NetBIOS over TCP and TCPIP for native SMB.

Synchronization property	Example setting	What is it?
Connection Timeout	5000	This specifies the timeout value in milliseconds when opening a session to an authentication server. The default is 5000.
Offline Check Interval	300	This specifies how often (in seconds) the passthru servers that are marked as offline are checked to see if they are now online. The default check interval is 5 minutes.

- Click **Save** to apply the changes you have made to the Passthru directory.
If you do not want to save the changes, click **Close**.

Configuring Kerberos

This topic describes the instructions for configuring Kerberos using the configuration properties in the Admin Console.

- Open the Admin Console.
- In the **Directories** section, click **Directory Management**.
You see the **Directory Management** page.
- In the **Authentication Chain** section, under **Actions**, click **Edit** corresponding to the Kerberos directory.
You see the **Edit Kerberos Directory** page.
- Set the configuration properties.

Synchronization property	Example setting	What is it?
User Config Entry Name	Alfresco	This specifies the entry in the JAAS configuration file that should be used for password-based authentication. The recommended default value is Alfresco.
Administrator User Names	-	This specifies a comma separated list of user names to be considered administrators by default.
CIFS Config Entry Name	AlfrescoCIFS	This specifies an entry in the JAAS configuration file that should be used for CIFS authentication. The recommended default value is AlfrescoCIFS.
Kerberos Authentication Realm	ALFRESCO.ORG	This specifies the Kerberos realm used for authentication. The realm should be the domain in upper case. For example, if the domain is 'alfresco.org', then the realm should be ALFRESCO.ORG.

Synchronization property	Example setting	What is it?
CIFS Password	secret	This specifies the password for the CIFS Kerberos principal. Click Show Password to reveal the password. Click Hide Password to hide the password.
HTTP Config Entry Name	AlfrescoHTTP	This specifies the entry in the JAAS configuration file used for web-based SSO. The recommended default value is AlfrescoHTTP.
Strip Username Suffix	Yes	This specifies that the @domain suffix is stripped from Kerberos authenticated user names in CIFS, SPP, WebDAV, and the Web Client. If not selected, multi-domain users can use the @domain suffix.
HTTP Password	secret	This specifies the password for the HTTP Kerberos principal. Click Show Password to reveal the password. Click Hide Password to hide the password.

- Click **Save** to apply the changes you have made to the Kerberos directory.
If you do not want to save the changes, click **Close**.

Configuring external authentication

This topic describes the instructions for configuring external authentication using the configuration properties in the Admin Console.

- Open the Admin Console.
- In the **Directories** section, click **Directory Management**.
You see the **Directory Management** page.
- In the **Authentication Chain** section, under **Actions**, click **Edit** corresponding to the External directory.
You see the **Edit External Directory** page.
- Set the configuration properties.

Synchronization property	Example setting	What is it?
Authentication Enabled	Yes	This enables the external directory user authentication. When enabled, Alfresco accepts external authentication tokens; ensure that no untrusted direct access to Alfresco's HTTP or AJP ports is allowed.

Synchronization property	Example setting	What is it?
Proxy Username	alfresco-system	This specifies the remote user that is considered as the proxy user. The default is alfresco-system.
Administrator User Names	-	This specifies a comma separated list of user names to be considered administrators by default.
Proxy Header	X-Alfresco-Remote-User	This specifies the HTTP header that carries the name of a proxied user. The default is X-Alfresco-Remote-User.
User ID Pattern	-	This specifies an optional regular expression used to extract a user ID from the HTTP header. The portion of the header matched by the first bracketed group in the regular expression becomes the user name. If not set, the entire header contents are assumed to be the proxied user name.

5. Click **Save** to apply the changes you have made to the External authentication directory. If you do not want to save the changes, click **Close**.

Configuring alfrescoNtlm

This topic describes the instructions for configuring alfrescoNtlm using the configuration properties in the Admin Console.

1. Open the Admin Console.
2. In the **Directories** section, click **Directory Management**.
You see the **Directory Management** page.
3. In the **Authentication Chain** section, under **Actions**, click **Edit** corresponding to the alfrescoNtlm1 directory.
You see the **Edit Internal Alfresco Directory** page.
4. Set the configuration properties.

Synchronization property	Example setting	What is it?
Allow Guest Login	Yes	This enables guest access to Alfresco.
Map Unknown User to Guest	alfresco-system	This enables unknown users to automatically log in as the Alfresco guest user during SSO.

5. Click **Save** to apply the changes you have made to the Alfresco Internal authentication directory.
If you do not want to save the changes, click **Close**.

Managing synchronization settings

The synchronization settings manage the synchronization of Alfresco with all the user registries (LDAP servers) in the authentication chain. This topic describes how to configure the synchronization subsystem.

1. Open the Admin Console.
2. In the **Directories** section, click **Directory Management**.
You see the **Directory Management** page.
3. Under the **Authentication Chain** section, click **Synchronization Settings**.
You see the **Synchronization Settings** page.
4. Set the synchronization properties.

Synchronization property	Example setting	What is it?
Sync on Startup	Yes	This triggers synchronization when the subsystem starts up. This ensures that when the user registries are first configured, bulk of synchronization work is done on server startup, rather than on the first login.
Sync When Missing People Login	Yes	This triggers synchronization when a user, who does not yet exist in Alfresco, is successfully authenticated. The default is true.
Allow Deletions	Yes	This triggers deletion of the local users and groups during synchronization when handling removals or collision resolution. The default is true. If false, then no sync job will be allowed to delete users or groups during the handling of removals or collision resolution.
Logging Interval	100	This specifies the number of user or group entries processed during synchronization before the progress is logged at INFO level. It requires the following default entry in <code>log4j.properties</code> : <code>log4j.logger.org.alfresco.repo.s</code> The default is 100.

Synchronization property	Example setting	What is it?
Auto Create People On Login	Yes	This specifies whether to create a user with default properties, when a user is successfully authenticated, who does not yet exist in Alfresco, and was not returned by synchronization (if enabled with the Sync When Missing People Login property). The default is true.
Sync Changes Only	Yes	This triggers a differential synchronization. Deselect this option, to run full synchronization. Regardless of this setting, a differential synchronization can still be triggered when a user, who does not yet exist in Alfresco, is successfully authenticated.
Import CRON Expression	0 0 0 * * ?	This specifies a cron expression which defines when the scheduled synchronization job should run. By default, this is every 24 hours at midnight.
Sync Worker Threads	1	This specifies the number of worker threads used for synchronization. The default is 1.

 Settings are common to all the directories for which synchronization is enabled.

- Click **Save** to apply the changes you have made to the authentication chain.
If you do not want to save the changes, click **Close**.

Managing CIFS authentication

- Open the Admin Console.
- In the **Directories** section, click **Directory Management**.
You see the **Directory Management** page.
- In the **CIFS Authentication** section, select a directory from the list to authenticate CIFS. Alternatively, select **Disabled** to disable CIFS authentication.

 CIFS uses a challenge or response to authenticate. Only a single directory can be used to authenticate.

- Click **Save** to apply the changes you have made to the authentication chain.
If you do not want to save the changes, click **Cancel**.

Managing browser based automatic login

- Open the Admin Console.
- In the **Directories** section, click **Directory Management**.
You see the **Directory Management** page.

3. In the **Browser Based Automatic Login** section, select a directory to automatically log users by using a browser. Alternatively, select **Disabled** to disable automatic login.
 -  You can configure other forms of SSO using the external authentication type, such as CAS or Siteminder.
4. Click **Save** to apply the changes you have made to the authentication chain. If you do not want to save the changes, click **Cancel**.

Admin Console: Virtual File Systems

The Virtual File Systems section on the Admin Console contains pages for setting up access to Alfresco from the CIFS and FTP protocols.

Enabling file servers

1. Open the Admin Console.
2. In the **Virtual File Systems** section, click **File Servers**.
You see the **File Servers** page.
3. Set the File Systems properties:

File Systems property	Example setting	What is it?
File System Name	Alfresco	The name given to the file system when using CIFS, WebDAV, or FTP.

4. Set the CIFS properties:

CIFS property	Example setting	What is it?
CIFS Enabled	Yes	This enables or disables the CIFS server.
Server Name	\${localname}A	The Alfresco CIFS server host name. This can be a maximum of 16 characters and must be unique on the network. You can use the special token \${localname} in place of the local server's host name and generate a unique name by prepending/appending to it.
Host Announce	Yes	Enables the announcement of the CIFS server to the local domain/workgroup so that it shows up in the Network Places/Network Neighborhood.
Session Timeout (seconds)	900	The default CIFS session timeout is 15 minutes. If no I/O occurs on the session within this time then the session will be closed by the server. Windows clients send keep-alive requests, usually within 15 minutes.

CIFS property	Example setting	What is it?
Domain		The domain or workgroup to which the server belongs. If not specified then the domain/workgroup of the server is used.

- Set the FTP properties:

FTP property	Example setting	What is it?
FTP Enabled	Yes	This enables or disables the FTP server.
Port	2121	This specifies the port on which the FTP server listens for connections.
Dataport From		This specifies the lower limit of the range of data ports.
Dataport To		This specifies the upper limit of the range of data ports.

- Click **Save** to apply the changes you have made to the properties.
If you do not want to save the changes, click **Cancel**.

Enabling the IMAP Service

- Open the Admin Console.
- In the **Virtual File Systems** section, click **IMAP Service**.
You see the **IMAP Service** page.
- Set the IMAP Service properties:

IMAP Service property	Example setting	What is it?
IMAP Server Enabled	No	This enables or disables the IMAP server.
Hostname	0.0.0.0	This specifies the host or IP address to which the IMAP service will bind.
Mail TO Default	alfresco@demo.alfresco.org	This specifies the default TO field that will be used when the TO field is not available, for example, when displaying documents.
Mail FROM Default	alfresco@demo.alfresco.org	This specifies the default FROM field that will be used when the FROM field is not available, for example, when displaying documents.

- Set the IMAP Protocol properties:

IMAP Protocol property	Example setting	What is it?
Enable IMAP	Yes	This enables or disables the IMAP service.

IMAP Protocol property	Example setting	What is it?
Port	143	This specifies the port number on which this service will listen. This is usually 143 but can be changed to an alternative number.

5. Set the IMAPS Protocol properties:

IMAPS Protocol property	Example setting	What is it?
Enable IMAP	Yes	This enables or disables the IMAPS service.
Port	993	This specifies the port number on which this service will listen. This is usually 993 but can be changed to an alternative number.

6. Click **Save** to apply the changes you have made to the properties.
If you do not want to save the changes, click **Cancel**.

Starting the Activiti workflow console

Alfresco provides the Activiti workflow console for managing Activiti based workflows and process definitions.

To start the Activiti workflow console:

1. [Launch the Admin console](#)
2. In the **Repository Services** section, click **Process Engines**
3. In the bottom right panel, click **Activiti Workflow Console**

Customizing the Alfresco Admin Console

The Alfresco Admin Console displays the most common Alfresco administration activities. You can customize the Admin Console to show different options, properties, and layout, or you can create completely new pages.

The Admin Console is composed of default administration pages. Each Admin Console page is a simple web script component built from a library of useful functions and macros that are imported into each Admin Console web script.

The JavaScript library functions do the background work for the Admin Console, retrieving the JMX MBean properties and then transferring them to flexible FreeMarker macros. The FreeMarker macros render the appropriate control for a JMX property automatically and consistently.

If no additional processing logic is required, the web script library functions automatically persist them back to the correct property.

JMX form-style pages are simple to build. Example pages that you can create include: Thread Dump, Active Sessions, Log4J settings, and Test Transforms.

Alfresco Admin Console Example page

When you customize the Alfresco Admin Console, you can use the example page as a starting point.

The Admin Console example page is called `admin-example` and contains comments to help you to understand the code.

The files that you use for working with the example Admin Console page are:

- [admin-example.get.js](#)
- [admin-example.get.html.ftl](#)
- [admin-example.get.desc.xml](#)
- [admin-example.get.properties](#)

See the [Web script components](#) section for more information on these files.

There are also additional properties files that contain the associated strings for localized content in the supported languages.

The following snippet shows the controller code from the `admin-example.get.js` file, which retrieves the `Subject`, `Issued`, and `RemainingDays` properties from the `License` JMX bean:

```
<import resource="classpath:alfresco/enterprise/webscripts/.../admin-
common.lib.js">
/* Repository Admin Console - Example GET method */
Admin.initModel(
    "Alfresco:Name=License",
    ["Subject", "Issued", "RemainingDays"],
    "admin-example"
);
```

The following snippet shows the template code from the `admin-example.get.html.ftl` file:

```
<#include "admin-template.ftl" />
<@page title="Example Page">
  <div class="column-left">
    <@section label="Some Values" />
    <@control attribute=attributes["Subject"] />
  </div>
  <div class="column-right">
    <@section label="More Values" />
    <@control attribute=attributes["Issued"] />
    <@control attribute=attributes["RemainingDays"] />
  </div>
</@page>
```

The resulting output from the `admin-example` web script displays the following:



Some Values

Subject: Enterprise - v4.2

More Values

Issued: 05-Nov-2012 15:21:07

RemainingDays: -1

The values from the `License` JMX bean are read-only. The template macros understand when the JMX beans are read-only, and therefore, display the text as read-only.

When the JMX beans are editable or if you want to show a different form field, add the following line to change the template:

```
<@attrtext attribute=attributes["Subject"] />
```

The resulting output then displays the following:



Some Values

Subject:

More Values

Issued: 05-Nov-2012 15:21:07

RemainingDays: -1

Using the Share Admin Tools

Share Admin Tools let you manage your administration operations.

Admin Tools enables Alfresco administrators to create and manage users and groups from within Share, set application preferences, manage categories and tags, and browse the system information in the node browser.

 Some of the tools previously found in Share Admin Tools have been moved to the Repository Administration Console, which is available in Alfresco Enterprise only.

The Admin Tools option is visible on the menu bar only if you are an Administrator user or a user who is a member of the `ALFRESCO_ADMINISTRATORS` group.

Opening the Share Admin Tools

You can only see the **Admin Tools** option on the menu bar if you are an administrator user or a user who is a member of the `ALFRESCO_ADMINISTRATORS` group.

1. Click **Admin Tools**.

You see the list of tools on the left-side of the page. The first set of tools are for general Alfresco administration:

- **Application**
- **Category Manager**
- **Node Browser**
- **Tag Manager**
- **Sites Manager**

The remaining tools are grouped into the following categories:

- **Repository**
 - **Replication Jobs**
- **Users and Groups**
 - **Groups**
 - **Users**

2. Select an Admin Tool from the left side to see the page for each tool.

Application settings

Use the Application tool in the **Admin Tools** to set application settings.

Changing the theme

The look and feel of the user interface is set by a theme. The Application tool lets you select a color scheme for the user interface.

1. Click **Admin Tools**, and then click **Application**.
2. On the **Options** page, select a theme from the list.

Choose one of the available themes:

- **Green Theme**
- **Blue Theme**
- **Light Theme**
- **Yellow Theme**
- **Google Docs Theme**
- **High Contrast Theme**

3. Click **Apply**.

The new theme applies the CSS and image assets across all pages.

The page refreshes to display with the selected theme. The changed theme affects all users from the next time they log in and persists across sessions.

A new installation uses the default theme.



Site managers can customize the theme for an individual site. If a site theme has been changed, this will override any theme setting made in the **Admin Tools**.

Changing the logo

The Alfresco logo on the top left is at the top left-side. You can change the logo to another image file.

1. Click **Admin Tools**, and then click **Application**.

2. On the **Options** page, click **Upload**.

You'll see the **Upload File** window.

3. Click **Select files to upload**.

4. Choose a file and click **Open**.

You can choose to upload any image you like but there are some recommendations for suitable sizes for the image. The maximum recommended image height for your image file is 48 pixels.

The file you chose shows in the **Upload File** window. If it's not the right file, click **Remove** to select another file.

5. Click **Upload File(s)**.

6. When you see that the file is successfully uploaded, click **OK**.

7. Click **Apply**.

The newly uploaded file now becomes the logo for Alfresco.

8. If you wish to change the logo back to the default Alfresco logo, click **Reset** to display the original logo, and then click **Apply**.

Managing categories

Manage your categories on the **Category Manager** page.

1. Click **Admin Tools**, and then click **Category Manager**.

The **Category Manager** page shows a tree structure of the categories created in the system. The top level is called **Category Root** and by default, the following sub-categories are listed:

- **Languages**
- **Regions**
- **Software Document Classification**
- **Tags**

2. Click the category icons () to expand the list of categories.

When you hover over the category name, you see the available action icons for: **Edit category** (), **Add category** (), and **Delete category** ().

3. To edit a category, click the **Edit Category** icon, edit the category name inline, and then click **Save**.
4. To add a category, click the **Add Category** icon, enter a name in the **Category name** field, and then click **OK**.
When using Solr, there maybe a delay before the new category appears in a search query until after Solr has been reindexed. Categories are eventually consistent.
5. To delete a category, click the **Delete Category** icon, and then click **Delete** to confirm that you wish to delete the category.

The category is deleted from the system. Any content is removed from that category label.

Using the Node Browser

The Node Browser is a debugging aid that allows you to browse the raw Alfresco repository structure. This is a read-only feature with basic search capability, and is intended for developers responsible for customizing the application.

1. Click **Admin Tools**, and then click **Node Browser**.

By default, the search criteria `PATH: "/"` is shown in the Node Browser field for the `workspace://SpacesStore` repository store. Each store is an area of the repository. The nodes contained within each store are organized hierarchically. The node displayed is the root node of the selected store.

The default search type is set to **fts-alfresco**. For most administrative tasks, you can use the default search type. See [Alfresco Full Text Search reference](#) for more detail.

2. Enter your search criteria in the Note Browser field.
3. Click **Search**.
4. Click the link in the **Reference** column to browse the details.

The details of the properties, aspects, children, parents, associations, source associations, and permissions are displayed for the node.

5. Click **Back to Search** to browse another node.

You can use another search syntax by choosing one of the following types from the **Search** list:

- **storerooot**
- **noderef**
- **xpath**
- **fts-alfresco**

- **cmis-strict**
- **cmis-alfresco**
- **db-afts**
- **db-cmis**

Managing tags

Tags can be added to content within the Document Library. Use the **Tag Manager** page to view, edit, and delete all the tags that have been created by users.

1. Click **Admin Tools**, and then click **Tag Manager**.

The **Tag Manager** page shows a list of the tags that have been created, the name of the user who created or modified the tag, and the date on which the change was made.

If there are no tags in the system, you see the message: **No tags found**.

When you hover over the right hand **Actions** column, you see the available action icons

for: **Edit tag** () and **Delete tag** (.

- a. To edit a tag, click the **Edit tag** icon, edit the tag name in the **Rename Tag** field, and then click **OK**.
- b. To delete a tag, click the **Delete tag** icon, and then click **Delete** to confirm that you wish to delete the tag.

The tag is deleted from the system and removed from any content where it was previously tagged.

2. Click the tag name to see a list of the repository content that uses this tag.
3. Click the user name to see the profile of the user who last modified the tag.

Sites Manager

The Sites Manager is a great tool for maintaining Alfresco sites. You have control over the visibility of all Alfresco sites as well as deleting sites or making yourself a site manager.



Sites Manager is available to users in the `ALFRESCO_ADMINISTRATORS` and `SITES_ADMINISTRATORS` permissions groups.

If you've have the necessary permissions then you'll have an additional **Sites Manager** option on the Alfresco toolbar.

Members of the `ALFRESCO_ADMINISTRATORS` group access the Sites Manager through the **Admin Tools** on the Alfresco toolbar.

The Sites Manager displays the names and status of created sites, regardless of their visibility setting. You can use the **Visibility** menu to change the visibility of any site, for example, change the site visibility to either **Public**, **Moderated**, or **Private**. Any visibility change you make to a site is made immediately.

With the **Actions** menu, there are two options:

- **Delete Site**
- **Become Site Manager**

You can delete any of the sites in the Site Manager list by selecting **Delete Site** from the **Actions** menu. This action deletes all site details and content.

The **I'm a Site Manager** column shows the sites where you have the Site Manager permission. If you aren't already a manager of a site, then select **Become Site Manager** from the **Actions** menu.

Managing replication jobs

The Replication Jobs tool enables you to create and manage jobs for content replication.

A replication job specifies the content to be replicated; the day and time the job is to be performed; and the target location for the replicated content.

The job is controlled by the Replication Service, and it calls the Transfer Service, which allows folders and content to be automatically copied between Alfresco repositories. A replication job can be run according to a schedule or on-demand.

By default, any replicated content is read-only in the target repository. This ensures the integrity of the content is not affected by uncontrolled updates.

Viewing a replication job

Select a replication job to view the job details and display the available actions.

1. Click **Admin Tools**, and then click **Replication Jobs**.

The Replication Jobs page displays a summary of recently run jobs and a list of existing replication jobs. In this list, use the menu provided to sort the jobs by Status, Name, and Last Run Date.

2. In the Jobs section, click a job to view its details.

The job appears highlighted in the list and its details appear on the right side of the page.

Creating a new replication job

You can create any number of replication jobs to suit your needs.

1. Click **Admin Tools**, and then click **Replication Jobs**.
2. In the Jobs section, click **Create Job**.

The **Create New Replication Job** page appears. Fields marked with an asterisk (*) are required.

3. Enter the details for the new replication job.

- a. Enter a name for the job, and enter a description, if required.
- b. In the Payload section, click **Select**.

Navigate the repository and click **Add** to the right of each space that you want to include in the payload. This content will be replicated (copied) when the job is run. Click **OK**.

- c. In the Transfer Target section, click **Select**.

Navigate the Transfer Target Groups and click **Select** to the right of the target. Click **OK**.

 Out of the box, one target group, **Default Group**, is available. Create additional target groups in **Data Dictionary > Transfers > Transfer Target Group**. A rule defined on the Transfer Target Groups folder specializes the type of any folder created within it.

See [Creating a new transfer target for content replication](#) for more information.

- d. Specify when you want the replication job to run.

Select the **Schedule job** check box, then enter the date and time the job is to run. Specify the repeat period for this job.

- e. Select the **Enabled** check box to enable to replication job to run.

 You must enable a replication job for it to be run.

4. Click **Create Job**.

The job created appears highlighted in the Jobs list. The job details appear on the right side of the page.

Managing existing jobs

The Replication Jobs page in **Admin Tools** displays a list of all existing replication jobs.

For each job in this list, you can perform any of the following actions to manage and maintain the jobs:

- Run a job
- Cancel a job
- Edit a job
- Delete a job

Running a replication job

The **Run Job** tool allows you to run a replication job. You can do this at any time. If a schedule is set for the job, it remains in place and will be run at the appropriate time.

1. Click **Admin Tools**, and then click **Replication Jobs**.
2. In the Jobs section, click the job that you want to run.

The job appears highlighted in the list and its details appear on the right side of the page.

 For a job to be run, it must be enabled.

3. Click **Run Job**.

The Status section on the right side of the page indicates that the job is running. The date and time the job started is displayed.

Canceling a replication job

You can cancel a job that is currently running, regardless of whether it was started automatically (that is, it is a scheduled job) or manually.

1. Click **Admin Tools**, and then click **Replication Jobs**.
2. In the Jobs section, click the currently running job that you want to cancel.

An icon () to the left of the job name indicates a job is currently running.

The Status section on the right side of the page indicates the start time of the selected job.

 If the job was already displayed, you might need to click **Refresh** to update the status.

3. Click **Cancel Job**.

The job is stopped and a report is created.

Editing a replication job

You can easily update existing replication jobs. In addition to changing the job details, you can use this feature to disable a job so that it will not be run.

1. Click **Admin Tools**, and then click **Replication Jobs**.
2. In the Jobs section, click the job you want to edit.

The job appears highlighted in the list and its details appear on the right side of the page.

3. Click **Edit**.

The Edit Replication Job page appears.

4. Edit the replication job as necessary. All job details—name, description, payload, transfer target, and schedule—are available for editing.

Add and remove source items as necessary. Click **Remove** to the right of a single item to remove it. Click **Remove All** beneath the list to remove all items.

Deselect the **Enabled** check box to prevent the job from being run.

5. Click **Save**.

The main page displays the updated job details.

Deleting a replication job

If you no longer need a replication job, you can delete it from the Jobs list. If there is a chance you might need the job again, you might prefer to edit the job and simply disable it.

1. Click **Admin Tools**, and then click **Replication Jobs**.
2. In the Jobs section, click the job you want to delete.

The job appears highlighted in the list and its details appear on the right side of the page.

3. Click **Delete**.

A message prompts you to confirm the deletion of the selected job.

4. Click **Delete**.

The selected job is deleted from the jobs list.

Viewing replication job reports

Two reports—local and remote—are available for each replication job run successfully.

The local report is the transfer report from the sending system, which manages the content being transferred to the receiving system. The local report details the speed at which the files were transferred and other related details.

The remote report is the transfer report from the receiving system. This report indicates whether files were created, updated, modified, or deleted as part of the transfer.

1. Click **Admin Tools**, and then click **Replication Jobs**.
2. In the Jobs section, click the job you want to view.

The job appears highlighted in the list and its details appear on the right side of the page.

3. Select the desired report:

- Click **View Local Report**.
- Click **View Remote Report**.

The selected report displays on the details page of the Repository Document Library.

Managing users

The Users tool lets you create and manage the user accounts.

Creating a new user

Create user accounts with the **Users** option.

1. Click **Admin Tools**, and then click **Users**.

You'll see the **User Search** page.

2. Click **New User**.

The **New User** page appears. Fields marked with an asterisk (*) are required.

3. Complete all the required user fields.

Field	What is it?
First Name	Type the first name of the new user.
Email	Type an email address that the user will use for receiving Alfresco notification emails.
User Name	Type a user name for the new user.
Password	Type a password for the user account.  Enter a minimum of five characters otherwise you'll not be able to see the Create User button.
Verify Password	Repeat the password. Make sure that you type the same password you typed in the Password field.

4. Add the user to existing user groups:
 - a. In the search box, type the full or partial name of the desired group.
You must enter a minimum of one (1) character. The search is not case sensitive.
 - b. Click **Search**.
 - c. In the list of returned results, click **Add** to the right of each group you want the user to be added to.
The groups appear beneath the **Groups** list. Click a group to remove it.
 - d. Perform additional searches as necessary to locate and add more groups.

5. In the **Quota** box, specify the maximum space available for this user and select the appropriate unit (GB, MB, or KB).

This information is not required. When no quota is provided, the user has no space limitations.

Content quotas are disabled by default. You can change the default setting by adding the following property to the `alfresco-global.properties` file:
`system.usages.enabled=true`.

6. Click **Create User**.

 The create buttons are not available until you complete all the required fields. If you didn't type in matching passwords, you'll see a message to say that the password fields do not match.

If you intend to immediately create another user, click **Create and Create Another**. This creates the user account specified and clears the fields without returning you to the **User Search** page.

Uploading multiple users

Use the Users tool to upload externally created users from within a comma-separated (CSV) file.

When initially setting up the accounts for your users, it can be time consuming to create multiple users individually. Alfresco lets you create these users by uploading a file that contains the list of all your users. The file needs to contain the names and other details, separated but commas.

You can create this file, either from a text file or from a Microsoft Office spreadsheet. You need to create the file using named headings and the following order:

```
User Name,First Name,Last Name,E-mail Address,,Password,Company,Job
Title,Location,Telephone,Mobile,
```

Skype,IM,Google User Name,Address,Address Line 2,Address Line 3,Post Code,Telephone,Fax,Email

You don't need values for all the headings for each users. For example, the following sample shows the content of a CSV file using Microsoft Excel:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	User Name	First Name	Last Name	E-mail Address		Password	Company	Job Title	Location	Telephone	Mobile	Skype	IM	Google User Name
2	maltos	Matthew	Altos	maltos@alfresco.com			Alfresco	Office Staff	Maidenhead	1628876500	1628876500	maltos12345		maltos@gmail.com
3	dcare	Danny	Care	dcare@alfresco.com			Alfresco	Office Staff	Maidenhead	1628876500	1628876500	dcare123456		dcare@gmail.com
4	daybar	Darryl	Aybar	daybar@alfresco.com			Alfresco	Office Staff	Maidenhead	1628876500	1628876500	daybar12345		daybay@example.com
5	bingham	Brenda	Ingham	bingham@alfresco.com			Alfresco	Office Staff	Maidenhead	1628876500	1628876500	bingham12345		bingham@gmail.com
6														

Save the file as a .csv file, which you can then upload into Alfresco.

```
User Name,First Name,Last Name,E-mail Address,,Password,Company,Job
Title,Location,Telephone,Mobile,Skype,IM,Google User Name,Address,Address
Line 2,Address Line 3,Post Code,Telephone,Fax,Email
maltos,Matthew,Altos,maltos@alfresco.com,,Alfresco,Office Staff,Maidenhead,
1628876500,1628876500,maltos12345,,maltos@gmail.com,,,,,
dcare,Danny,Care,dcare@alfresco.com,,Alfresco,Office Staff,Maidenhead,
1628876500,1628876500,dcare123456,,dcare@gmail.com,,,,,
daybar,Darryl,Aybar,daybar@alfresco.com,,Alfresco,Office Staff,Maidenhead,
1628876500,1628876500,daybar12345,,daybay@example.com,,,,,
bingham,Brenda,Ingham,bingham@alfresco.com,,Alfresco,Office Staff,Maidenhead,
1628876500,1628876500,bingham12345,,bingham@gmail.com,,,,,|
```

1. Click **Admin Tools**, and then click **Users**.
You'll see the **User Search** page.
2. Click **Upload User CVS File**.
3. Locate and upload the CSV file:
 - a. Click the Select file(s) to upload icon.
 - b. Browse for the CSV file containing the users.
The CSV file has an extension of .csv.
 - c. Select the file, and then click **Open**.
 - d. Click **Upload File(s)**.

The users from the CSV file are uploaded into Alfresco and you see the **Upload Results** page showing the list of user names and status. An email will be sent to the user informing them of their new Alfresco user account.

Searching for and viewing a user account

The User Search tool lets you locate any user and view that user's account information.

1. Click **Admin Tools**, and then click **Users**.
You see the **User Search** page.
2. In the search box, enter the full or partial name of the user.
The search is not case sensitive.
3. Click **Search**.
In the results table, you can click the column headings to sort the results.
In the first column, a green dot indicates the user account is currently enabled; a red dot indicates the account is disabled.
4. Click the name of a user to show the related user profile and account details.

You see the **User Profile** page. From here you can edit or delete the user account.

Editing a user account

Edit a user account to change a user's personal information, group affiliation, quota, and password.

1. Click **Admin Tools**, and then click **Users**.
You'll see the **User Search** page.
2. Search for a user, and then select the user.
3. On the **User Profile** page, click **Edit User**.
The **Edit User** page appears.
4. Edit the user's personal details as necessary: **First Name** and **Email**.
5. Edit the groups to which this user belongs:
 - a. To add a user to a group, use the search field provided to locate a group. Click **Add** to the right of each group you want the user to be a part of. The groups the user belongs to display beneath the **Groups** list.
 - b. To remove a user from a group, simply click the group you want to remove beneath the **Groups** list.
6. Provide or edit the **Quota**, which indicates the maximum space available for this user. Select the appropriate unit.
7. Change the password, if necessary.
8. Click **Use Default** to reset the user's picture to the default image.
9. Click **Save Changes**.

Deleting a user account

Delete a user account to remove the user from the system.

 Deleting a user does not remove their permissions from the repository. These permissions are reused if the user is recreated in the future. To keep an account but stop the user having access to the application, consider disabling the user account instead.

1. Click **Admin Tools**, and then click **Users**.
You see the **User Search** page.
2. Search for a user, and then select the user.
3. On the **User Profile** page, click **Delete User**.
A message prompts you to confirm that you want to delete the user account.
4. Click **Delete**.

Disabling a user account

Disable a user account to prevent a user from having any access to the application. You perform this task as part of editing a user account.

1. Click **Admin Tools**, and then click **Users**.
You see the **User Search** page.
2. Search for a user, and then select the user.
3. On the **User Profile** page, click **Edit User**.
You see the **Edit User** page.
4. Click **Disable Account**.
A check mark indicates the account for the current user will be disabled.

5. Click **Save Changes**.

On the **User Profile** page, the Account Status shows as **Disabled**. On the **User Search** page, the user displays in the search results list with a red dot, indicating the account is disabled.

Changing a user's password

You can change a user's password as part of editing the user account.

1. Click **Admin Tools**, and then click **Users**.

You see the **User Search** page.

2. Search for a user, and then select the user.

3. On the **User Profile** page, click **Edit User**.

You see the **Edit User** page.

4. Enter and confirm the new password for this user in the **New Password** and **Verify Password** boxes.

The password is case sensitive.

5. Click **Save Changes**.

Managing the user's group membership

Within a user account, you can manage the user's membership in existing user groups. You can edit a user account at any time to add and remove the user from groups.

1. Click **Admin Tools**, and then click **Users**.

You see the **User Search** page.

2. Search for a user, and then select the user.

3. On the **User Profile** page, click **Edit User**.

You see the **Edit User** page.

4. Edit the groups to which this user belongs:

a. To add a user to a group, use the search field provided to locate the group. Click **Add** to the right of each group you want the user to be a part of. The groups the user belongs to show beneath the **Groups** list.

b. To remove a user from a group, simply click the group you want to remove beneath the **Groups** list.

5. Click **Save Changes**.

Managing groups

The Groups tool lets you create and manage user groups.

Browsing the user groups

The Groups page contains a multi-paned panel that lets you navigate the hierarchy of user groups.

1. Click **Admin Tools**, and then click **Groups**.

2. On the **Groups** page, click **Browse**.

The leftmost pane displays all top-level user groups.

3. To view all groups, including the system groups, select the **Show System Groups** check box, and then click **Browse**.

System groups are created in the background, for example, when you create a site. You can show these groups so that you can edit the **Display Name**, add users, or delete the group.

4. Click a group to display its contents in the panel directly to the right.
The content can be subgroups and/or individual users. Text at the bottom of this panel indicates the number of groups and users that belong to the selected group.
5. As you browse the group structure, a navigation path is displayed at the top of the panel indicating your selections stemming from the initial pane. Click any link in this path to step back to that selection.
6. To browse a different group, click the first link in the navigation path to return to the top-level groups, then select a new group to browse.

Searching for a group

The Search feature enables you to locate any user group, regardless of where it exists in the group hierarchy. Once located, you can edit or delete the group.

1. Click **Admin Tools**, and then click **Groups**.
2. In the search box, type the full or partial identifier, not display name.
The search is not case sensitive.
3. Click **Search**.
In the results table, click the column headings to sort the results as desired.

Creating a new group

Use the Groups tool to create both top level user groups and subgroups within existing groups.

1. Click **Admin Tools**, and then click **Groups**.
2. On the **Groups** page, click **Browse**.
The leftmost pane displays all top-level user groups.
3. Navigate to the user group where you want to create the new group.
 - To create a top-level group, click the **New Group** icon at the top of the initial pane.
 - To create a subgroup, browse the group structure to locate the desired parent group. Select this group and then click the **New Subgroup** icon at the top of the pane immediately to the right.

The **New Group** page appears. Fields marked with an asterisk (*) are required.

4. Complete the required fields.

Field	What is it?
Identifier	This is a name that the system uses to identify the group. Once you have created the group, you cannot change this identifier.
Display Name	This is the group name that shows in Alfresco where you manage groups and also is the name shown to members of this group.

5. Click **Create Group**.
If you intend to immediately create another group at the same level, click **Create and Create Another**. This creates the group specified and clears the fields without returning you to the **Groups** page.

Editing an existing group

Edit a user group to change the group's display name. Once created, you cannot edit the group's identifier.

1. Click **Admin Tools**, and then click **Groups**.
2. On the **Groups** page, click **Browse**.
The leftmost pane shows all the top-level user groups.
3. Navigate the group structure or use the search feature to locate the user group you want to edit.
The search is not case sensitive.
4. Position the cursor over the desired group to display its available actions, and then click the **Edit Group** icon.
5. Edit the group's **Display Name**.
6. Click **Save Changes**.

Deleting an existing group

Delete a user group to remove it from the system.

1. Click **Admin Tools**, and then click **Groups**.
2. On the **Groups** page, click **Browse**.
The leftmost pane shows all the top-level user groups.
3. Navigate the group structure or use the search feature to locate the user group you want to delete.
You must enter a minimum of one (1) character. The search is not case sensitive.
4. Position the cursor over the desired group to display its available actions.
5. Click the **Delete Group** icon.
A message prompts you to confirm the deletion.
6. Click **Delete**.

Managing group membership

To populate a user group, you can add both individual users and existing user groups.

1. Click **Admin Tools**, and then click **Groups**.
2. On the **Groups** page, click **Browse**.
The leftmost pane shows all the top-level user groups.
3. Navigate the group structure to locate the user group you want to work with. Click a user group to select it.
4. Using the icons in the pane directly to the right of where you selected the group, perform the desired action:
 - a. To add a user, click the **Add User** icon. Using the search feature provided, locate the user you want to add to the selected group. Click **Add** to the right of the user.
 - b. To add a group, click the **Add Group** icon. Using the search feature provided, locate the group you want to add to the selected group. Click **Add** to the right of the user.

The individual user or group is added as a child to the group selected in the panel.

Configuring Alfresco

Configuration overview

Alfresco is preconfigured with a set of system configuration parameters. Many of the system configuration parameters are completely exposed as properties, which you can configure for your specific environment requirements.

The recommended way to configure Alfresco using the following methods:

- Alfresco Admin Console
- Editing the global properties
- Using a JMX client, such as JConsole

Alfresco Admin Console

The Alfresco Admin Console is a tool that gives a quick and easy way to manage Alfresco configuration. For more information, see [Using the Admin Console](#).

Global properties file

The global properties file (`alfresco-global.properties`) is used by Alfresco to detect the extended properties. For example, when you install Alfresco, many of the installation settings are saved in the global properties file. You can continue to use the global properties to do all your property extensions; however, whenever you make a change, you must restart the Alfresco server.

JMX client

The JMX client allows you to edit the settings while the system is running. The settings you change are automatically persisted in the database and synchronized across a cluster. When you start up Alfresco, the system initially uses the `alfresco-global.properties` file to set the properties within the JMX client, but then any changes you make in the JMX client persist in the database but are not reflected back into the `alfresco-global.properties` file.

There are two types of property that you can edit:

Type 1: Properties specified directly in XML files

For example:

```
<bean id="wcm_deployment_receiver"
class="org.alfresco.repo.management.subsystems.ChildApplicationContextFactory"
  <parent="abstractPropertyBackedBean">
    <property name="autoStart">
      <value>true</value>
    </property>
  </bean>
```

The value for the property `autoStart` is set to `true` directly in the `wcm-bootstrap-context.xml` file.

Type 2: Properties set by variables in XML files

For example:

```
<bean id="userInstallationURI" class="org.alfresco.util.OpenOfficeURI">
  <constructor-arg>
    <value>${ooo.user}</value>
  </constructor-arg>
</bean>
```

The value for the property `constructor-arg` is replaced with a variable `${ooo.user}`.

When Alfresco starts up, type 1 properties are read from the XML file; type 2 properties get their values read from all the various property files. Then, the database is checked to see if there are any property values set there, and if any property has been changed, this value is used instead.

Some of the type 2 properties can be viewed and changed by the JMX console, some cannot. For example, `ooo.exe` can be viewed and changed using the JMX client; `index.recovery.mode` cannot be viewed or changed using the JMX client.

In a new Alfresco installation, none of these properties are stored in the database. If you set a property using the JMX interface, Alfresco stores the value of the property in the database. If you never use JMX to set the value of a property, you can continue using the `alfresco-global.properties` file to set the value of the property. Once you change the property setting using JMX, and it is therefore stored in the database, you cannot use the properties files to change the value of that property.

 For advanced configuration, you can also extend or override the Spring bean definitions that control Alfresco's Java classes. To do so, add or copy a Spring bean file named `*-context.xml` to the `<extension>` directory, or `<web-extension>` directory to extend Share. For examples of the Spring bean extensions, download the sample extension files.

Modify Alfresco applications

This topic describes the basic configuration updates you might want to do to customize Alfresco. It also lists the different properties files you can modify to apply configuration changes to Alfresco.

- [Updating system configuration parameters](#): You can configure Alfresco for your specific environment requirements either by using the Alfresco Admin Console, or by editing the `alfresco-global.properties` file, or by using a JMX client.
- [Alfresco Share](#): A number of options are available to customize Alfresco Share. To configure Share, use the configuration file named `share-config-custom.xml`.
- [Solr configuration](#): When you install Alfresco, several Solr-related configuration files are made available to you. To configure Solr, use the configuration file named `solrcore.properties`.

 Remember not to use the default user names, URLs, or passwords with different environment.

 You can customize or scale up Alfresco to meet your login and security requirements. See [Setting up Alfresco authentication and security](#) for more information.

Runtime administration with a JMX client

By default, you can reconfigure Alfresco by shutting down the server, editing the relevant property in the configuration files, and then restarting the server. There are some support operations that can be performed on-demand at runtime without needing to restart the server.

The Java Management Extension (JMX) interface allows you to access Alfresco through a standard JMX console that supports JMX Remoting (JSR-160). This lets you:

- Manage Alfresco subsystems
- Change log levels
- Enable or disable file servers (FTP/CIFS/NFS)
- Set server read-only mode
- Set server single-user mode
- Set server maximum user limit, including ability to prevent further logins
- Count user sessions/tickets
- User session/ticket invalidation

Example consoles include:

- JConsole (supplied with Java SE 5.0 and higher)

- MC4J
- JManage

Some of these consoles also provide basic graphs and/or alerts for monitoring JMX-managed attributes.

Connecting to Alfresco through JMX client

You can connect to the Alfresco MBean server through a JMX client that supports JSR-160.

1. Ensure that sure you have this somewhere in your `java_opts`:

```
-Dcom.sun.management.jmxremote
```

This tells the running JVM to start the JMX service.

2. Ensure that you have the following properties set in the `alfresco-global.properties` file:

```
alfresco.rmi.services.port=50500
alfresco.rmi.services.host=<hostname>
```

Check that the `<hostname>` can be resolved from where you are running the JMX client.

3. Open a JMX client that supports JMX Remoting (JSR-160).
4. Connect to the JMX URL:

```
service:jmx:rmi:///jndi/rmi://<hostname>:50500/alfresco/jmxrmi
```

Where `<hostname>` is the name of a reachable domain name or an IP address. If you running this on the local server, you can use `localhost`.

5. Enter the default JMX user name: `controlRole`
6. Enter the default JMX password: `change_asap`

 You must change the default JMX password as soon as possible.

The user `controlRole` is the default user name used to access and configure Alfresco with a JMX client.

The user `monitorRole` is the default user name used within monitoring tools, for example, Nagios or Hyperic.

7. Change the default JMX password as soon as possible. You can set a new password in override configuration files.

Create two new files called:

```
alfresco-jmxrmi.password
alfresco-jmxrmi.access
```

Copy the files to a location of your choice and then add the `alfresco.jmx.dir=` property to the `alfresco-global.properties` file to specify the directory path of the configuration files. For example:

```
alfresco.jmx.dir=/etc/alfresco/config
```

You can also set this on the Alfresco command line:

```
-Dalfresco.jmx.dir=/etc/alfresco/config
```

8. Open the `alfresco-jmxrmi.password` file and add the following properties for the `monitorRole` and `controlRole` users, where `new_pw` is your preferred password.

```
monitorRole new_pw
controlRole new_pw
```

9. Save the file.

10. Open the `alfresco-jmxrmi.access` file and add the following properties for the read-only or read/write access levels of each user.

```
monitorRole    readonly
controlRole    readwrite
```

11. Save the file.



It is possible to set the JVM (Oracle/Sun JVM-specific) arguments directly:

```
-Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.access.file=/etc/alfresco/config/
jmxremote.access
-Dcom.sun.management.jmxremote.password.file=/etc/alfresco/config/
jmxremote.password
-Dcom.sun.management.jmxremote.authenticate=true
```

Disabling JMX

JMX functionality is set using the `alfresco.jmx.connector.enabled` property in the `<classpathRoot>/alfresco-global.properties` file.

To disable the JMX server:

1. Open the `<classpathRoot>/alfresco-global.properties` file.
2. Set the following property: `alfresco.jmx.connector.enabled=false`
3. Save the file.

Configuring Alfresco with JConsole

This section describes how to use the JMX client, JConsole for Alfresco runtime administration. JConsole is a JMX client available in the Oracle Java SE Development Kit (JDK).

The initial configuration that displays in JConsole is set from the `alfresco-global.properties` file.

1. Open a command console.
2. Locate your JDK installation directory.
For example, the JDK directory is often `java/bin`.

3. Enter the following command:

```
jconsole
```

The **JConsole New Connection** window displays.

4. Double-click on the Alfresco Java process.

For Tomcat, this the Java process is usually labelled as **org.apache.catalina.startup.Bootstrap start**.

JConsole connects to the managed bean (or MBean) server hosting the Alfresco subsystems.

5. Select the **MBeans** tab.

The available managed beans display in JConsole.

6. Navigate to **Alfresco > Configuration**.

The available Alfresco subsystems display in an expandable tree structure. When you select a subsystem, the **Attributes** and **Operations** display below it in the tree.

7. Select **Attributes** and set the required Alfresco subsystem attribute values.

Values that can be edited are shown with blue text.

When you change a configuration setting, the subsystem automatically stops.

8. Restart the Alfresco subsystem:
 - a. Navigate to the subsystem.
 - b. Select **Operations**.
 - c. Click **Start**.
9. To stop the Alfresco subsystem without editing any properties:
 - a. Navigate to the subsystem.
 - b. Select **Operations**.
 - c. Click **Stop**.
10. To revert back to all the previous edits of the Alfresco subsystem and restores the default settings:
 - a. Navigate to the subsystem.
 - b. Select **Operations**.
 - c. Click **Revert**.
11. Click **Connection > Close**.

The settings that you change in a JMX client, like JConsole, are persisted in the Alfresco database. When you make a dynamic edit to a subsystem:

1. When a subsystem, that is currently running, is stopped, its resources are released and it stops actively listening for events. This action is like a sub-part of the server being brought down. This 'stop' event is broadcast across the cluster so that the subsystem is brought down simultaneously in all nodes.
2. The new value for the property is persisted to the Alfresco database.

There are two ways to trigger a subsystem to start:

- The start operation
- An event that requires the subsystem

About the `alfresco-global.properties` file

The global properties `alfresco-global.properties` file contains the customizations for extending Alfresco.

If you install Alfresco using one of the installation wizards, the `alfresco-global.properties` file is modified with the settings that you chose during installation. If you install Alfresco manually using the WAR file, you can modify properties in the `alfresco-global.properties` file.

A sample global properties file is supplied with the Alfresco installation. By default, the file contains sample settings for running Alfresco, for example, the location of the content and index data, the database connection properties, the location of third-party software, and database driver properties.

Modifying the global properties file

-  For edits to the `alfresco-global.properties` file, when specifying paths for Windows systems, you must replace the Windows path separator characters with either the `\\` separator or the forward slash `/` Unix path separator.

1. Locate and open the `alfresco-global.properties.sample` file.

For example, for Tomcat, browse to the `$TOMCAT_HOME/shared/classes/` directory.

This file contains sample configuration settings for Alfresco. To enable or modify a setting, remove the comment (`#`) character. Comment out all the properties you do not want to modify by adding the `"#"` character.

2. Ensure that the `dir.root=` property points to a root location for the storage of content binaries and index files.

For example, `dir.root=C:/Alfresco/alf_data`.

3. Set the database connection properties.

Property	Description
<code>db.username=alfresco</code>	Specifies the name of the main Alfresco database user. This name is used to authenticate with the database.
<code>db.password=alfresco</code>	Specifies the password for the Alfresco database user. This password is used to authenticate with the database.

Additional database properties can be set for further configuration. Refer to the [Configuring databases](#) for more information.

4. Specify the locations of the following external software:

Property	Description
<code>ooo.exe=</code>	Specifies the location of the OpenOffice installation.
<code>ooo.enabled=</code>	Specifies whether to use the Direct OpenOffice subsystem.
<code>jodconverter.officeHome=</code>	Specifies the location of the OpenOffice installation for JODConverter transformations. To use the JODConverter, uncomment the <code>ooo.enabled=false</code> and <code>jodconverter.enabled=true</code> properties.
<code>jodconverter.portNumbers=</code>	Specifies the port numbers used by each JODConverter processing thread. The number of process will match the number of ports.
<code>jodconverter.enabled=</code>	Specifies whether to use the JODConverter. Set the property to <code>jodconverter.enabled=true</code> .
<code>img.root=</code>	Specifies the location of the ImageMagick installation.
<code>swf.exe=</code>	Specifies the location of the SWF tools installation.

5. Configure your supported database for use with Alfresco. See [Configuring databases](#).
6. Select a JDBC driver used with each connection type.
7. Add your global custom configurations.
8. Save your file without the `.sample` extension.

You need to restart the Alfresco server for the configuration changes to take effect.

Setting composite properties in the global properties file

This section uses the `imap.server.mountPoints` property as an example.

The `ImapConfigMountPointsBean` class that holds the component beans has four properties of its own:

- `beanName`

- store
- rootPath
- mode

1. Open the `<classpathRoot>/alfresco-global.properties` file.
2. To set some overall defaults for all component instances, use the format:

```
<property>.default.<component property>
```

These values would show up, for example, when you added a new component instance but did not specify its properties.

For example:

```
imap.server.mountPoints.default.store=${spaces.store}
imap.server.mountPoints.default.rootPath=/
${spaces.company_home.childname}
imap.server.mountPoints.default.mode=virtual
```

This example does not define a default for `beanName` because there is a way of populating it for each instance.

3. To set up the `imap.server.mountPoints` with a composite value, set the master composite property using a comma-separated list.

For example:

```
imap.server.mountPoints=Repository_virtual,Repository_archive
```

This defines that the property contains two `ImapConfigMountPointsBean` instances, named `Repository_virtual` and `Repository_archive`. Because `ImapConfigMountPointsBean` implements the `BeanNameAware` Spring interface and has a `beanName` property, these instance names are automatically set as the bean names.

4. To define component properties specific to each component instance, use the format:

```
<property>.value.<component instance name>.<component property>
```

For example:

```
imap.server.mountPoints.value.Repository_virtual.mode=virtual
imap.server.mountPoints.value.Repository_archive.mode=archive
```

Java command line

The most common use of the Java command line is in a multiple-machine environment where the basic, common customizations are set using standard properties and the machine-specific values are set using command line options. For example, an administrator is likely to configure all Alfresco installs to behave similarly by setting properties in the configuration files, but will use the Java command line to vary settings like the database connection, Content Store locations, and CIFS domain name.

Setting properties on the Java command line

- Add a `-Dprop=value` to `JAVA_OPTS` or for anything that is sent to the Java command line. For example, `-Ddir.root=/alfresco/data -Ddb.url=xxxxx`.

Customizing individual configuration items

The Alfresco configuration is implemented using three types of files:

- Extension files
- Bean files
- Spring bean definitions

Customizing extension files

A configuration file contains `<alfresco-config>` tags outside the `<config>` tags. You must preserve these tags in your customized file.

1. Open the configuration file that you want to customize.
2. Edit each pair of `<config>` `</config>` tags that you want to modify.

Replacing a configuration

To replace the configuration, add a `replace="true"` attribute to the configuration element. For example: `<config evaluator="xx" condition="yy" replace="true">`

 Any configuration within a section marked this way completely replaces any configuration found in the Alfresco-maintained files.

Modifying one property

The attribute `replace` completely replaces the configuration. To modify one property, add the changed piece.

3. Save your customized file.

Modifying Spring bean definition files

The Spring bean definitions are within configuration files in the following directories:

- The `<extension>` directory contains the configuration files for extending Alfresco.
 - The `<web-extension>` directory contains the configuration files for extending Alfresco Share.
1. Browse to the `<extension>` directory. For example, for Tomcat:
 - (Windows) `C:\Alfresco\tomcat\shared\classes\alfresco\extension`
 - (Linux) `tomcat/shared/classes/alfresco/extension`

Each file has a copy with a `.sample` extension.

2. Open the configuration file with the `.sample` extension.
3. Add your configurations to the file.
4. Save the file without the `.sample` extension.

Customizing bean files

There are two common uses of beans:

- To define properties
- To point to one or more of your customized files

A typical bean file is `<extension>/custom-repository-context.xml`. A bean file contains `<?xml>` and `<!DOCTYPE>` headers, and `<beans>` tags outside the `<bean>` tags. You must preserve these items in your customized file.

 When you override a `<bean>`, the entire effects of the original bean are lost. The effect is the same as if you had overridden a `<config>` by using `replace="true"`. Therefore, the overriding `<bean>` must contain any information from the default bean that you want to keep, as well as any additional information.

For example, if a core bean has four values, and you want to modify a single value, the resultant bean must still have four values. However, if you want to add a value, then the resultant bean must have five values - the original four values plus the added value.

1. Open the bean file that you want to customize.

For example, the following `<bean>` is from the `<configRoot>/classes/alfresco/action-services-context.xml` file:

```
<bean id="mail"
  class="org.alfresco.repo.action.executer.MailActionExecuter"
  parent="action-executer">
  <property name="publicAction">
    <value>true</value> <!-- setting to true -->
  </property>
  <property name="mailService">
    <ref bean="mailService"></ref>
  </property>
</bean>
```

2. Delete each pair of `<bean>` `</bean>` tags that you do not want to modify.
3. Modify the contents of the remaining `<bean>` tags.

For example, the following overrides the `publicAction` property from the previous example:

```
<bean id="mail"
  class="org.alfresco.repo.action.executer.MailActionExecuter"
  parent="action-executer">
  <property name="publicAction">
    <value>false</value> <!-- setting to false -->
  </property>
  <property name="mailService">
    <ref bean="mailService"></ref>
  </property>
</bean>
```

4. Save the file.

Configuring databases

Configuring a PostgreSQL database

This section describes how to configure a PostgreSQL database for use with Alfresco.

1. Install the PostgreSQL database connector. The database connector allows PostgreSQL database to talk to the Alfresco server.
 - a. Download `postgresql-9.3-1102.jdbc4.jar` from the PostgreSQL download site: <http://www.postgresql.org/download/>.
 - b. Copy the JAR file into the `/lib` directory.
For example, for Tomcat, copy the JAR file into the `<TOMCAT_HOME>/lib` directory.
2. Create a database named `alfresco`.
3. Create a user named `alfresco`.
This user must have write permissions on all tables and sequences.
4. Set the new user's password to `alfresco`.
5. Ensure the `alfresco` user has the required privileges to create and modify tables.
6. Open the `<classpathRoot>/alfresco-global.properties.sample` file.
7. Locate the following line:

```
dir.root=./alf_data
```

8. Edit the line with an absolute path to point to the directory in which you want to store Alfresco data. For example: `dir.root=C:/Alfresco/alf_data`

- Uncomment the following properties:

```
# PostgreSQL connection (requires postgresql-8.2-504.jdbc3.jar or
equivalent)
#
db.driver=org.postgresql.Driver
db.url=jdbc:postgresql://${db.host}:${db.port}/${db.name}
```

- Set the other database connection properties.

```
db.name=alfresco
db.username=alfresco
db.password=alfresco
db.host=localhost
db.port=5432
db.pool.max=275
```

 Ensure that these database connection properties are not commented out.

- Save the file without the `.sample` extension.
- To allow password-authenticated connections through TCP/IP, ensure that the PostgreSQL configuration file, `pg_hba.conf`, contains the following line:

```
host all all 127.0.0.1/32 password
```

- Restart the Alfresco server.

If you receive JDBC errors, ensure the location of the PostgreSQL JDBC drivers are on the system path, or add them to the relevant `lib` directory of the application server.

Configuring an Oracle database

This section describes how to configure an Oracle RDBMS database for use with Alfresco.

The Oracle database is case sensitive, so any configuration setting that you add into the `alfresco-global.properties` file must match the case used in Oracle.

 The Oracle database must be created with the AL32UTF8 character set.

- Install the Oracle database connector. The database connector allows Oracle database to talk to the Alfresco server.
 - Download `ojdbc7.jar` from the Oracle download site.
Use the `ojdbc7.jar` from within the Oracle Database 12c Release 1 (12.1.0.1) drivers.
 - Copy the JAR file into the `/lib` directory.
For example, for Tomcat, copy the JAR file into the `<TOMCAT_HOME>/lib` directory.
- Create a database named `alfresco`.
- Create a user named `alfresco`.
The `alfresco` user must have Connect and Resource privileges in Oracle.
This user must have write permissions on all tables and sequences.
- Set the new user's password to `alfresco`.
- Ensure the `alfresco` user has the required privileges to create and modify tables.
You can remove these privileges once the server has started, but they might also be required for upgrades.

 When connecting to Oracle Database 12c, you must configure privileges on tablespace "USERS" to avoid the following error:

```
ORA-01950: no privileges on tablespace 'USERS'
```

You can do this by using one of the following commands:

```
ALTER USER <username> QUOTA <QUOTE_M> ON <tablespace name>
```

or

```
GRANT UNLIMITED TABLESPACE TO <username>
```

- Open the `<classpathRoot>/alfresco-global.properties.sample` file.
- Locate the following line:

```
dir.root=./alf_data
```

- Edit the line with an absolute path to point to the directory in which you want to store Alfresco data. For example: `dir.root=C:/Alfresco/alf_data`
- Uncomment the following properties:

```
# Oracle connection
#
db.driver=oracle.jdbc.OracleDriver
db.url=jdbc:oracle:thin:@${db.host}:${db.port}:${db.name}
```

- Set the other database connection properties.

```
db.name=alfresco
db.username=alfresco
db.password=alfresco
db.host=localhost
db.port=1521
db.pool.max=275
```

 Ensure that these database connection properties are not commented out.

- Save the file without the `.sample` extension.
- Copy the Oracle JDBC driver JAR into `/lib`.
- Restart the Alfresco server.

If you receive JDBC errors, ensure the location of the Oracle JDBC drivers are on the system path, or add them to the relevant `lib` directory of the application server. The Oracle JDBC drivers are located in the `<oraInst>/ora<ver>/jdbc/lib` directory (for example, `c:\oracle\ora92\jdbc\lib`).

The JDBC driver for Oracle is in the JAR file: `ojdbc7.jar`. However, if you see the following error, then add the `Doracle.jdbc.thinLogonCapability=o3` parameter to `JAVA_OPTS`:

```
java.sql.SQLException: OAUTH marshaling failure
```

Configuring a SQL Server database

This section describes how to configure a Microsoft SQL Server database for use with Alfresco. To modify the default database configuration, you must edit values in the `<classpathRoot>/alfresco-global.properties` file.

- Install the Microsoft SQL Server database connector. The database connector allows SQL Server database to talk to the Alfresco server.

This release requires the Microsoft SQL Server JDBC Driver 4.0 for compatibility with the SQL Server database.

- Download `sqljdbc4.jar` from the Microsoft SQL Server download site.
 - Copy the JDBC driver into the `<TOMCAT_HOME>/lib` directory.
- Create a database named `alfresco`.
Create the database using default collation settings.

3. Create a user named `alfresco`.

This user must have write permissions on all tables and sequences.

4. Set the new user's password to `alfresco`.
5. Ensure the `alfresco` user has the required privileges to create and modify tables.
This can be removed once the server has started, but may be required during upgrades.
6. Enable snapshot isolation mode with the following command:

```
ALTER DATABASE alfresco SET ALLOW_SNAPSHOT_ISOLATION ON;
```

7. Ensure that the TCP connectivity is enabled on the fixed port number 1433.
8. Open the `<classpathRoot>/alfresco-global.properties` file.
9. Locate the following property:
`dir.root=`
10. Edit the line with an absolute path to point to the directory in which you want to store Alfresco data. For example: `dir.root=C:/Alfresco/alf_data`
11. Set the database connection properties.

```
db.name=alfresco
db.username=alfresco
db.password=alfresco
db.host=localhost
db.port=1433
db.pool.max=275
```

12. Add the following properties to register the driver and set up the connection:

```
db.driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
db.url=jdbc:sqlserver://${db.host}:${db.port};databaseName=${db.name}
db.txn.isolation=4096
```

13. Save the file.
14. Restart the Alfresco server.

If you receive JDBC errors, ensure the location of the SQL Server JDBC drivers are on the system path, or add them to the relevant `lib` directory of the application server.

Configuring the MySQL database

1. Install the MySQL database connector.

The MySQL database connector is required when installing Alfresco with MySQL. The database connector allows MySQL database to talk to the Alfresco server.

- a. Download `mysql-connector-java-5.1.32` from the MySQL download site: <http://dev.mysql.com/>.
- b. Copy the JAR file into the `/lib` directory.

For example, for Tomcat, copy the JAR file into the `<TOMCAT_HOME>/lib` directory.

2. Create a database named `alfresco`.

If you are using MySQL and require the use of non-US-ASCII characters, you need to set the encoding for internationalization. This allows you to store content with accents in the repository. The database must be created with the UTF-8 character set and the `utf8_bin` collation. Although MySQL is a unicode database, and Unicode strings in Java, the JDBC driver might corrupt your non-English data. Ensure that you keep the `?useUnicode=yes&characterEncoding=UTF-8` parameters at the end of the JDBC URL.



You also must ensure that the MySQL database is set to use UTF-8 and InnoDB. Refer to [Configuration settings for using MySQL with Alfresco](#).

3. Create a user named `alfresco`.

4. Set the new user's password to `alfresco`.
5. Navigate to the `<ALFRESCO_HOME>/alf_data/` directory and empty the `<contentstore>` directory.

This is because the `contentstore` must be consistent with the database. Step 2 created an empty database, and so the `contentstore` must also be empty.

6. Open the `<classpathRoot>/alfresco-global.properties.sample` file.
7. Edit the following line with an absolute path to point to the directory in which you want to store Alfresco data.

For example: `dir.root=C:/Alfresco/alf_data`

8. Uncomment the following properties:

```
db.driver=org.gjt.mm.mysql.Driver
db.url=jdbc:mysql://${db.host}:${db.port}/${db.name}?
useUnicode=yes&characterEncoding=UTF-8
```

9. Set the other database connection properties.

```
db.name=alfresco
db.username=alfresco
db.password=alfresco
db.host=localhost
db.port=3306
db.pool.max=275
```

 Ensure that these database connection properties are not commented out.

10. Copy the `keystore` directory from the `alf_data` directory at the old location to the `alf_data` directory at the new location, which is specified in Step 7.
11. (Optional) Enable case sensitivity.

The default, and ideal, database setting for Alfresco is to be case-insensitive. For example, the user name properties in the `<configRoot>\classes\alfresco\repository.properties` file are:

```
# Are user names case sensitive?
user.name.caseSensitive=false
domain.name.caseSensitive=false
domain.separator=
```

If your preference is to set the database to be case-sensitive, add the following line to the `alfresco-global.properties` file:

```
user.name.caseSensitive=true
```

12. Save the file without the `.sample` extension.
13. Restart the Alfresco server.

If you receive JDBC errors, ensure the location of the MySQL JDBC drivers are on the system path, or add them to the relevant `lib` directory of the application server.

Optimizing MySQL to work with Alfresco

When installing MySQL, there are some settings that are required for it to work with Alfresco. This section describes the configuration settings that you should use in your MySQL instance.

The following table represents the specific settings in the MySQL configuration wizard that enable MySQL to work effectively with Alfresco.

Configuration wizard dialog	Setting for Alfresco
Server Type	Choose Dedicated MySQL Server Machine . The option selected determines the memory allocation.

Configuration wizard dialog	Setting for Alfresco
Database usage	Choose Transactional Database Only . This creates a database that uses InnoDB as its storage engine.
InnoDB Tablespace	Accept the default drive and path.
Concurrent Connections	Select Decision Support (DSS) OLAP . This sets the approximate number of concurrent connections to the server.
Networking and Strict Mode Options	Accept the default networking options (Enable TCP/IP Networking, Port Number 3306), and the default server SQL mode (Enable Strict Mode).
Character Set	Select Best Support for Multilingualism . This sets the default character set to be UTF-8 (set in <code>character-set-server</code>).
Security Options	Select Modify Security Settings . Type the root password <code>admin</code> , then retype the password.

Use the following variable setting to enable MySQL to prevent some update operations from locking database access. Add this setting to your MySQL configuration file (`/etc/my.cnf`) in the `[mysqld]` section:

```
innodb_locks_unsafe_for_binlog = 1
```

Ensure that you restart the MySQL server after adding this setting.

The effect of enabling this variable is similar to setting the transaction isolation level to `READ_COMMITTED`.

By default, table aliases are case sensitive on Unix but not on Windows or Mac OS X. Use the following variable setting to enable MySQL server to handle case sensitivity of database and table names:

```
lower_case_table_names=1
```

Using this variable setting allows MySQL to convert all table names to lowercase on storage and lookup. This behavior also applies to database names and table aliases. This setting also prevents data transfer problems between platforms and between file systems with varying case sensitivity.

Refer to the <http://dev.mysql.com/> website for more information on this variable.

Configuring the MariaDB database

1. Download `mariadb-java-client-1.1.7.jar` from the [MariaDB download site](#).

2. Copy the JAR file into the `/lib` directory.

For example, for Tomcat, copy the JAR file into the `<TOMCAT_HOME>/lib` directory.

3. Create a database named `alfresco`.

If you are using MariaDB and require the use of non-US-ASCII characters, you need to set the encoding for internationalization. This allows you to store content with accents in the repository. The database must be created with the UTF-8 character set and the `utf8_bin` collation. Although MariaDB is a unicode database, and Unicode strings in Java, the JDBC driver might corrupt your non-English data. Ensure that you keep the `?useUnicode=yes&characterEncoding=UTF-8` parameters at the end of the JDBC URL.

4. Create a user named `alfresco`.
5. Set the new user's password to `alfresco`.

- Navigate to the `<ALFRESCO_HOME>/alf_data/` directory and empty the `<contentstore>` directory.

This is because the `contentstore` must be consistent with the database. Step 3 created an empty database, and so the `contentstore` must also be empty.

- Open the `<classpathRoot>/alfresco-global.properties.sample` file.
- Edit the following line with an absolute path to point to the directory in which you want to store Alfresco data.

For example: `dir.root=C:/Alfresco/alf_data`

- Uncomment the following properties:

```
db.driver=org.mariadb.jdbc.Driver
db.url=jdbc:mariadb://${db.host}:${db.port}/${db.name}?
useUnicode=yes&characterEncoding=UTF-8
```

- Set the other database connection properties.

```
db.name=alfresco
db.username=alfresco
db.password=alfresco
db.host=localhost
db.port=3306
db.pool.max=275
```



Ensure that these database connection properties are not commented out.

- Copy the `keystore` directory from the `alf_data` directory at the old location to the `alf_data` directory at the new location, which is specified in Step 8.
- (Optional) Enable case sensitivity.

The default, and ideal, database setting for Alfresco is to be case-insensitive. For example, the user name properties in the `<configRoot>\classes\alfresco\repository.properties` file are:

```
# Are user names case sensitive?
user.name.caseSensitive=false
domain.name.caseSensitive=false
domain.separator=
```

If your preference is to set the database to be case-sensitive, add the following line to the `alfresco-global.properties` file:

```
user.name.caseSensitive=true
```

- Save the file without the `.sample` extension.
- Restart the Alfresco server.

If you receive JDBC errors, ensure the location of the MariaDB JDBC drivers are on the system path, or add them to the relevant `lib` directory of the application server.

Configuring a DB2 database

This section describes how to configure a DB2 database for use with Alfresco.

- Install the DB2 database connector. The database connector allows DB2 database to talk to the Alfresco server.
 - Obtain a copy of `db2jcc4.jar`. This should be available in the `/java` or `/jdbc` directory of your DB2 installation.
 - Copy the JAR file into the `<TOMCAT_HOME>/lib` directory for Tomcat.
- Create a database named `alfresco`.

Create the database with a larger page size of 32 KB. Ensure that the database is created with the UTF-8 character set.

If you do not create the database with these settings, you will see error SQL0286N (sqlCode -286, sqlstate 42727) because the schema is created for tables that do not fit the page size.

3. Ensure that the `cur_commit` database configuration parameter is set to `ON`.

For new databases, this parameter is set to `ON`, by default. If you have upgraded from a previous DB2 release, you must set this parameter manually.

4. Create a user named `alfresco` and set the associated schema.

This user must have write permissions on all tables and sequences.

DB2 only integrates with the operating system security. You can not add a database user with a password in the DB2 database as you can with some other databases, for example the Oracle database.

5. Open the `<classpathRoot>/alfresco-global.properties.sample` file.
6. Locate the following line:

```
dir.root=./alf_data
```

7. Edit the line with an absolute path to point to the directory in which you want to store Alfresco data. For example: `dir.root=C:/Alfresco/alf_data`
8. Uncomment the following properties:

```
# DB2 connection
#
db.driver=com.ibm.db2.jcc.DB2Driver
db.url=jdbc:db2://${db.host}:${db.port}/
${db.name}:retrieveMessagesFromServerOnGetMessage=true;
```

9. Set the other database connection properties.

```
db.name=alfresco
db.host=localhost
db.port=50000
db.pool.max=275
```



Ensure that these database connection properties are not commented out.

10. Save the file without the `.sample` extension.
11. Restart the Alfresco server.

If you receive JDBC errors, ensure the location of the DB2 JDBC drivers are on the system path, or add them to the relevant `lib` directory of the application server.

Advanced database configuration properties

As an administrator, you need to edit some advanced properties to customize your database configuration. Many properties, however, do not need to be edited.

Alfresco One 5.0 supports Oracle, Microsoft SQL Server, DB2, as well as MySQL and PostgreSQL.

The advanced database configuration properties are categorized into two groups based on their relevance:

- properties that you **SHOULD** edit
- properties that you **COULD** edit

The following table describes the properties that you **SHOULD** edit:

Property name	Description	Default value
<code>db.txn.isolation</code>	The JDBC code number for the transaction isolation level, corresponding to those in the <code>java.sql.Connection</code> class. The value of -1 indicates that the database's default transaction isolation level should be used. For the Microsoft SQL Server JDBC driver, the special value of 4096 should be used to enable snapshot isolation.	-1
<code>db.pool.initial</code>	The number of connections opened when the pool is initialized.	10
<code>db.pool.validate.query</code>	The SQL query that will be used to ensure that your connections are still alive. This is useful if your database closes long-running connections after periods of inactivity.	For Oracle database, use <code>SELECT 1 from dual</code> For MySQL database, use <code>SELECT 1</code> For SQL Server database, use <code>SELECT 1</code>

The following table describes the properties that you **COULD** edit:

Property name	Description	Default value
<code>db.pool.statements.enable</code>	A Boolean property. When set to <code>true</code> it indicates that all pre-compiled statements used on a connection will be kept open and cached for reuse.	<code>true</code>
<code>db.pool.statements.max</code>	The maximum number of pre-compiled statements to cache for each connection. The Alfresco default is 40. Note that Oracle does not allow more than 50 by default.	40
<code>db.pool.idle</code>	The maximum number of connections that are not in use kept open.	10
<code>db.pool.max</code>	The maximum number of connections in the pool. See the Note below for more information on this property.	275
<code>db.pool.min</code>	The minimum number of connections in the pool.	10
<code>db.pool.wait.max</code>	Time (in milliseconds) to wait for a connection to be returned before generating an exception when connections are unavailable. A value of 0 or -1 indicates that the exception should not be generated.	-1
<code>db.pool.validate.borrow</code>	A Boolean property. When set to <code>true</code> it indicates that connections will be validated before being borrowed from the pool.	<code>true</code>
<code>db.pool.validate.return</code>	A Boolean property. When set to <code>true</code> it indicates that connections will be validated before being returned to the pool.	<code>false</code>
<code>db.pool.evict.interval</code>	Indicates the interval (in milliseconds) between eviction runs. If the value of this property is zero or less, idle objects will not be evicted in the background.	600000

Property name	Description	Default value
<code>db.pool.evict.idle.min</code>	The minimum number of milliseconds that a connection may remain idle before it is eligible for eviction.	1800000
<code>db.pool.evict.validate</code>	A Boolean property. When set to <code>true</code> it indicates that the idle connections will be validated during eviction runs.	<code>false</code>
<code>db.pool.abandoned.detect</code>	A Boolean property. When set to <code>true</code> it indicates that a connection is considered abandoned and eligible for removal if it has been idle longer than the <code>db.pool.abandoned.time</code> .	<code>false</code>
<code>db.pool.abandoned.time</code>	The time in seconds before an abandoned connection can be removed.	300

The `db.pool.max` property is the most important. By default, each Alfresco instance is configured to use up to a maximum of 275. All operations in Alfresco require a database connection, which places a hard upper limit on the amount of concurrent requests a single Alfresco instance can service (that is, 40), from all protocols, by default.

Most Java application servers have higher default settings for concurrent access (Tomcat allows up to 200 concurrent HTTP requests by default). Coupled with other threads in Alfresco (non-HTTP protocol threads, background jobs, and so on) this can quickly result in excessive contention for database connections within Alfresco, manifesting as poor performance for users.

If you are using Alfresco in anything other than a single-user evaluation mode, increase the maximum size of the database connection pool to at least the following setting.

```
[number of application server worker threads] + 75.
```

For a Tomcat default HTTP worker thread configuration, and with all other Alfresco thread pools left at the defaults, this means this property should be set to at least 275.

To increase the database connection pool, add the `db.pool.max` property to the `alfresco.global.properties` file and set it to the recommended value of 275, for example:

```
db.pool.max=275
```

For clarity, add this property immediately after the other database properties.

 After increasing the size of the Alfresco database connection pools, you must also increase the number of concurrent connections your database can handle to at least the size of the cumulative Alfresco connection pools. In a cluster, each node has its own independent database connection pool. You must configure sufficient database connections for all of the Alfresco cluster nodes to be able to connect simultaneously. Alfresco recommends that you configure at least 10 more connections to the database than are configured cumulatively across all of the Alfresco connection pools to ensure that you can still connect to the database even if Alfresco saturates its own connection pools. Remember to factor in cluster nodes (which can each use up to 275 database connections) as well as connections required by other applications that are using the same database server as Alfresco.

The precise mechanism for reconfiguring your database's connection limit depends on the relational database product you are using; contact your DBA for configuration details.

Configuring OpenOffice subsystem

Within Alfresco, you can transform a document from one format to another. This feature requires you to install LibreOffice or OpenOffice.org.

OOoJODconverter

The JODConverter integration, which is a library that improves the stability and performance of OpenOffice.org or LibreOffice within Alfresco. The OOoJODConverter runs on the same machine as the Alfresco server and it supports:

- a pool of separate OpenOffice processes
- automatic restart of crashed OpenOffice processes
- automatic termination of slow OpenOffice operations
- automatic restart of any OpenOffice process after a number of operations (this is a workaround for OpenOffice memory leaks)

OOoDirect

The subsystem for direct OpenOffice integration, in which the Alfresco server manages OpenOffice directly. To enable or disable this subsystem, use the following property:

```
ooo.enabled=false
```



If you install Alfresco manually, by default, the OOoDirect subsystem is enabled, and the OOoJodconverter subsystem is disabled. Although it is possible to run both subsystems, Alfresco recommends that you enable only one at a time. To take advantage of the stability and performance benefits of the OOoJodconverter subsystem, ensure that you disable OOoDirect and enable OOoJodConverter using the following properties in the `alfresco-global.properties` file:

```
ooo.enabled=false
jodconverter.enabled=true
```

Changing the Office subsystems

When you install Alfresco using the setup wizards, the default subsystem for OpenOffice transformations is OOoJodconverter. Alfresco also supports the OOoDirect subsystem.

The JODConverter subsystem requires OpenOffice.org 3.0.0 or later and recommends 3.1.0+.

You can change the Office subsystem using the following ways:

- Alfresco Admin Console
- Runtime administration using your JMX client
- Modifying the `alfresco-global.properties` file

Alfresco Admin Console

You can also change which Office subsystem is enabled on the Alfresco Admin Console.

JMX interface runtime administration

1. Open your JMX client, for example, JConsole.
2. Locate the **OOoDirect** subsystem.
3. Edit the **ooo.enabled** value to `false`.
4. Restart the subsystem.
5. Locate the **OOoJodconverter** subsystem.
6. Edit the **jodconverter.enabled** value to `true`.
7. Restart the subsystem.

Global properties file

1. Open the `alfresco-global.properties` file.

2. Edit the following lines:

```
ooo.enabled=false
jodconverter.enabled=true
```

3. Save the file.
4. Restart the Alfresco server.

Configuring transformations in the global properties file

The subsystem for OpenOffice transformations is called OOoDirect. Using this direct OpenOffice integration, the Alfresco server manages OpenOffice directly.

1. Open the `alfresco-global.properties` file.
2. Set the `ooo.exe` property to the path of the OpenOffice.org or LibreOffice installation.
3. Ensure that the following line is set to true:

```
ooo.enabled=true
```

4. Save the file.
5. Restart the Alfresco server.

OOoDirect subsystem configuration properties

The following properties can be configured for the OOoDirect subsystem.

ooo.exe

Specifies the LibreOffice installation path.

ooo.enabled

Enables or disables the OOoDirect subsystem.

OOoJodconverter subsystem configuration properties

The following properties can be configured for the OOoJodconverter subsystem.

jodconverter.enabled

Enables or disables the Jodconverter process(es).

jodconverter.maxTasksPerProcess

Specifies the number of transforms before the process restarts. The default is 200.

jodconverter.officeHome

Specifies the name of the LibreOffice install directory. The following are examples of install directory paths:

- Linux: `jodconverter.officeHome=/Applications/alfresco/libreoffice.app/Contents`
- Windows: `jodconverter.officeHome=c:/Alfresco/LibreOffice.org`

jodconverter.portNumbers

Specifies the port numbers used by each processing thread. The number of process will match the number of ports. The default numbers are 2022, 2023, and 2024.

jodconverter.taskExecutionTimeout

Specifies the maximum number of milliseconds that an operation is allowed to run before it is aborted. It is used to recover from operations that have hung. The default is 120000 milliseconds (2 minutes).

jodconverter.taskQueueTimeout

Specifies the maximum number of milliseconds a task waits in the transformation queue before the process restarts. It is used to recover hung LibreOffice processes. The default is 30000 milliseconds (30 seconds).

Configuring synchronization

The synchronization subsystem manages the synchronization of Alfresco with all the user registries (LDAP servers) in the authentication chain.

The synchronization subsystem supports three modes of synchronization:

Full

All users and groups are queried, regardless of when they were last modified. All local copies of these users and groups already existing are then updated and new copies are made of new users and groups. Since processing all users and groups in this manner can be fairly time consuming, this mode of synchronization is usually only triggered on the very first sync when the subsystem first starts up. However, synchronization can also be triggered in this mode by the scheduled synchronization job, if `synchronization.synchronizeChangesOnly` is set to false.

Differential

Only those users and groups changed since the last query are queried and created/updated locally. This differential mode is much faster than full synchronization. By default, it is triggered when the subsystem starts up after the first time and also when a user is successfully authenticated who does not yet have a local person object in Alfresco. This means that new users, and their group information, are pulled over from LDAP servers as and when required with minimal overhead.

Differential With Removals

All users and groups are queried to determine which ones no longer exist and can be disabled or deleted locally. In order to synchronize the attributes of the remaining users and groups, a differential sync is performed so only those users and groups that have changed since the last sync are updated or added locally.

Synchronization triggers

Synchronization can be triggered by each of the following events:

Startup

On system startup or restart of the Synchronization subsystem, a differential sync is triggered (unless disabled with configuration).

Authentication

On successful authentication of a user who does not yet exist locally, a differential sync is triggered (unless disabled with configuration).

Schedule

A scheduled job triggers synchronization in differential with removals mode every 24 hours. This can instead be scheduled in full mode if you set the `synchronization.synchronizeChangesOnly` property to false. The scheduling of this job can also be altered.

Synchronization deletion

Users and groups removed from the LDAP directory or query are only identified when synchronization is triggered by the schedule job in either full mode or differential with removals mode.

Users and groups in Alfresco created as a result of a synchronization operation are tagged with an originating zone ID. This records the ID of the authentication subsystem instance that the user or group was queried from. On synchronization with a zone, only those users and groups tagged with that zone are candidates for deletion from Alfresco. This avoids accidental deletion of built-in groups, such as `ALFRESCO_ADMINISTRATORS`.

When a removed user or group is detected, Alfresco will behave in one of two ways, depending on the value of the `synchronization.allowDeletions` property. When `true` (the default value), Alfresco simply deletes the user or group from the local repository. When `false`, the user or group is simply untagged from its zone, thus converting it to an Alfresco local user or group. A removed user also loses its memberships from any of the LDAP groups they were in, whereas, a removed group is cleared of all their members. As the user or group is retained in the Alfresco repository, this setting has the advantage that the site memberships for that user or group are remembered, should they later be reactivated.

Collision resolution

If there are overlaps between the contents of two user registries in the authentication chain (for example, where two user registries both contain a user with the same user name), then the registry that occurs earlier in the authentication chain will be given precedence. This means that exactly the same order of precedence used during authentication will be used during synchronization.

For example, if user `A` is queried from zone `z1` but already exists in Alfresco in zone `z2`:

- `A` is ignored if `z1` is later in the authentication chain than `z2`
- `A` is moved to `z1` if `z2` does not exist in the authentication chain or `z1` is earlier in the authentication chain and the `synchronization.allowDeletions` property is `false`.
- `A` is deleted from `z2` and recreated in `z1` if `z1` is earlier in the authentication chain and the `synchronization.allowDeletions` property is `true`.

Synchronization configuration properties

The synchronization subsystem manages synchronization of Alfresco by configuring the subsystem's properties.

The following properties can be configured for the synchronization subsystem.

synchronization.synchronizeChangesOnly

Specifies whether the scheduled synchronization job is run in differential mode. The default is `true`, which means that the scheduled sync job is run in differential mode (rather than full mode). Regardless of this setting a differential sync can still be triggered when a user who does not yet exist in Alfresco is successfully authenticated.

synchronization.allowDeletions

Specifies if deletion of local users and groups is allowed. See the information about [Synchronization deletion](#) and [Collision resolution](#) for the circumstances under which this can happen. The default is `true`. If `false`, then no sync job will be allowed to delete users or groups during the handling of removals or collision resolution.

synchronization.import.cron

Specifies a cron expression defining when the scheduled synchronization job should run, by default at midnight every day.

synchronization.syncOnStartup

Specifies whether to trigger a differential sync when the subsystem starts up. The default is `true`. This ensures that when user registries are first configured, the bulk of the synchronization work is done on server startup, rather than on the first login.

synchronization.syncWhenMissingPeopleLogIn

Specifies whether to trigger a differential sync when a user, who does not yet exist in Alfresco, is successfully authenticated. The default is `true`. If there are users created in the LDAP server that do not already exist in Alfresco, when you start Alfresco, a differential synchronization is triggered.

synchronization.autoCreatePeopleOnLogin

Specifies whether to create a user with default properties when a user is successfully authenticated, who does not yet exist in Alfresco, and was not returned by a differential sync (if enabled with the specified property). The default is true. Setting this to false allows you to restrict Alfresco to a subset of those users who could be authenticated by LDAP; only those created by synchronization are allowed to log in. You can control the set of users in this more restricted set by overriding the user query properties of the LDAP authentication subsystem.

Configuring email

The email subsystem allows you to configure the outbound and inbound SMTP email settings to interact with Alfresco.

There are two methods of running Alfresco email server:

- Running the email server process in the same JVM context as the repository
- Running the email server remotely and communicate with the repository using Remote Method Invocation (RMI)

Admin Console: Managing inbound emails

Set these inbound email properties to activate sending and receiving site invites, and also for receiving activity notification emails.

1. Open the Admin Console.
2. In the **Email Services** section, click **Inbound Email**.

You see the **Inbound Email** page.

3. Set the email properties:

Inbound Email property	Example setting	What is it?
Enabled	No	Use check box to enable or disable the inbound email service. By default, it is not enabled.
Unknown User	anonymous	This is the user name to authenticate as when the sender address is not recognized.
Allowed Senders	.*	To allow senders, enter a comma-separated list of email REGEX patterns of allowed senders. If there are any values in the list, then all sender email addresses must match. For example:.* \@alfresco\.com, .*@alfresco \.org.
Overwrite Duplicates	Yes	By default, duplicate messages to a folder will overwrite each other. Deselect this check box to keep duplicate messages and apply a unique number.

Inbound Email property	Example setting	What is it?
Maximum Server Connections	3	This provides the maximum number of connections allowed in order to control the performance of the system. To prioritize the email subsystem higher, increase this number. The default setting is 3.
SMTP Authentication Enabled	No	Use this check box to enable or disable the authentication of inbound email against the repository.
Email Server Port	25	This is the default port number for the email server.
Email Server Domain	alfresco.com	This is the default domain for the email server.
Blocked Senders		To block senders, enter a comma-separated list of email REGEX patterns, for example: .*@hotmail\.com, .*@googlemail\.com. If the sender email address matches a listed value, then the message will be rejected.
Email Authentication Group	EMAIL_CONTRIBUTORS	This is the name of the group in which users must be a member to add content to the repository by email. The default group is EMAIL_CONTRIBUTORS.
Transport Layer Security (TLS)	Enabled	This enables the TLS protocol, which upgrades a plain text connection to an encrypted TLS or SSL connection instead of using a separate port for encrypted communication. Select the TLS support setting: <ul style="list-style-type: none"> • Disabled: TLS support is disabled • Hidden: On the EHLO command, server support for TLS is hidden, though TLS will still be accepted if the client uses it • Enabled: On the EHLO command, server support for TLS is announced • Required: TLS authentication is required

4. Click **Save** to apply the changes you have made to the properties.

If you do not want to save the changes, click **Cancel**.

Admin Console: Managing outbound emails

1. Open the Admin Console.
2. In the **Email Services** section, click **Outbound Email**.
You see the **Outbound Email** page.
3. Set the email properties:

Outbound Email property	Example setting	What is it?
Hostname	smtp.example.com	This is the name of the SMTP(S) host server.
Encoding	UTF-8	This is the email encoding type. The default is UTF-8.
Editable Sender Address		This check box enables the From field in outbound emails to be edited to differ from the Default Sender's Address. When you deselect this check box, the Default Sender's Address is always used. You should deselect this option if your email server security settings require the From field to match the user name used for email server authentication.
Email Server Port	25	This is the default port number for the email server.
Default Sender's Address	alfresco@demo.alfresco.org	The default address that is used in the From field of outbound emails if no alternative is available.
Email Protocol	SMTP	Select a protocol from the list. This is the protocol that will be used when sending email.
Username	anonymous	The account user name that connects to the SMTP server. The user name and password are only required if your server requires them for authentication.
Password		The account user password.

4. Click **Save** to apply the changes you have made to the properties.
If you do not want to save the changes, click **Cancel**.

OutboundSMTP configuration properties

The following properties can be configured for the OutboundSMTP subsystem type.



You must set the Outbound email configuration for Share invitations to work correctly. If you do not set the email configuration, when you invite a user to a site, the user will not receive the assigned task notification.

The email service is exposed as a spring bean called mailService, which is contained within the OutboundSMTP subsystem.

Configure the Alfresco repository to send emails to an external SMTP server by overriding the default settings. Set the email property overrides in the `alfresco-global.properties` file.

The following properties can be configured for the OutboundSMTP subsystem type.

mail.host=yourmailhost.com

Specifies the host name of the SMTP host, that is, the host name or IP address of the server to which email should be sent.

mail.port

Specifies the port number on which the SMTP service runs (the default is 25). By convention, the TCP port number 25 is reserved for SMTP, but this can be changed by your email administrator.

mail.username

Specifies the user name of the account that connects to the smtp server.

mail.password

Specifies the password for the user name used in `mail.username`.

mail.encoding

Specifies UTF-8 encoding for email. Set this value to UTF-8 or similar if encoding of email messages is required.

mail.from.default

Specifies the email address from which email notifications are sent. This setting is for emails that are not triggered by a user, for example, feed notification emails. If the current user does not have an email address, this property is used for the `from` field by the `MailActionExecutor`.

mail.from.enabled

If this property is set to false, then the value set in `mail.from.default` is always used. If this property is set to true, then the `from` field may be changed. This property may be required if your email server security settings insist on matching the `from` field with the authentication details.

mail.protocol

Specifies which protocol to use for sending email. The value can be either `smtp` or `smtps`.

mail.header

Optionally specifies the content transfer encoding for the message. If specified the **Content-Transfer-Encoding** is set to the value you specify.

The following properties are for SMTP.

mail.smtp.auth

Specifies if authentication is required or not. Specifies the use of SMTPS authentication. If true, attempt to authenticate the user using the `AUTH` command. Defaults to false.

mail.smtp.timeout

Specifies the timeout in milliseconds for SMTP.

mail.smtp.starttls.enable

Specifies if the transport layer security needs to be enabled or not. Specifies the use of `STARTTLS` command. If true, enables the use of the `STARTTLS` command to switch the connection to a TLS-protected connection before issuing any login commands. Defaults to false.

mail.smtp.debug

Specifies if debugging SMTP is required or not.

The following properties are for SMTPS.

mail.smtps.starttls.enable

Specifies if the transport layer security for smtps needs to be enabled or not.

mail.smtps.auth

Specifies if authentication for smtps is required or not.

The following properties can be set to define a test message when the subsystem starts.

mail.testmessage.send

Defines whether or not to send a test message.

mail.testmessage.to

Specifies the recipient of the test message.

mail.testmessage.subject

Specifies the message subject of the test message.

mail.testmessage.text

Specifies the message body of the test message.

The following property is for setting the email site invitation behavior.

notification.email.siteinvite

You must set the outbound email configuration for Share invitations to work correctly. This property allows you to control whether or not emails are sent out for site invites. If you have not configured the outbound email properties, set this property to false.

The following examples show which properties to set for two different email clients. Add these properties to the `alfresco-global.properties` file.

The following example shows the properties that you need to set to configure Gmail with Alfresco.

```
# Sample Gmail settings
mail.host=smtp.gmail.com
mail.port=465
mail.username=user@gmail.com
mail.password=password
mail.protocol=smtps
mail.smtps.starttls.enable=true
mail.smtps.auth=true
```

The following example shows the properties that you need to set to configure Zimbra with Alfresco.

```
# Sample Zimbra settings
Not authenticated.

mail.host=zimbra.<your company>
mail.port=25
mail.username=anonymous
mail.password=
# Set this value to UTF-8 or similar for encoding of email
  messages as required
mail.encoding=UTF-8
# Set this value to 7bit or similar for Asian encoding of email
  headers as required
mail.header=
mail.from.default=<default from address>
mail.smtp.auth=false
mail.smtp.timeout=30000
```

InboundSMTP configuration properties

The InboundSMTP email subsystem type allows you to configure the behavior of the email server and service.

The following properties can be set for Inbound SMTP email in the `alfresco.global.properties` file.

email.inbound.unknownUser=anonymous

Specifies the user name to authenticate as when the sender address is not recognized.

email.inbound.enabled=true

Enables or disables the inbound email service. The service could be used by processes other than the email server (for example, direct RMI access), so this flag is independent of the email service.

email.server.enabled=true

Enables the email server.

email.server.port=25

Specifies the default port number for the email server.

email.server.domain=alfresco.com

Specifies the default domain for the email server.

email.server.allowed.senders=.*

Provides a comma-separated list of email REGEX patterns of allowed senders. If there are any values in the list, then all sender email addresses must match. For example: `.*\@alfresco\.com, .*\@alfresco\.org`.

email.server.blocked.senders=

Provides a comma-separated list of email REGEX patterns of blocked senders. If the sender email address matches this, then the message will be rejected. For example: `.*\@hotmail\.com, .*\@googlemail\.com`.

Handling messages by target node type

This section describes the default behaviors for incoming email to different types of referenced nodes.

You can modify or extend the default behaviors by adding in custom handlers.

Folder(Space)

Content added with emailed aspect.

Forum(Discussion)

Content specialized to Post with emailed aspect; if email subject matches a topic, then add to topic, otherwise create new topic based on subject.

Topic(Post)

Content specialized to Post with emailed aspect; if referenced node is a Post, add to Post's parent Topic.

Document(Content)

If discussion exists, same as for forums, otherwise add discussion with email as initial topic and post.

Groups and permissions for email

An email arriving at the Alfresco email server is unauthenticated. An authentication group, `EMAIL_CONTRIBUTORS`, must exist to allow permissions to be handled at a high level by the administrator.

When an email comes into the system, the only identification is the sender's email address. The user is looked up based on the email address.

- If a matching user is not found, then the current user is assumed to be unknown, if unknown exists
- If unknown does not exist, then the email is rejected as authentication will not be possible

- If the user selected is not part of email contributor's group, then the email is rejected

The current request's user is set and all subsequent processes are run as the authenticated user. If any type of authentication error is generated, then the email is rejected. The authentication will also imply that the authenticated user may not have visibility of the target node, in which case the email is also rejected. Effectively, this means that the target recipient of the email does not exist, at least not for the sender.

The current default server configuration creates the `EMAIL_CONTRIBUTORS` group and adds the `admin` user to this group.

Configuring file servers

 Alfresco recommends that you implement an allowed authentication mechanism relative to the file server you are using. For more information on the different types of authentication subsystems in Alfresco and their use, see [Authentication subsystem types](#).

As with other Alfresco subsystems, the File Server subsystem exposes all of its configuration options as properties that can be controlled through a JMX interface or the global properties file.

Configuring SMB/CIFS server

The server includes Java socket-based implementations of the SMB/CIFS protocol that can be used on any platform.

The server can listen for SMB traffic over the TCP protocol (native SMB) supported by Windows 2000 and later versions, and the NetBIOS over TCP (NBT) protocol, supported by all Windows versions. There is also a Windows-specific interface that uses Win32 NetBIOS API calls using JNI code. This allows the Alfresco CIFS server to run alongside the native Windows file server.

The default configuration uses the JNI-based code under Windows and the Java socket based code under Linux, Solaris, and Mac OS X.

CIFS file server properties

The following properties can be configured for the SMB/CIFS server.

cifs.enabled

Enables or disables the CIFS server.

cifs.serverName

Specifies the host name for the Alfresco CIFS server. This can be a maximum of 16 characters and must be unique on the network. Use the special token `{localname}` in place of the local server's host name and you can generate a unique name by prepending/ appending to it.

On Windows systems, the value of this property must be different from the server's host name, it should resolve to the same IP address as the server, and must be different from any other host name on the network.

cifs.domain

An optional property. When not empty, specifies the domain or workgroup to which the server belongs. This defaults to the domain/workgroup of the server, if not specified.

cifs.hostannounce

Enables announcement of the CIFS server to the local domain/workgroup so that it shows up in Network Places/Network Neighborhood.

cifs.sessionTimeout

Specifies the CIFS session timeout value in seconds. The default session timeout is 15 minutes. If no I/O occurs on the session within this time then the session will be closed by the server. Windows clients send keep-alive requests, usually within 15 minutes.

Java-based SMB properties

The following properties will only take effect on non-Windows servers, where the Java-based SMB implementation is used.

cifs.broadcast

Specifies the broadcast mask for the network.

cifs.bindto

Specifies the network adapter to which to bind. If not specified, the server will bind to all available adapters/addresses.

cifs.tcpipSMB.port

Controls the port used to listen for the SMB over TCP/IP protocol (or native SMB), supported by Win2000 and above clients. The default port is 445.

cifs.ipv6.enabled

Enables the use of IP v6 in addition to IP v4 for native SMB. When `true`, the server will listen for incoming connections on IPv6 and IPv4 sockets.

cifs.netBIOSMB.namePort

Controls the NetBIOS name server port on which to listen. The default is 137.

cifs.netBIOSMB.datagramPort

Controls the NetBIOS datagram port. The default is 138.

cifs.netBIOSMB.sessionPort

Controls the NetBIOS session port on which to listen for incoming session requests. The default is 139.

cifs.WINS.autoDetectEnabled

When `true` causes the `cifs.WINS.primary` and `cifs.WINS.secondary` properties to be ignored.

cifs.WINS.primary

Specifies a primary WINS server with which to register the server name.

cifs.WINS.secondary

Specifies a secondary WINS server with which to register the server name.

cifs.disableNIO

Disables the new NIO-based CIFS server code and reverts to using the older socket based code.

Running SMB/CIFS from a normal user account

The CIFS server can be configured to run using non-privileged ports and then use firewall rules to forward requests from the privileged ports to the non-privileged ports.

1. To configure the CIFS server to use non-privileged ports, use the following property settings:

```
cifs.tcpipSMB.port=1445
cifs.netBIOSMB.namePort=1137
cifs.netBIOSMB.datagramPort=1138
cifs.netBIOSMB.sessionPort=1139
```

Other port numbers can be used but must be above 1024 to be in the non-privileged range.

The firewall rules should then be set up to forward requests:

- TCP ports 139/445 to TCP 1139/1445
- UDP ports 137/138 to UDP 1137/1138

2. On Mac OS X the following commands can be used:

```
sysctl -w net.inet.ip.fw.enable=1
```

```

sysctl -w net.inet.ip.forwarding=1
sysctl -w net.inet.ip.fw.verbose=1
sysctl -w net.inet.ip.fw.debug=0
ipfw flush
ipfw add 100 allow ip from any to any via lo0
# Forward native SMB and NetBIOS sessions to non-privileged ports
ipfw add 200 fwd <local-ip>,1445 tcp from any to me dst-port 445
ipfw add 300 fwd <local-ip>,1139 tcp from any to me dst-port 139
# Forward NetBIOS datagrams to non-privileged ports (does not currently
work)
ipfw add 400 fwd <local-ip>,1137 udp from any to me dst-port 137
ipfw add 500 fwd <local-ip>,1138 udp from any to me dst-port 138

```

Replace <local-ip> with the IP address of the server that Alfresco is running on.

3. On Linux, the following commands can be used to get started, but be aware these commands will delete all existing firewall and NAT rules and could be a security risk:

```

echo 1 > /proc/sys/net/ipv4/ip_forward
modprobe iptable_nat
iptables -F
iptables -t nat -F
iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT
iptables -t nat -A PREROUTING -p tcp --dport 445 -j REDIRECT --to-ports
1445
iptables -t nat -A PREROUTING -p tcp --dport 139 -j REDIRECT --to-ports
1139
iptables -t nat -A PREROUTING -p udp --dport 137 -j REDIRECT --to-ports
1137
iptables -t nat -A PREROUTING -p udp --dport 138 -j REDIRECT --to-ports
1138

```

The UDP forwarding does not work, which affects the NetBIOS name lookups. A workaround is either to add a DNS entry matching the CIFS server name and/or add a static WINS mapping, or add an entry to the clients `LMHOSTS` file.

SMB/CIFS advanced Spring overrides

The SMB/CIFS server beans are declared in the `file-servers-context.xml` file. Using the subsystem extension classpath mechanism, you can place site specific customization of these default values in a Spring bean file in `<extension>\subsystems\fileServers\default\default\custom-file-servers-context.xml` (note that the `default\default` part of the path is intentional).

The main bean that drives the CIFS server configuration is called `cifsServerConfig`. This has several properties that can be populated with child beans that control various optional SMB implementations.

tcpipSMB

Controls the Java-based SMB over TCP/IP implementation, which is compatible with Windows 2000 clients and later.

netBIOSsmb

Controls the Java-based NetBIOS over TCP/IP implementation, which is compatible with all Windows clients.

win32NetBIOS

Controls the JNI-based NetBIOS over TCP/IP implementation, which is only enabled for Alfresco servers running on Windows.

When one of the specified properties is not set, it deactivates support for the corresponding protocol implementation. The `tcpipSMB` and `netBIOSsmb` beans have a `platforms` property that allows their configuration to be targeted to Alfresco servers running on specific platforms. The

property is formatted as a comma-separated list of platform identifiers. Valid platform identifiers are `linux`, `solaris`, `macosx`, and `aix`.

1. The `serverComment` property of the `cifsServerConfig` bean controls the comment that is displayed in various information windows.
2. Use the following steps for troubleshooting CIFS.
 - a. The `sessionDebugFlags` property of the `cifsServerConfig` bean enables debug output levels for CIFS server debugging. The value should be in the form of a comma-separated list of the flag names.

Flag	Description
NetBIOS	NetBIOS layer
State	Session state changes
Tree	File system connection/disconnection
Search	Folder searches
Info	File information requests
File	File open/close
FileIO	File read/write
Tran	Transaction requests
Echo	Echo requests
Errors	Responses returning an error status
IPC	IPC\$ named pipe
Lock	File byte range lock/unlock
Pktttype	Received packet type
Dcerpc	DCE/RPC requests
Statecache	File state caching
Notify	Change notifications
Streams	NTFS streams
Socket	NetBIOS/native SMB socket connections
PktPool	Memory pool allocations/de-allocations
PktStats	Memory pool statistics dumped at server shutdown
ThreadPool	Thread pool

- b. The `log4j.properties` file must also have SMB/CIFS protocol debug output enabled using:

```
log4j.logger.org.alfresco.smb.protocol=debug
```

- c. The following logging level must also be enabled to log debug output from the core file server code:

```
log4j.logger.org.alfresco.fileserver=debug
```

Configuring CIFS on Windows Server 2008 R2

The following instructions describe how to configure the Alfresco CIFS server on Windows Server 2008 R2.



Alfresco does **not** recommend that you use the CIFS file Server on an Alfresco installation running on Windows. Due to limitations and workarounds necessary for the operating

system, it requires a complicated setup and provides poor performance compared to non-Windows systems.

1. Install Windows Server 2008 R2 out-of-the box.
 -  To use these instructions, you must not have altered the hosts file on the server or client. Also, you must not have modified the Windows Registry either on the server or client. You do not need to change the hosts file or File and Printer Sharing configuration.
2. Configure a WINS server.
 - a. If the server is a domain controller or already part of a domain, this might already be controlled by a Domain Policy.
To install one on Windows Server 2008 R2, see the following article: <http://technet.microsoft.com/en-us/library/ff710463%28WS.10%29.aspx>.
 - b. To manually configure an existing WINS server:
 1. Go to **Control Panel\Network and Internet\Network and Sharing Center > Change Adapter Settings > Local Area Connection > Properties**.
 2. Select **Internet Protocol Version 4 (TCP/IPv4)** and then select **Properties**.
 3. On the **General** tab, select **Advanced** and then select the **WINS** tab.
 4. Click **Add** and then add the IP address of the WINS server in your network and select **Enable NetBIOS over TCP/IP**.
 5. Click **OK > OK > Close**.
3. Ensure that you install Alfresco using the x64 setup wizard.
See [Installing Alfresco Enterprise on Windows](#).
4. Configure the Windows Server 2008 R2 firewall to create a rule to block 445.
 - a. Open **Control Panel\Network and Internet\Network and Sharing Center > Windows Firewall > Advanced Settings**.
 - b. Select **Inbound Rules**.
 - c. On the right-side of the window, click **New Rule**.
 - d. Follow the instructions on the wizard:
 1. Rule Type > Port, Next.
 2. Rule apply to "TCP", Specific Local Ports > 445, Next,
 3. Action > Block the connection, Next,
 4. Profile > Select ALL network types (Domain, Public, Private)
 5. Name > "Alfresco CIFS (Block 445)", Description the same.
 - e. Select **Finish**.
5. Configure the Windows Server 2008 R2 firewall to create a rule to allow 137,138,139.
 - a. Open the **Control Panel\Network and Internet\Network and Sharing Center > Windows Firewall > Advanced Settings**.
 - b. Select **Inbound Rules**.
 - c. On the right-side of the window, click **New Rule**.
 - d. Follow the instructions on the wizard:
 1. Rule Type > Port, Next.
 2. Rule apply to "TCP", Specific Local Ports > 137,138,139, Next,
 3. Action > Allow the connection, Next,

4. Profile > Select ALL network types (Domain, Public, Private)
5. Name > "Alfresco CIFS (Allow 137,138,139)", Description the same.
- e. Select **Finish**.
6. Configure the client (Windows XP and Windows 7)
 - a. Go to **Control Panel\Network and Internet\Network and Sharing Center > Change Adapter Settings > Local Area Connection > Properties**.
 - b. Select **Internet Protocol Version 4 (TCP/IPv4)** and click **Properties**.
 - c. On the **General** tab, select **Advanced** and then select the **WINS** tab.
 - d. Click **Add** and then add the IP address of the WINS server in your network and select **Enable NetBIOS over TCP/IP**.
 - e. Click **OK > OK > Close**.
 - f. Use the `net use R: \\{HOSTNAME}\A\Alfresco * /USER:admin` command to check your connection.
If the WINS server works correctly, you are then connected to Alfresco CIFS successfully.

Additional information for CIFS on Windows

This section provides additional information to assist you when setting up CIFS servers on Windows.

CIFS on Windows works only with NetBIOS.

The process CIFS uses on a supported Windows installation is:

- The client sends a request to the CIFS server.
- If the client wants to access a path that starts with the Windows server name, then the CIFS request will be handled by Windows CIFS.
- If the path starts with the Alfresco CIFS server name, then the CIFS request will be handled by Alfresco CIFS.

The dispatching is made at the Windows-level by the NetBIOS Windows DLLs, however this dispatching is not available with native CIFS (port 445).

If you leave port 445 open, requests aimed at Alfresco CIFS are routed to Windows CIFS and will fail. A CIFS client does not know in advance if a CIFS server listens on NetBIOS ports (137, 138, 139) or native CIFS port (445). It typically sends two connections requests: one to the NetBIOS ports and one to the native CIFS port. The faster request wins and, as native CIFS is typically faster, the connection is likely to fail.

 The Java CIFS code that Alfresco supports on Linux is not supported on Windows.

The drawback of using CIFS on a Windows server is performance degradation.

The supported process of using CIFS on Windows forces the clients to use NetBIOS to talk to Alfresco. NetBIOS is a protocol that is much less efficient and more chatty than the more recent native CIFS (port 445) protocol. An Alfresco CIFS setup on Windows will suffer performance issues when compared to a Linux/Unix system due to this chattiness.

Configuring the FTP file server

This section describes how to configure the FTP file server.

FTP file server properties

The following properties can be configured for the FTP server.

ftp.enabled

Enables or disables the FTP server.

ftp.port

Specifies the port that the FTP server listens for incoming connections on. Defaults to port 21. On some platforms ports below 1024 require the server to be run under a privileged account.

ftp.bindto

Specifies the network adapter to bind with. If the network adapter is not specified, the server will bind to all the available adapters/addresses.

ftp.sessionDebug

Enable debug output by setting the SSL debug flag using `ftp.sessionDebug=SSL`, and also by enabling the `log4j.logger.org.alfresco.fileserver=debug` log4j output.

ftp.dataPortFrom

Limits the data ports to a specific range of ports. This property sets the lower limit.

ftp.dataPortTo

Limits the data ports to a specific range of ports. This property sets the upper limit.

ftp.keyStore

Specifies the path to the `keystore` file for FTPS support.

ftp.keyStoreType

Specifies the file type of the `keystore` file. The default is JKS.

ftp.keyStorePassphrase

Specifies the passphrase for the `keystore` file.

ftp.trustStore

Specifies the path to the `truststore` file for FTPS support.

ftp.trustStoreType

Specifies the file type of the `truststore` file. The default is JKS.

ftp.trustStorePassphrase

Specifies the passphrase for the `truststore` file.

ftp.requireSecureSession

Specifies whether only secure FTPS sessions will be allowed to log in to the FTP server. To force all connections to use FTPS, set `ftp.requireSecureSession=true`.

ftp.sslEngineDebug

Specifies the FTP session debug flags, which enables additional debug output from the Java SSLEngine class. The list of values can be STATE, RXDATA, TXDATA, DUMPDATA, SEARCH, INFO, FILE, FILEIO, ERROR, PKTTYPER, TIMING, DATAPORT, DIRECTORY, SSL.

If you have IPv6 enabled on your system, Alfresco automatically uses IPv6.

The FTPS support runs over the same socket as normal connections; the connection is switched into SSL mode at the request of the client, usually before the user name and password is sent. The client can switch the socket back to plain text mode using the `ccc` command.

The `ftp.keyStore`, `ftp.trustStore`, and respective `ftp.keyStorePassphrase` and `ftp.trustStorePassphrase` values must all be specified to enable FTPS support. Only explicit FTP over SSL/TLS mode is supported. Encrypted data sessions are not supported.

To setup the `keystore` and `truststore` files, follow the instructions from the Java6 JSSE Reference Guide. This will provide the values required for the `ftp.keyStore`, `ftp.trustStore`, `ftp.keyStorePassphrase` and `ftp.trustStorePassphrase` values.

FTP advanced Spring overrides

The FTP server beans are declared in the `file-servers-context.xml` file. Using the subsystem extension classpath mechanism, site specific customization of these default values can be placed in a Spring bean file in `<extension>\subsystems\fileServers\default\default\custom-file-servers-context.xml` (note that the `default\default` part of the path is intentional).

The following properties can be overridden on the `ftpServerConfig` bean.

bindTo

Specifies the address the FTP server binds to, if not specified the server will bind to all available addresses.

rootDirectory

Specifies the path of the root directory as an FTP format path, that is, using forward slashes. The first part of the path should be the file system name, optionally followed by one or more folder names, for example:

```
/Alfresco/myfolder/
```

charSet

Specifies the character set to be used. The character set name should be a valid Java character set, see the Java `CharSet` class for more information.

1. The `debugFlags` property enables debug output levels for FTP server debugging. The value should be a comma-separated list of flag names from the following table:

Flag	Description
State	Session state changes
Search	Folder searches
Info	File information requests
File	File open/close
FileIO	File read/write
Error	Errors
Pkttype	Received packet type
Timing	Time packet processing
Dataport	Data port
Directory	Directory commands

2. Configure logging levels for the FTP server in `$ALF_HOME/tomcat/webapps/alfresco/WEB-INF/classes/log4j.properties` using:

```
log4j.logger.org.alfresco.ftp.protocol=debug
log4j.logger.org.alfresco.ftp.server=debug
```

Configuring the NFS file server

It is recommended that TCP connections are used to connect to the Alfresco NFS server. Using a read/write size of 32 KB will also help to optimize the performance.

NFS file server properties

The following properties can be configured for the NFS server.

nfs.enabled

Enables or disables the NFS server.

nfs.user.mappings

A composite property that configures the user ID/group ID to the Alfresco user name mappings that are used by the current RPC authentication implementation.

For example, the following configuration gives `admin` a `uid` and `gid` of 0 and `auser` a `uid` and `gid` of 501.

```
nfs.user.mappings=admin,auser
nfs.user.mappings.value.admin.uid=0
nfs.user.mappings.value.admin.gid=0
nfs.user.mappings.value.auser.uid=501
nfs.user.mappings.value.auser.gid=501
```

NFS advanced Spring overrides

The following properties can be overridden on the `nfsServerConfig` bean.

portMapperEnabled

Enables the built-in portmapper service. This would usually be enabled on Windows where there is no default portmapper service. Under Linux/Unix operating systems, the built-in portmapper service can be used, which also saves having to run the Alfresco server using the root account.

threadPool

Sets the size of the RPC processing thread pool. The minimum number of threads is 4, the default setting is 8.

packetPool

Sets the size of the packet pool used to receive RPC requests and send RPC replies. The minimum number of packets is 10, the default setting is 50.

portMapperPort

The port number to run the portmapper service on. The default port is 111.

mountServerPort

The port number to run the mountserver service on. The default is to allocate an available non-privileged port.

nfsServerPort

The port number to run main NFS server service on. The default is to allocate the default NFS port: 2049. This will likely clash with a running native NFS server.

1. The `debugFlags` property enables debug output levels for NFS server debugging. The value should be in the form of a comma-separated list of the flag names in the following table.

Flag	Description
RxData	Request data details
TxData	Response data details
DumpData	Hex dump request/response data
Search	Folder searches
Info	File information requests
File	File open/close
FileIO	File read/write
Error	Errors
Directory	Directory commands

Flag	Description
Timing	Time packet processing
Session	Session creation/deletion

- The `log4j.properties` file must also have NFS protocol debug output enabled using:
`log4j.logger.org.alfresco.nfs.server=debug`
- The following logging level must also be enabled to log debug output from the core file server code:
`log4j.logger.org.alfresco.fileserver=debug`
- There are also the following log4j output options for the NFS/mount/portmapper services:
`log4j.logger.org.alfresco.nfs.protocol=debug`
- Output server level debug information from the NFS, mount and portmapper services.
`log4j.logger.org.alfresco.nfs.protocol.auth=debug`

Configuring IMAP protocol support

IMAP protocol support allows email applications that support IMAP (including Outlook, Apple Mail, Thunderbird, and so on) to connect to and interact with Alfresco repositories.

Each user has their own set of mailboxes stored within Alfresco, for example, they have their own INBOX. Users can manage emails within Alfresco ECM, and the workflow, transformation, and permissions features are available.

In addition, Share sites can be nominated as IMAP Favorites. This means that the site contents show as a set of IMAP folders. Non-favorite sites are not shown.

A metadata extractor for IMAP emails (RFC822 messages) can extract values from the contents of the email message and store the values as Alfresco properties.

Enabling the IMAP protocol

The IMAP protocol server is disabled by default. You need to enable the IMAP protocol server to start interaction between the email client and the Alfresco repository.

- Open the `alfresco-global.properties` file.
- Enable the IMAP server by setting the following property to `true`:
`imap.server.enabled=true`
- Set the IMAP server to listen on a specific interface using the following property:
`imap.server.host=x.x.x.x`

Where `x.x.x.x` is the IP address (or corresponding DNS address) of your external IP interface. Do not use `127.0.0.1` or `localhost`.

By default, the IMAP server listens on all interfaces on the default IMAP port of 143. You can set this property to use an alternative port number, for example 144.

- Restart the Alfresco server.

Once the Alfresco server has restarted, the new configuration will take effect. Since the IMAP server has only one instance, make your configuration changes to the `<extension root>alfresco-global.properties` file. You can also make your changes to `<extension root>\alfresco\extension\subsystems\imap\default\default` for the IMAP subsystem configuration to take precedence.

IMAP subsystem properties

The following properties can be configured for the IMAP subsystem.

Enabling the IMAP protocol

The following properties control the IMAP subsystem:

imap.server.enabled

Enables or disables the IMAP subsystem.

imap.server.port=143

IMAP has a reserved port number of 143. You can change it using this property.

imap.server.host=<your host name>

Replace this value with the IP address (or corresponding DNS address) of your external IP interface.

Configure the following properties of the sysAdmin subsystem:

alfresco.protocol

The protocol component of the alfresco web application URL, for example, `http`.

alfresco.host

The host name of the Alfresco URL, which is externally resolved. Use `${localname}` for the locally-configured host name.

alfresco.port

The port number of the Alfresco URL, which is externally resolved. For example, `8080`

alfresco.context

The context path component of the Alfresco URL. Typically this is `alfresco`.

To configure the IMAP Home space, which is used to store user mailboxes in ARCHIVE mode, in particular the user's INBOX, use the following properties:

imap.config.home.store=\${spaces.store}

Specifies the default location for the IMAP mount point. For example, `${spaces.store}`.

imap.config.home.rootPath=/\${spaces.company_home.childname}

Specifies the default location for the IMAP mount point. For example, `/${spaces.company_home.childname}`.

imap.config.home.folderPath=cm:Imap Home

Specifies the QName of the default location for the IMAP mount point. For example, `cm:Imap Home`.

Enabling IMAPS

IMAPS is a secure IMAP that is encrypted using SSL. IMAPS is assigned to port number 993 by default. When you have enabled the IMAP subsystem, you must configure the default Java keystore, and then enable IMAPS.

To configure the default Java keystore, use the following properties:

javax.net.ssl.keyStore=mySrvKeystore

Specifies the keystore to be used

javax.net.ssl.keyStorePassword=123456

Specifies the keystore password

To enable IMAPS, use the following properties:

imap.server.imaps.enabled=true

Specifies that IMAPS is enabled

imap.server.imaps.port=993

Specifies the IMAPS port number

Extracting attachments

An IMAP message can contain a message and a set of attachments, and the IMAP server can split the attachments into separate content nodes. Use this property with caution if you have a large repository. See [Troubleshooting IMAP](#) for more information.

imap.server.attachments.extraction.enabled

Defines whether or not attachments are extracted.

IMAP mount points

IMAP mount points are used to control which folders are available using IMAP and the mode in which they are accessed. Modes are used to define the type of interaction available.

The IMAP integration offers the following access modes:

Archive

Allows emails to be written to and read from Alfresco by the IMAP client by drag and drop, copy/paste, and so on, from the email client.

Virtual

Documents managed by Alfresco can be viewed as emails from the IMAP client. Documents are shown as virtual emails with the ability to view metadata and trigger actions on the document, using links included in the email body.

Mixed

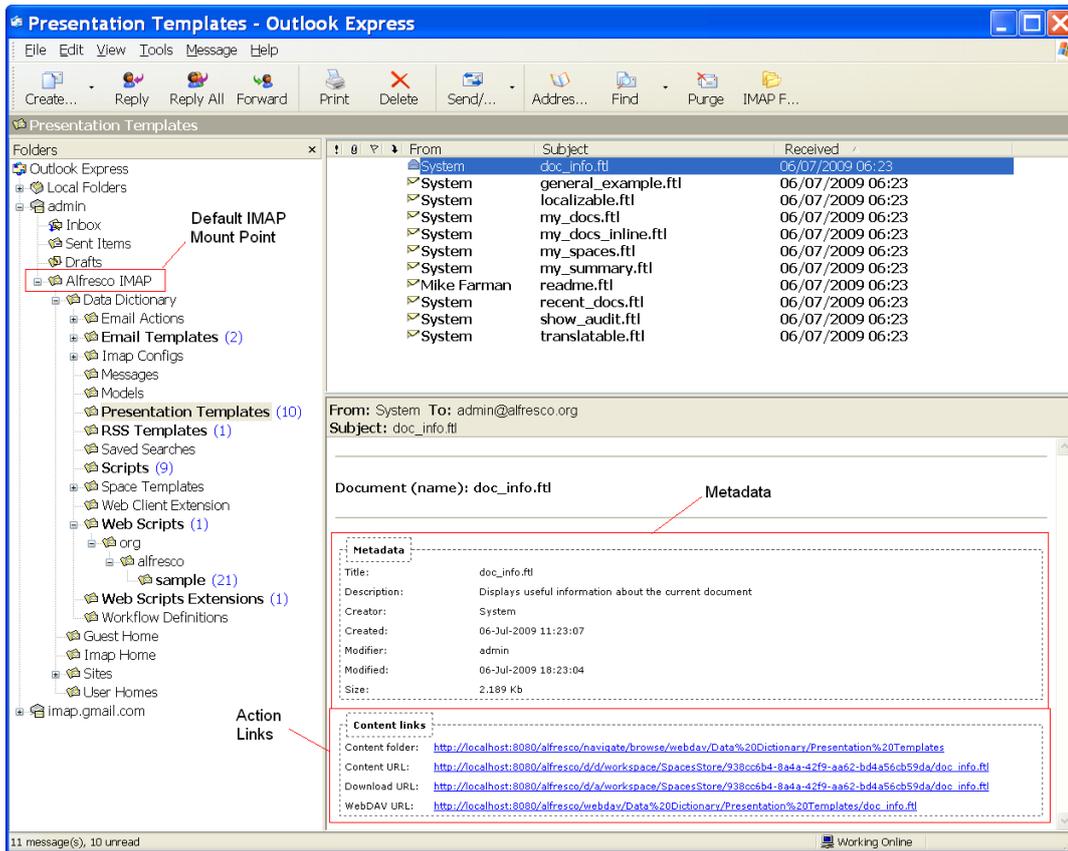
A combination of both archive and virtual modes, that is, both document access and email management are available.

By default, a single mount point called **AlfrescoIMAP** is defined for **Company Home** and you can change it or add more mount points.

Virtual view email format

The virtualized view uses presentation templates to generate the mail body and display document metadata, action links (for download, view, webdav, folder) and Start Workflow form (HTML view only).

The templates are stored in the repository in **Company Home > Data Dictionary > Imap Configs > Templates**. Separate templates are available to generate either a HTML or plain text body, based on the format request by the email client. The templates can be customized to change the metadata and actions available in the email body.



Marking sites as IMAP favorites

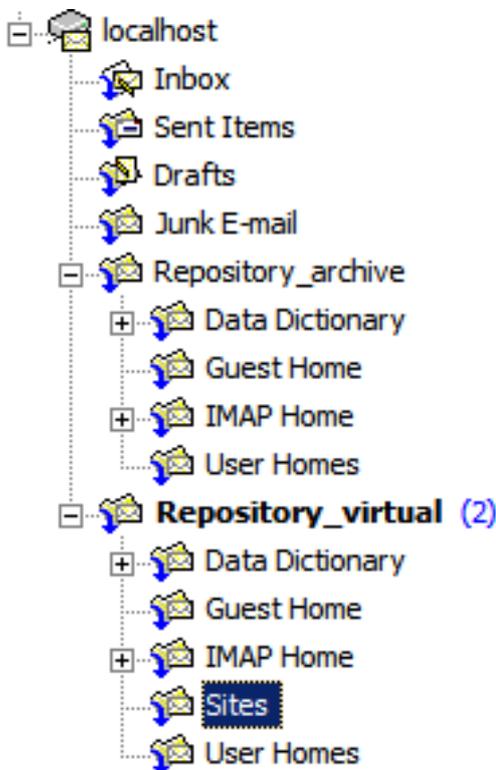
To have access to Alfresco Share sites using IMAP, the site(s) need to be added to your list of sites using Share IMAP Favorites.

1. Select **IMAP Favorites** in the Share **My Sites** dashlet on your **My Dashboard** page:





2. Refresh your IMAP view to see the new sites.



You can see the site added to the IMAP Sites folder.

Configuring system properties

The sysAdmin subsystem allows real time control across some of the general repository properties. The sysAdmin subsystem replaces the `RepoServerMgmt` management bean.

sysAdmin subsystem properties

The following properties can be configured for the sysAdmin subsystem.

server.maxusers

The maximum number of users who are allowed to log in or -1 if there is no limit.

server.allowedusers

A comma-separated list of users who are allowed to log in. Leave empty if all users are allowed to log in.

server.transaction.allow-writes

A Boolean property that when true indicates that the repository will allow write operations (provided that the license is valid). When false the repository is in read-only mode.

The following properties specify the parameters that control how Alfresco generates URLs to the repository and Share. These parameters might need to be edited from their default values to allow the URLs to be resolved by an external computer.

alfresco.context

Specifies the context path of the Alfresco repository web application. The default is `alfresco`.

alfresco.host

Specifies the externally resolvable host name of the UR Alfresco web application. The default value is `${localname}`. If this is used for the value of this property, the token `${localname}` will be automatically replaced by the domain name of the repository server.

alfresco.port

Specifies the externally resolvable port number of the Alfresco web application URL. The default is `8080`.

alfresco.protocol

Specifies the protocol component of the Alfresco web application. The default is `http`.

share.context

Specifies context path component of the Share web application URL. The default is `share`.

share.host

Specifies the externally resolvable host name of the Share web application URL. The default value is `${localname}`.

share.port

Specifies the externally resolvable port number of the Share web application URL. The default is `8080`.

share.protocol

Specifies the protocol to use. The default is `http`.

Encrypting configuration properties

The `alfresco-global.properties` file holds configuration properties that contain sensitive information or passwords, such as `db.password`. This section provides information on the properties that are encryptable and the process to encrypt them using the Alfresco Encrypted Properties Management Tool.

 Boolean properties, number properties, and properties that contain expressions cannot be encrypted. Alfresco One 5.0 provides support for encrypting the following configuration properties:

- `dir.root`
- `db.driver`
- `db.username`
- `db.password`
- `db.name`
- `db.pool.validate.query`

- ooo.exe
- swf.exe
- swf.languagedir
- jodconverter.officeHome
- alfresco_user_store.adminpassword
- dir.license.external
- index.subsystem.name
- cryptodoc.jce.keystore.path
- cryptodoc.jce.keystore.password
- cryptodoc.jce.key.aliases
- cryptodoc.jce.key.passwords

Encrypting configuration properties: overall process

This topic describes how to encrypt sensitive properties in the `alfresco-global.properties` configuration file.

1. Run the Alfresco Encrypted Properties Management Tool.
 - a. Navigate to `<ALFRESCO_HOME>/bin` directory.
 - b. Locate the Alfresco Encrypted Properties Management Tool, `alfresco-spring-encryptor.jar`.
 - c. Use the Module Management Tool (MMT) to run the executable jar file.

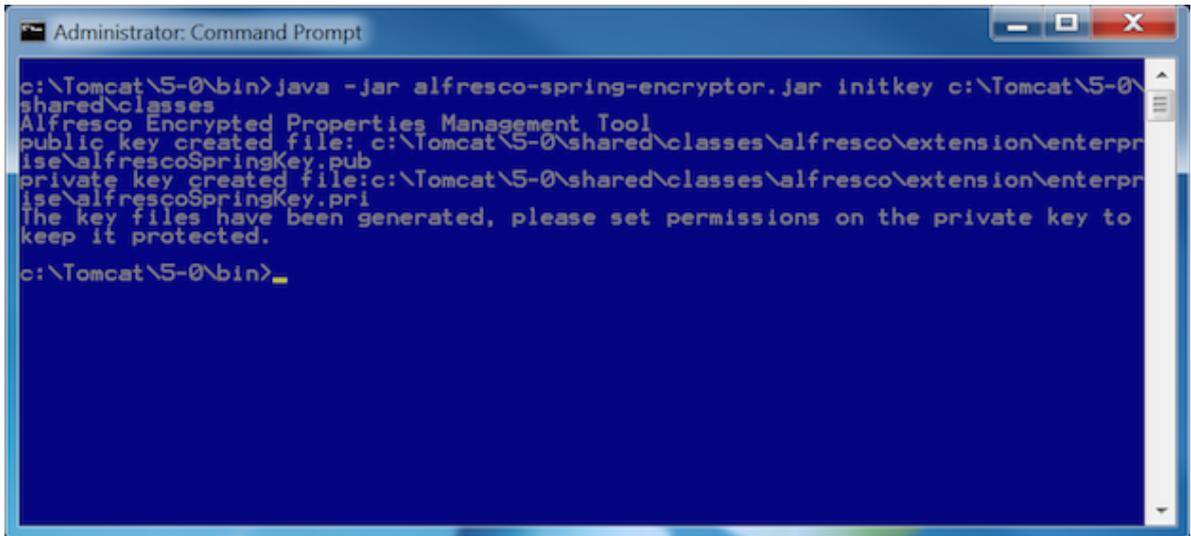
```
java -jar alfresco-spring-encryptor.jar
```

```
Administrator: Command Prompt
c:\Tomcat\5-0\bin>java -jar alfresco-spring-encryptor.jar
Alfresco Encrypted Properties Management Tool
USAGE : org.alfresco.enterprise.spring.encrypted.PublicPrivateKeyStringEncryptor
initkey | encrypt | validate <shared dir> [args]

<shared dir> is where you put your alfresco-global.properties file e.g. c:/to
mcat/shared/classes
initkey : initialise the public and private keystores
encrypt : encrypt a value
validate : compare an encrypted value with a value to see if they match
c:\Tomcat\5-0\bin>
```

2. Generate the public and private keys using the `initkey` function. The public and private key pair is stored in the `enterprise` directory.

```
java -jar alfresco-spring-encryptor.jar initkey c:/alfresco/tomcat/
shared/classes
```



```
Administrator: Command Prompt
c:\Tomcat\5-0\bin>java -jar alfresco-spring-encryptor.jar initkey c:\Tomcat\5-0\
shared\classes
Alfresco Encrypted Properties Management Tool
public key created file: c:\Tomcat\5-0\shared\classes\alfresco\extension\enterpr
ise\alfrescoSpringKey.pub
private key created file:c:\Tomcat\5-0\shared\classes\alfresco\extension\enterpr
ise\alfrescoSpringKey.pri
The key files have been generated, please set permissions on the private key to
keep it protected.
c:\Tomcat\5-0\bin>
```

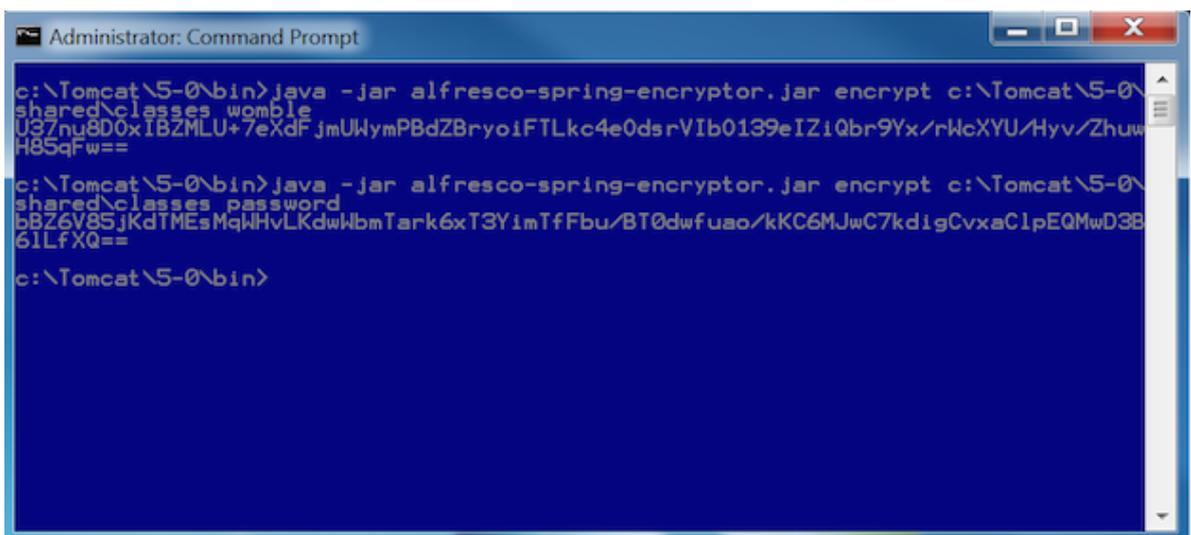
You now have a public key (alfrescoSpringKey.pub) and a private key (alfrescoSpringKey.pri) in your <ALFRESCO_HOME>/tomcat/shared/classes/alfresco/extension/enterprise directory.

-  The private key file should be secured with the operating system permissions so that only the Alfresco process can read it.
-  Anyone can encrypt new values with the public key but only the Alfresco process can read the plain text value with the private key.

3. Generate the encrypted string for your password/value using the encrypt function.

```
java -jar alfresco-spring-encryptor.jar encrypt c:/alfresco/tomcat/
shared/classes <password>
```

-  In the above command, remember to replace <password> with the actual password that you want to encrypt.

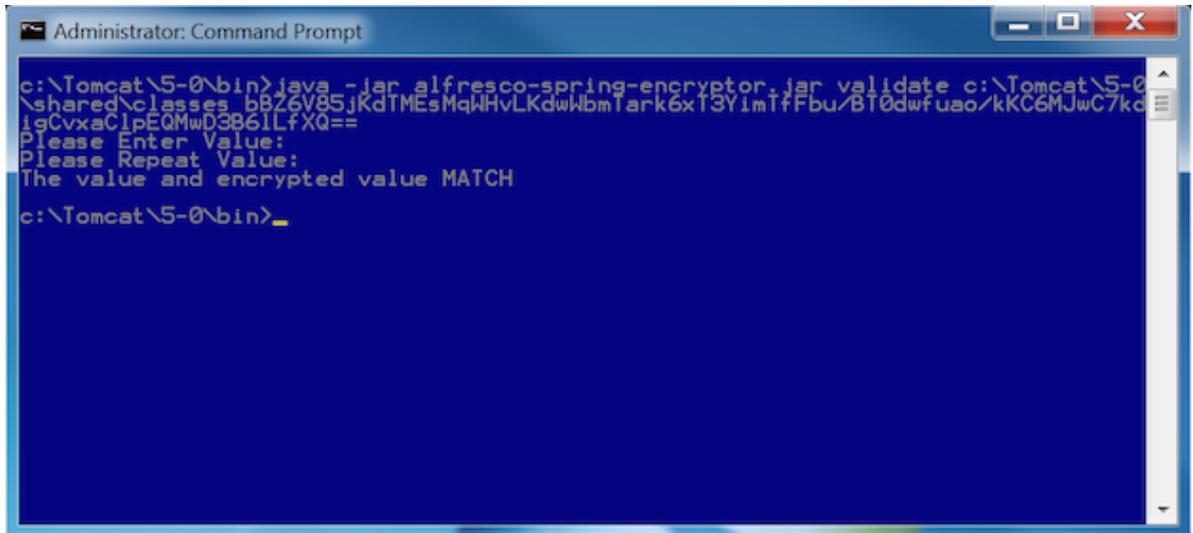


```
Administrator: Command Prompt
c:\Tomcat\5-0\bin>java -jar alfresco-spring-encryptor.jar encrypt c:\Tomcat\5-0\
shared\classes womble
U37nu8D0xIBZMLU+7eXdF jmUHymPBdZBryoiFTLkc4e0dsrVIb0139eIZiQbr9Yx/rWcXYU/Hyv/Zhuw
H85qFw==
c:\Tomcat\5-0\bin>java -jar alfresco-spring-encryptor.jar encrypt c:\Tomcat\5-0\
shared\classes password
bBZ6V85JKdTMESMqHHvLKdWbTark6xT3YimTfFbu/BT0dfuao/kKC6MJwC7kdigCvxaC1pEQMwD3B
61LfXQ==
c:\Tomcat\5-0\bin>
```

4. Validate that the encrypted value obtained in Step 3 will decrypt the password.
 - a. Run the validate function.

```
java -jar alfresco-spring-encryptor.jar validate c:/alfresco/tomcat/
shared/classes <encrypted value>
```

-  In the above command, remember to replace `<encrypted value>` with encrypted string value obtained in Step 3.
- b. You will be prompted to specify the value. Enter the password/value you want to encrypt.
- c. You will be prompted to specify the value again. Enter the password/value you want to encrypt.



```
Administrator: Command Prompt
c:\Tomcat\5-0\bin>java -jar alfresco-spring-encryptor.jar validate c:\Tomcat\5-0\
\shared\classes_bBZ6V85jKdTMEsMqWHvLKdwWbmTark6xT3YImTfFbu/BT0dwfuao/kKC6MJwC7kd
igCvxaCipEQMwD3861LfXQ==
Please Enter Value:
Please Repeat Value:
The value and encrypted value MATCH
c:\Tomcat\5-0\bin>
```

5. Add the encrypted password to `<ALFRESCO_HOME>/tomcat/shared/classes/alfresco-encrypted.properties` file.

```
db.password.enc=ENC(<enter encrypted password here>)
```

```
db.password.enc=ENC(QcAf1Lr81meuP2p6Lu9ZQqFY1AsCfoWd)
```

 Uncomment the `db.password.enc` property by removing the `"#"` character.

6. Set the value of the `db.password` property in the `alfresco-global.properties` file to point to the `db.password.enc` property in the `alfresco-encrypted.properties` file.

```
db.password=${db.password.enc}
```

 Uncomment the `db.password` property by removing the `"#"` character.

Configuring the repository

Deploying Alfresco with a different context path

There are a number of updates that you need to make if you want to deploy Alfresco to a context path that is not `/alfresco`.

The context path is the path that is used by applications (for example, Share, SOLR, SharePoint, and others) to access the Alfresco repository. If you change this value, you must reflect the change in your application server configuration.

 You cannot install Alfresco at the server root (`/`). In other words, the context path cannot be the server root.

Follow these steps if you want to deploy Alfresco to a context path that is not `/alfresco`. The string `new-context-path` is used to represent the name of the context path that you are using:

1. Deploy the `alfresco.war` file to a different context path; for example, if you are using Tomcat, rename the `alfresco.war` file to `new-context-path.war` and then deploy it. For other application servers, set the context path in the Admin Console during deployment.
2. Update `alfresco-global.properties` with the name of the context path:
`alfresco.context=new-context-path.`
3. Update `share-config-custom.xml` as described in [Configuring the Share default port](#).
4. Update the context path setting in the `_vti_bin` application:

- a. Unpack the `_vti_bin.war` file.
- b. Locate the `WEB-INF/web.xml` file in the `_vti_bin` application.
- c. Replace the `<param-value>` value with `/new-context-path/aos` to update the context parameter with the new context path. The example shows the default values in the `WEB-INF/web.xml` file:

```
<context-param>
  <param-
name>org.alfresco.enterprise.repo.officeservices.dispatch.SERVICES</
param-name>
  <param-value>/alfresco/aos</param-value>
  <description>A space separated list of url-encoded context paths of
SharePoint protocol enabled applications (e.g. Alfresco One, Alfresco
Office Workdesk)</description>
</context-param>
```

- d. Repack the contents of the `_vti_bin` application into a `_vti_bin.war` file and deploy it.
5. Unpack `ROOT.war` and edit the `index.jsp` file to set the context path:

Change `/alfresco` to `/new-context-path`:

```
if(request.getMethod().equals("PROPFIND") ||
request.getMethod().equals("OPTIONS"))
{ ServletContext alfrescoContext = application.getContext("/
alfresco"); ... }
```

6. Repack the contents of `ROOT.war` and deploy it.
7. Update the Solr 4 or Solr configuration to specify the new context path:

If you are using Solr 4, modify the following files:

```
solr4/workspace-SpacesStore/conf/solrcore.properties
solr4/archive-SpacesStore/conf/solrcore.properties
```

If you are using Solr, modify the following files:

```
solr/workspace-SpacesStore/conf/solrcore.properties
solr/archive-SpacesStore/conf/solrcore.properties
```

to specify the properties relevant to your configuration:

```
alfresco.host=localhost
alfresco.port=8080
alfresco.port.ssl=8443
alfresco.baseUrl=/alfresco
```

Deploying Alfresco with a reverse proxy

Follow this guidance if you want to run Alfresco with a reverse proxy.

If the reverse proxy maps the target server to a different context path, or if you deployed Alfresco specifically to a different context path, you need to follow the steps in [Deploying Alfresco with a different context path](#), with the following changes:

- a. In step 2, update the values in the `alfresco-global.properties` file:

```
alfresco.context=xxx
alfresco.host=xxx
```

```
alfresco.port=xxx
alfresco.protocol=xxx
```

where `xxx` are the externally visible context, host name, port number and protocol values.

- b. You must specify the context path that is externally visible in all steps, and not the context path that the repository is actually running on. Exceptions are in step 1 and in step 3 if Share is connecting to the repository directly and not through the reverse proxy. The other exception is in step 7 if Solr or Solr 4 is contacted directly and not through the reverse proxy.

Tuning the JVM

The hardware requirements for the Alfresco repository and Share are variable and depend on the number of concurrent users that access the system. You can tune the memory and garbage collection parameters for the JVM to be appropriate for your situation. This section suggests metrics and estimates, but your system can vary.



In the following sections, the terms concurrent users and casual users are used. Concurrent users are users who are constantly accessing the system through Alfresco with only a small pause between requests (3-10 seconds maximum) with continuous access 24/7. Casual users are users occasionally accessing the system through the Alfresco or WebDAV/CIFS interfaces with a large gap between requests (for example, occasional document access during the working day).

Hardware

Alfresco degrades gracefully on low-powered hardware, and small installations can run well on any modern server. However, for optimum performance, we recommend the following:

- Use 64 bit systems only.
- Use a system with a clock speed above 2.0 GHz.
- Reserve enough RAM for your operating system beyond the memory required for your JVM.
- Keep search indexes on your local disk instead of on network storage.

Disk space usage

The size of your Alfresco repository defines how much disk space you will need; it is a very simple calculation. Content in Alfresco is, by default, stored directly on the disk. Therefore, to hold 1000 documents of 1 MB will require 1000 MB of disk space. You should also make sure there is sufficient space overhead for temporary files and versions. Each version of a file (whether in DM or WCM) is stored on disk as a separate copy of that file, so make allowances for that in your disk size calculations (for DM, use versioning judiciously).

Use a server class machine with SCSI Raid disk array. The performance of reading/writing content is almost solely dependent on the speed of your network and the speed of your disk array. The overhead of the Alfresco server itself for reading content is very low as content is streamed directly from the disks to the output stream. The overhead of writing content is also low but if Solr is installed on the same machine, additional overhead should be allowed for the indexing process. For more information, see [Calculate the memory needed for Solr nodes](#).

Virtualization

Alfresco runs well when virtualized, but you should expect a reduction in performance. When using the rough sizing requirements given, it might be necessary to allocate twice as many resources for a given number of users when those resources are virtual. Para-virtualization, or virtualized accesses to native host volumes do not require as many resources. Benchmarking your environment is necessary to get a precise understanding of what resources are required.

JVM memory and CPU hardware for multiple users

The repository L2 Cache, plus initial VM overhead, plus basic Alfresco system memory, is setup with a default installation to require a maximum of approximately 1024 MB.

This means that you can run the Alfresco repository and web client with many users accessing the system with a basic single CPU server and only 1024 MB of memory assigned to the Alfresco JVM. However, you must add additional memory as your user base grows, and add CPUs depending on the complexity of the tasks you expect your users to perform, and how many concurrent users are accessing the client.

 Note that for these metrics, **N** concurrent users is considered equivalent to **10xN** casual users that the server could support.

Number of users	Recommended memory / CPU settings per server
For 50 concurrent or up to 500 casual users	2.0 GB JVM RAM 2x server CPU (or 1xDual-core)
For 100 concurrent users or up to 1000 casual users	2.0 GB JVM RAM 4x server CPU (or 2xDual-core)
For 200 concurrent users or up to 2000 casual users	2.5 GB JVM RAM 8x server CPU (or 4xDual-core)

JVM settings

There are a number of typical JVM settings that you can use in your repository configuration.

The standard JVM settings are as follows:

```
-XX:MaxPermSize=256M
-Xss1024K
-Xms1G
-Xmx2G
-Dcom.sun.management.jmxremote
```

Tune the JVM using the following three steps:

1. Use as much RAM as possible for the JVM (`-Xmx32GB`).
2. Set the Permanent Generation to 256 MB (`-XX:MaxPermSize:256m`).
3. Do not add any other configuration settings.

To avoid memory swapping, `-Xmx` should never exceed the available RAM in the system. Remember to leave room for memory used by the operating system and other applications, like OpenOffice using JOD (JOD often uses 1 GB of RAM per OO instance).

In general, if you do not give the JVM enough heap, adjusting the other JVM settings will not make any difference. Once the JVM has enough heap, you should not need to change the other JVM settings. The 1.6 JVM is generally excellent at memory optimization and is capable of functioning without adjustment.

The remaining information on this page might help in exceptional circumstances only. It is unlikely to apply to your use case, and we advise against JVM tuning beyond what has already been discussed in this section.

Permanent Generation (PermGen) Size

The default PermGen size in Sun JVMs is 64 MB, which is very close to the total size of permanent objects (Spring beans, caches, and so on) that Alfresco creates. For this reason it is quite easy to overflow the PermGen using configuration changes or with the addition of custom

extensions, and so it is recommended that you increase the PermGen to avoid OutOfMemory errors. For example, `-XX:MaxPermSize=160M` is a good starting point.

 The size of the PermGen is now increased in the Alfresco startup scripts, so provided you are using those scripts, no changes should be necessary.

Maximum JVM heap size 32/64 bit

An important calculation to keep in mind is:

```
(Managed Heap + native heap + (thread stack size * number of threads)) cannot exceed 2 GB on 32bit x86 Windows or Linux systems
```

This is a limitation of the Sun Java VM. It means that even if you install 4 GB of RAM into your server, a single instance of the JVM cannot grow beyond 2 GB on a 32 bit server machine.

 A 64 bit OS/JVM has much bigger values. It is recommended that a 64 bit OS with large memory hardware (>2 GB assigned to the JVM) is used for deployments of >250 concurrent or >2500 casual users.

You can also set up your machine to cluster if you prefer to solve multi-user access performance issues with additional machines rather than a single powerful server.

Example

The following settings are used on a high-volume clustered 64 bit, dual 2.6 GHz Xeon / dual-core per CPU, 8 GB RAM environment. Note the different memory ratios and try to preserve them on different environments. A minimum MaxPermSize of 128 MB is recommended but might sometimes require 256 MB.

```
-Xmx3G -XX:MaxPermSize=256M
```

Low end machines

The stack size of 1024 KB (`-Xss1024K`) is generous. Some installations might require a little over 512 KB. Many use only 256 KB. If the per-thread memory consumption is too high for your installation, reduce the stack size to 512 KB and then to 256 KB and note any memory-related errors in the logs.

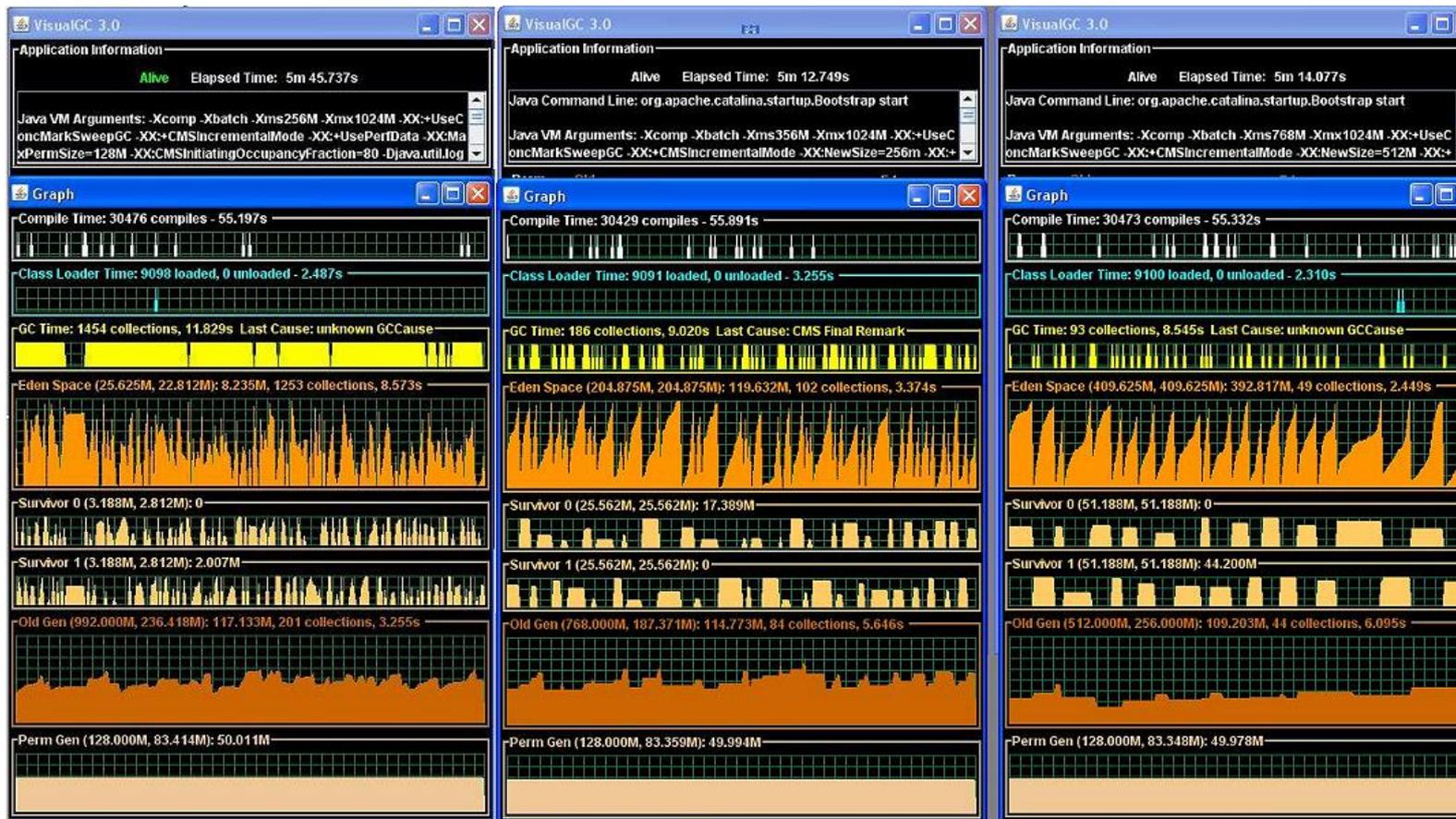
The `NewSize` should be kept as large as possible. It can be reduced, but the memory consumption should be watched on a monitoring tool, for example, JConsole, to ensure that the rate of spillover of temporary objects is kept down. If the machine is supporting 500 simultaneous operations, for instance, then the spillover of temporary objects (from `NewSize` being too small) will cause hold-ups on memory assignment as the garbage collector does sweeps.

Effects of NewSize

Given that the OldGen is composed primarily of cache data of up to about 520 MB, at least 1 GB should be reserved for OldGen. Once `-Xmx` increases, the OldGen can be increased to 2 GB. 512 MB should be left as a buffer to account for miscellaneous (PermGen, and so on). So the following variations might be applied:

```
-Xmx2G -Xms1G -XX:NewSize=512M (OldGen at least 1 GB)
-Xmx3G -Xms1G -XX:NewSize=512M (OldGen at least 2 GB)
-Xmx4G -Xms2G -XX:NewSize=1G (OldGen at least 2.5 GB)
-Xmx6G -Xms3G -XX:NewSize=2G (OldGen at least 3.5 GB)
-Xmx8G -Xms4G -XX:NewSize=3G (OldGen at least 4.5 GB)
```

If you need these levels, you will need to run JConsole (and Java 6) to observe the rate of spillover from Eden space to Survivor to OldGen. If, after the system has been running for a while, the OldGen size stabilizes, then the `NewSize` can be increased appropriately. The following diagram (using VisualGC) shows how varying the `NewSize` value affects overall garbage collection activity:



Command line configuration

Setting properties on the JVM

- (Windows) At a command prompt, enter the following:

```
Set JAVA_OPTS=-Ddir.root=e:/alfresco/data
```
- (Linux) At a command prompt, enter the following:

```
export JAVA_OPTS=-Ddir.root=/srv/alfresco/data
```

Mixing global properties and system property settings

1. Activate the properties in the `<classpathRoot>/alfresco-global.properties` file.
2. Set all common defaults for your system.
3. On each installation, add the final configuration values. For example:

```
-Ddb.username=alfresco
-Ddb.password=alfresco
-Dindex.tracking.cronExpression='0/5 * * * * ?'
-Dindex.recovery.mode=AUTO
-Dalfresco.cluster.name=ALFRESCO_DEV
```

Configuring Alfresco to work with a web proxy

This topic describes the standard JVM system properties that you can use to set proxies for various protocol handlers, such as HTTP and HTTPS. These properties are used by Surf and all other parts of the system that make `http` call-outs.

All proxies are defined by a host name and a port number. The port number is optional and if not specified, a standard default port will be used.

The following two properties can be set to specify the proxy that will be used by the HTTP protocol handler:

System Properties	Description
<code>http.proxyHost</code>	Specifies the host name or IP address for the proxy server.
<code>http.proxyPort</code>	Specifies the port number for the proxy server. The default port number is 80.

The following two properties can be set to specify the proxy that will be used by the HTTPS protocol handler:

System Properties	Description
<code>https.proxyHost</code>	Specifies the host name or IP address for the proxy server when using https (http over SSL).
<code>https.proxyPort</code>	Specifies the port number for the proxy server when using https (http over SSL). The default port number is 443.

For example, the following command directs all http connections to go through the proxy server with the IP address 172.21.1.130, and the port number 8080:

```
java -Dhttp.proxyHost=172.21.1.130 -Dhttp.proxyPort=8080
```

In addition, you can also set the following non-standard properties for authenticated proxies:

Non-standard Properties	Description
<code>http.proxyUser</code>	Specifies the user name to use with an authenticated proxy used by the HTTP protocol handler. It should be left unset if the proxy does not require authentication.
<code>http.proxyPassword</code>	Specifies the password to use with an authenticated proxy used by the HTTP protocol handler. It should be left unset if the proxy does not require authentication.
<code>https.proxyUser</code>	Specifies the user name to use with an authenticated proxy used by the HTTPS protocol handler. It should be left unset if the proxy does not require authentication.
<code>https.proxyPassword</code>	Specifies the password to use with an authenticated proxy used by the HTTPS protocol handler. It should be left unset if the proxy does not require authentication.

Controlling JVM system properties

This topic describes how to control JVM system properties.

In a standard Linux/Unix installation, system properties can be specified in `-Dname=value` format (separated by spaces) in the `JAVA_OPTS` variable set by the script:

```
tomcat/scripts/ctl.sh
```

In a standard Windows installation, system properties can be listed in `-Dname=value` format (separated by semicolons) before `;-Dalfresco.home` in:

```
tomcat/bin/service.bat
```

Once edited, the commands:

```
tomcat/scripts/serviceinstall.bat REMOVE
```

```
tomcat/scripts/serviceinstall.bat INSTALL
```

must be run to re-register the Alfresco service with the new options.

Secure Sockets Layer (SSL) and the Alfresco repository

There are a number of ways to handle SSL communication when connecting to the Alfresco repository, and some information that you should know about automatic configuration in Alfresco.

When you install Alfresco, port 8443 is automatically configured for SSL communication between Solr and the Alfresco repository. This means that Alfresco, by default, is set to use client certificates for any authentication (the connector on port 8443 is configured with `clientAuth="want"`).

This causes complications when there is communication between a browser protocol and the repository, because Tomcat requests a client certificate for that communication too; for example, when you are using Alfresco Office Services to connect between a Microsoft application and the Alfresco repository. For more information about Alfresco Office Services, see [Installing and configuring Alfresco Office Services](#).

You can still connect to the repository without a client certificate, however if a certificate is present (for example, if you have installed certificates in your Windows certificate store), then the certificate must be signed by the same Certificate Authority that is used for authentication between the repository and Solr. If you select one of the Windows installed certificates, you will not be able to progress, because the certificate is not one that is expected for the Solr to repository communication. In this situation, you need to cancel the certificate window and then you can proceed. If you have no client certificates, you can use port 8443 without issues.

These topics discuss how to set up SSL for non-Solr communication with the Alfresco repository, and the method that you use to configure SSL varies depending on whether you are configuring your production or test environments. For example, if you are setting up a production environment, we would recommend that you use a proxy server to handle SSL communication. If you are configuring a test environment, you might want to edit your configuration files directly (and listen for SSL on a port that is not port 8443; for example, port 443).

If you are interested in setting up SSL and security for Solr, this is discussed in detail in [Solr 4 security](#) on page 198.

Configuring SSL for a production environment

Several proxy application servers are available to configure for SSL communications; for example, HAProxy, Apache HTTP Server or Nginx. Using a proxy server means that you do not have to edit your Alfresco configuration files directly. We discuss how to configure Alfresco with HAProxy.

1. If you are using HAProxy 1.5, detailed information is here: [HAProxy Configuration Manual](#). You can also see the specific configuration for the Alfresco Cloud service here: [HAProxy for Alfresco](#).

You define information in a number of sections including:

- *global*: global settings (including `pidfile` which writes PIDs of all daemons into a file)
- *defaults*: default settings (including `mode` which sets the working mode to `http` instead of `tcp` and `log` which sets the log context)
- *option*: how HAProxy will operate; for example, logging, session redistribution and performance settings
- *timeout*: timeouts appropriate to your environment
- *compression*: to reduce amount of data being sent using gzip compression

- *frontend*: to specify listening ports that accept client connections (one for `http` and one for `https`)
- *acl*: ACLs to map incoming requests to the appropriate service
- *rspadd*: setup of an HSTS (HTTP Strict Transport Security) header
- *http-request*: HTTP request deny settings to limit access to certain services (optional)
- *redirect*: redirection service, in case the URL path is different to the path used to access the service
- *backend*: to define servers that receive the incoming connections from the proxy (matched against the defined ACL list)

Here is an example configuration, where Share and the Alfresco repository are running in the same JVM:

```
global
log 127.0.0.1 local2 info
pidfile /var/run/haproxy.pid

defaults
mode http
log global

option httplog
option dontlognull
option forwardfor
option http-server-close
option redispatch
#optimisations
option tcp-smart-accept
option tcp-smart-connect

timeout http-request 10s
timeout queue 1m
timeout connect 5s
timeout client 2m
timeout server 2m
timeout http-keep-alive 10s
timeout check 5s
retries 3

compression algo gzip
compression type text/html text/html;charset=utf-8 text/plain text/
css text/javascript application/x-javascript application/javascript
application/ecmascript application/rss+xml application/atomsvc+xml
application/atom+xml application/atom+xml;type=entry application/atom
+xml;type=feed application/cmismquery+xml application/cmismallowableactions
+xml application/cmismatom+xml application/cmismtree+xml application/
cmismacl+xml application/msword application/vnd.ms-excel application/
vnd.ms-powerpoint application/json

#####
# Front end for http to https redirect
frontend http
bind *:80
redirect location https://<domain.name>/

#####
#HTTPS Frontend
frontend https
bind *:443 ssl crt </etc/haproxy/yourcert.pem>
capture request header X-Forwarded-For len 64
capture request header User-agent len 256
```

```

capture request header Cookie len 64
capture request header Accept-Language len 64

# ACL for backend mapping based on host header
acl is_my_hdr_beg(host) -i <domain.name>

# ACL for backend mapping based on url paths
acl solr_share_path path_reg ^/share./*/proxy/alfresco/api/solr/.*
acl solr_alf_path path_reg ^/alfresco/service/api/solr/.*
acl share_redirect path_reg ^$|^/$

# Changes to header responses
rspadd Strict-Transport-Security:\ max-age=15768000

# http-request denied paths
http-request deny if solr_share_path
http-request deny if solr_alf_path

# Redirects
redirect location /share/ if share_redirect is_my

# List of backends
use_backend alfresco if is_my

default_backend alfresco

backend alfresco
option httpchk GET /share
#balance url_param JSESSIONID check_post
balance leastconn
cookie JSESSIONID prefix
server alfresco1 <10:0.0.1:8080> cookie alfresco1 check inter 5000

```

2. If you are using Apache HTTP Server, you can use the module `mod_proxy` to configure a web server to act as a proxy server. Configure Apache to include the `mod_proxy` to forward requests for a web application to a Tomcat instance.

The Apache 1.3 Proxy Support section in [Apache Tomcat 7 Proxy Support](#) gives detailed instructions on how to set up the proxy server.

When you set up the `<Connector>` element in Tomcat, use `port=8443` and `proxyPort=443`. See [Configuring SSL for a test environment](#) on page 111 for guidance if you are not able to use these ports in your configuration.

Configuring SSL for a test environment



These instructions should only be used for configuring a test environment. The recommended approach when configuring your production environment is to use a proxy server to handle all SSL communication. See [Configuring SSL for a production environment](#) on page 109 for more information.

Several proxy application servers are available to configure for SSL communications; for example, Apache Tomcat, HAProxy or Nginx. We explain how to configure SSL using Tomcat on Linux.

1. Navigate to `<TOMCAT_HOME>/conf/server.xml` and add a new Connector configuration. The default `Connector port` is set to 8443, and `SSLEnabled` is set to true. Port 8443 is configured on installation as an SSL port, but should only be used to communicate with Solr. Use one of the following three options for your new Connector configuration and replace `keystoreFile="/path/to/ssl.keystore"` and `keystorePass="password"` with appropriate values:

- a. Start Tomcat on an alternative port (for example, port 7070), and create a redirect rule from the default HTTPS port 443 to your chosen port, as shown in the example:

```
<Connector port="7070" proxyPort="443" URIEncoding="UTF-8"
protocol="org.apache.coyote.http11.Http11Protocol"
SSLEnabled="true" maxThreads="150" scheme="https"
keystoreFile="/path/to/ssl.keystore"
keystorePass="password"
keystoreType="JCEKS"
secure="true" connectionTimeout="240000"
clientAuth="false"
sslProtocol="TLS"
allowUnsafeLegacyRenegotiation="true"
maxHttpHeaderSize="32768" />
```

A non-privileged user cannot start a server on a port below 1024, therefore port 443 is not accessible.

Edit the server's iptables configuration to specify the redirection:

```
# Redirect external packets
-A PREROUTING -j NAT-Port-Redirect

# redirect http traffic
-A NAT-Port-Redirect -p tcp -m tcp --dport 80 -j REDIRECT --to-ports
8080
# redirect https traffic
-A NAT-Port-Redirect -p tcp -m tcp --dport 443 -j REDIRECT --to-ports
7070
```

- b. Alternatively, configure an available port (for example, port 7070) without a proxy port, as shown in the example:

```
<Connector port="7070" URIEncoding="UTF-8"
protocol="org.apache.coyote.http11.Http11Protocol"
SSLEnabled="true" maxThreads="150" scheme="https"
keystoreFile="/path/to/ssl.keystore"
keystorePass="password"
keystoreType="JCEKS"
secure="true" connectionTimeout="240000"
clientAuth="false"
sslProtocol="TLS"
allowUnsafeLegacyRenegotiation="true"
maxHttpHeaderSize="32768" />
```

This is similar to the previous Connector configuration, except that there is no proxy port.

- c. Alternatively, configure SSL on the default port 443 directly, as shown in the example:

```
<Connector port="443" URIEncoding="UTF-8"
protocol="org.apache.coyote.http11.Http11Protocol"
SSLEnabled="true" maxThreads="150" scheme="https"
keystoreFile="/path/to/ssl.keystore"
keystorePass="password"
keystoreType="JCEKS"
secure="true" connectionTimeout="240000"
clientAuth="false"
sslProtocol="TLS"
allowUnsafeLegacyRenegotiation="true"
maxHttpHeaderSize="32768" />
```

This is similar to the earlier Connector configurations, except that the Connector port is set to 443, and there is no proxy port.

2. Edit the `alfresco-global.properties` file with these values:

```
alfresco.protocol=https
alfresco.host=servername
alfresco.port=8443
```

```
alfresco.context=alfresco
```

where `alfresco.host=servername` is your host name, `alfresco.port` is the external facing port of your choice, and `alfresco.context` is the path that you use for your context path (`alfresco` is the default).

Configuring the repository cache

The Alfresco repository provides in-memory caches. These caches are transaction safe and can be clustered. Caches greatly improve repository performance but they use Java heap memory.

Clustering is enabled and initialized by installing a valid clustering license. In this case, the caches are provided and managed by Hazelcast. On the other hand, when clustering is not enabled, caching is provided by Google's `ConcurrentLinkedHashMap` library, which is a high performance version of `java.util.LinkedHashMap` for use as a software cache. For details, see <https://code.google.com/p/concurrentlinkedhashmap/>.

In both clustered and non-clustered cases, caching is configured and used in the same unified way.

Individual cache settings

To configure a cache, specify a series of properties where the property names begin with the cache name as specified in the Spring cache definition. For example, if the cache name is `cache.myCache`, then the properties should all start with `cache.myCache`.

For example:

```
cache.myCache.maxItems=20000
cache.myCache.timeToLiveSeconds=0
```

The following properties are supported by both clustered and non-clustered (for example, `cluster.type=local`) caches:

maxItems

The `maxItems` attribute is the maximum size a cache can reach. Use zero to set to `Integer.MAX_VALUE`.

eviction-policy

When the `eviction-policy` attribute is set to `NONE`, the cache will not have a bounded capacity and the `maxItems` attribute will not apply. Any other value will cause the `maxItems` attribute to be enabled.

Also, use `LRU` (Least Recently Used) or `LFU` (Least Frequently Used) algorithm with clustered caches so that the value is compatible in both modes (required during startup). Note that the actual value (for example, `LRU`) is of no consequence for the non-clustered caches and eviction is performed as for any Google Guava `CacheBuilder` created cache.

timeToLiveSeconds

The `timeToLiveSeconds` attribute specifies that the cache items will expire once this time has passed after creation.

maxIdleSeconds

The `maxIdleSeconds` attribute specifies that the cache items will expire when not accessed for this period.

tx.maxItems

The `overflowToDisk` attribute is not a fully supported property as `TransactionalCache` is a separate entity but where a `TransactionalCache` bean has been defined, use `{cacheName}.tx.maxItems` to specify its capacity.

The following properties are available for fully-distributed caches and are not supported by the other cache types:

cluster.type

The `cluster.type` attribute determines what type of cache is created when clustering is available. The acceptable values are:

- `fully-distributed`: Uses a Hazelcast IMap backed distributed cache. The cache values can be stored on any member of the cluster, hence the term fully-distributed.
- `local`: Always use a non-clustered cache. The cache values will not reflect updates made to the equivalent cache on another cluster member.
- `invalidating`: Uses a local cache, but when an update or a removal is issued to the cache, an invalidation message is broadcast to all members of the cluster and those members will remove the value from their cache. This value is useful where frequent reads are causing performance problems (due to remote reads) or where values are non-serializable.

backup-count:

The `backup-count` attribute controls how many cluster members should hold a backup of the key/value pair.

eviction-percentage

The `eviction-percentage` attribute controls what percentage of cache entries are shed when the capacity is reached.

merge-policy

The `merge-policy` attribute determines how Hazelcast recovers from split brain syndrome, for example, `haz.ADD_NEW_ENTRY`. See [Network Partitioning \(Split-Brain Syndrome\)](#) for more information.

Adding a MIME type

The MIME type default definitions are in the [mimetype-map.xml](#) file.

Override the `mimetypeConfigService` bean using an extension file.

1. Copy the default definition file and place it in a file called `<extension>/mimetype/mimetypes-extension-map.xml`.
2. Modify the inserted MIME type to match your requirements. For example:

```

<alfresco-config area="mimetype-map">
  <config evaluator="string-compare" condition="Mimetype Map">
    <mimetypes>
      <mimetype mimetype="application/xxx" display="My Example Mimetype">
        <extension>ex</extension>
      </mimetype>
    </mimetypes>
  </config>
</alfresco-config>
```

3. Save the file.
4. Restart Alfresco.

The Mime type will be available in the repository and in Share.

Configuring metadata extraction

Metadata extractors offer server-side extraction of values from added or updated content.

1. Download the [content-services-context.xml](#) file.
2. Copy the file to `<extension>` and save it with the name `custom-repository-context.xml`.

This file contains definitions of the default set of extractors.

3. Declare a new extractor in the `<extension>/custom-repository-context.xml` file.

The following example shows a new extractor written in class `com.company.MyExtractor`:

```
<bean id="com.company.MyExtractor" class="com.company.MyExtractor"
parent="baseMetadataExtractor" />
```

4. Save the file and then restart the Alfresco server.

About aspects

Aspects allow you to add functionality to existing content types.

Aspects use properties to enhance the content types. You can attach behaviors and workflows to aspects. The following table lists the aspects available in Alfresco.

Aspects	Description	Changes in Behavior/Share Interface
Classifiable	Enables categories to be assigned to a content item. For example, content items can be categorized under Languages, Region, Software Document Classification, and so on.	Adding Classifiable aspect displays an additional Categories property on the Edit properties page.
Complianceable	This aspect is no longer valid. For compliance-related behavior, please use Alfresco's Record Management module.	
Dublin Core	Enables metadata (such as publisher, contributor, identifier) to be added to a content item.	Adding Dublin Core aspect displays the following additional metadata properties on the Edit Properties page: <ul style="list-style-type: none"> • Publisher • Contributor • Type • Identifier • Source • Coverage • Rights • Subject
Effectivity	This aspect is no longer valid. For compliance-related behavior, please use Alfresco's Record Management module.	
Summarizable	Enables addition of a brief description about the content item.	Adding Summarizable aspect displays additional Summary property on the Edit Properties page.
Versionable	Enables versioning of a content item each time it is edited (checked out and checked back in or updated). In AlfrescoShare, content items are versionable by default.	Adding Versionable aspect displays the version history of a content item in the Version History section.

Aspects	Description	Changes in Behavior/Share Interface
Emailed	Captures email-related information of the content item, if it is received as an email attachment.	Adding Emailed aspect displays additional properties (such as Originator, Addressee, Addresses, Sent Date and Subject) on the Edit Properties page.
Inline Editable	Enables content items to be edited directly within the document library.	Adding Inline Editable aspect displays the Inline Edit link in the Document Actions section.
Taggable	Enables tagging of content items using keywords. In AlfrescoShare, content items are taggable by default.	Adding Taggable aspect displays the tagged keywords in the Tags section. You can also search for content items in the Document Library using the keywords displayed.
Geographic	Enables a content item to be geographically tagged using latitude and longitude information. The location of content item is displayed as a marker on Google Maps. Click on the marker to display the Document Details page for that content item.	Adding Geographic aspect displays additional Latitude and Longitude properties on the Edit Properties page. Also, the View on Google Maps link is displayed in the Document Actions section.
EXIF	Enables capturing and viewing of additional image-related metadata of a content item.  This aspect is automatically applied to an image content item.	Adding EXIF aspect displays additional information (such as Camera Model, Camera Software, Resolution Unit) about the image in the Edit Properties page.
Audio	Enables capturing and viewing of additional audio-related metadata of a content item.  This aspect is automatically applied to an audio content item.	Adding Audio aspect displays additional information (such as Album, Artist, Composer, Track Number) about the audio file in the Edit Properties page.
Index Control	Enables control over how a content item is indexed.	Adding Index Control aspect displays additional Is Indexed and Is Content Indexed in the Edit Properties page.

About versioning

Versioning allows you to track content history. By default, content that is created in the repository is not versionable. When creating content, users must specify `versionable` on a case-by-case basis.

When content is versionable, the version history is started. The first version of the content is the content that exists at the time of versioning. If you want all content to be versionable at the time of creation, you can modify the definition of that content type in the data dictionary. The definition must include the mandatory aspect `versionable`.

By default, all versionable content has auto-version set to `on`. As a result, when content is updated, the version number is updated.

The auto-version capability can be disabled on a content-by-content basis in the user interface. If you want auto-versioning to be `off` for all content, you can modify the definition of that content type in the data dictionary.

Making all content versionable

1. Download the [contentModel.xml](#) file.
2. Copy the file to the `<configRoot>\classes\alfresco` directory.

3. Search for the `<type>`: `<type name="cm:content">`
4. Immediately after the closing `</properties>` tag, insert the following lines:

```
<type name="cm:content">
  <properties>
    ...
  </properties>
  <mandatory-aspects>
    <aspect>cm:versionable</aspect>
  </mandatory-aspects>
</type>
```

5. Save the file.
6. Restart the Alfresco server.

Disabling the auto-versioning feature

1. Open the `alfresco-global.properties` file.
2. Add the following property:

```
version.store.enableAutoVersioning=true
```

When this property is set to false, the `VersionableAspect` will not respond to any events; even if the aspect is present, it will not create versions.

3. Save the global properties file
4. Restart the Alfresco server.

Setting up database replication

Replication allows you to continuously copy a database to a different server.

To enable replication, you set one server (the slave) to take all its updates from the other server (the master). During replication, no *data* is actually copied. It is the *SQL statements* that manipulate the data that is copied.

All statements that change the master database are stored in the master's binary logs. The slave reads these logs and repeats the statements on its own database. The databases will not necessarily be exactly synchronized. Even with identical hardware, if the database is actually in use, the slave will always be behind the master. The amount by which the slave is behind the master depends on factors such as network bandwidth and geographic location. The other server can be on the same computer or on a different computer. The effect of replication is to allow you to have a nearly current standby server.

Using more than one server allows you to share the read load. You can use two slaves. If one of the three servers fails, you can use one server for service while another server can copy to the failed server. The slaves need not be running continuously. When they are restarted, they catch up. With one or more slaves you can stop the slave server to use a traditional backup method on its data files.

Each slave uses as much space as the master (unless you choose not to replicate some tables) and must do as much write work as the master does to keep up with the write rate. Do not be without at least one slave or comparable solution if high reliability matters to you.



Replication is not another form of back up. You must do normal backups as well as replication. If a user mistypes a `DELETE` statement on the master, the deletion is faithfully reproduced on the slave.

Setting up MySQL replication

1. Open a MySQL command prompt on the master server.

- Grant the slave permission to replicate:

```
GRANT REPLICATION SLAVE ON *.* TO <slave_user> IDENTIFIED BY
'<slave_password>'
```

- If the master is not using the `binary update log`, add the following lines to `my.cnf` (Linux) or `my.ini` (Windows) configuration file on the master, and restart the server:

```
[mysqld]
log-bin
server-id=1
```

 By convention, `server-id` for the master is usually `server-id 1`, and any slaves from 2 onwards, although you can change this. If the master is already using the binary update log, either note the offset at the moment of the backup (the next step), or use the `RESET MASTER` statement to clear all binary logs and immediately begin the backup. You might want to make a copy of the binary logs before doing this if you need to use the binary logs to restore from backup.

- Make a backup of the database.

This will be used to start the slave server. You can skip this step if you use the `LOAD DATA FROM MASTER` statement, but first review the following comments about locking the master.

- Add the following to the configuration file on the slave:

```
master-host=master-hostname
master-user=slave-user
master-password=slave-password
server-id=2
```

The slave user and slave password are those to which you set when you granted `REPLICATION SLAVE` permission on the master. The `server-id` must be a unique number, different to the master or any other slaves in the system. There are also two other options: `master-port`, used if the master is running on a non-standard port (3306 is default), and `master-connect-retry`, a time in seconds for the slave to attempt to reconnect if the master goes down. The default is 60 seconds.

Restore the data from the master, either as you would normally restore a backup or with the statement `LOAD DATA FROM MASTER`. The latter will lock the master for the duration of the operation, which could be quite lengthy, so you might not be able to spare the downtime.

Customizing content transformations

This task describes how to customize content transformations.

- Download the [content-services-context.xml](#) file:
- Paste this file into the `<extension>` directory.
- Open the file.

Transformers start below the comment:

```
<!-- Content Transformations -->
```

- Locate the bean containing a transformer that is most similar to the transformer that you want to add.

(It is unlikely that you would want to modify an existing transformer.)

- Delete every pair of `<bean>` `</bean>` tags except the pair containing the similar transformer.
- Rename and modify the bean.
- Save the file.

If you save the file in the `<extension>` directory, the filename must end with `#context.xml`.

Controlling indexes

You can use the `cm:indexControl` aspect to control the indexing of content in Alfresco Share. Using this aspect you can choose to disable repository-wide indexing. This can prove useful in certain situations, such as bulk loading.

The `cm:indexControl` aspect enables you to control indexing for the nodes to which it is applied. The aspect exposes the following two properties:

- `cm:isIndexed ((content + metadata))`: This property controls whether or not the node is indexed.
- `cm:isContentIndexed`: This property controls whether or not the node content (binary) is indexed. Setting this to `false` inhibits full text indexing of the document binary.

The following table shows the possible combinations of settings along with the behavior for each case:

<code>cm:isIndexed</code>	<code>cm:isContentIndexed</code>	Result
True	True	Metadata is indexed. Content is indexed.
True	False	Metadata is indexed. Content is not indexed.
False	True	No indexing at all.
False	False	No indexing at all.

For more information on working with aspects, see [Managing aspects](#).

Deferring the start of cron based jobs

You can configure `alfresco-global.properties` and `dev-log4j.properties` to implement a global delay to cron based jobs; for example, until after the server has fully started.

You can set a delay for all cron based jobs; in other words, jobs that use the `org.alfresco.util.CronTriggerBean` class. The default value is 10 minutes.

1. Shut down the Alfresco server.
2. Locate and edit the `alfresco-global.properties` file in the `<classpathRoot>` directory. For information about modifying the `alfresco-global.properties` file, see [Modifying the global properties file](#) on page 61.
3. Add two configurations to the `alfresco-global.properties` file, where the number in `startDelayMins=` is the number of minutes you want to delay your job. In this example, the delay length is 2 minutes:

```
activities.feed.cleaner.cronExpression=0/1 * * * * ?
activities.feed.cleaner.startDelayMins=2
```

4. Extend the `dev-log4j.properties` with a new configuration in the `<classpathRoot>/alfresco/extension` directory:

```
log4j.logger.org.alfresco.repo.activities.feed.cleanup.FeedCleaner=trace
```

This file will override subsystem settings that are not applicable in `alfresco-global.properties`. For more information about log4j extensions, see [log4j.properties file](#).

5. Start the Alfresco server.

After the specified interval, the `FeedCleaner` trace logs will be generated. In the example, the logs will start after two minutes.

Configuring Alfresco subsystems

An Alfresco subsystem is a configurable module responsible for a sub-part of Alfresco functionality. Typically, a subsystem wraps an optional functional area, such as IMAP bindings, or one with several alternative implementations, such as authentication.

A subsystem can be thought of as miniature Alfresco server embedded within the main server. A subsystem can be started, stopped, and configured independently, and it has its own isolated Spring application context and configuration.

The application context is a child of the main context, therefore, it can reference all the beans in the main application context. However, the subsystem's beans cannot be seen by the main application context and communication with the subsystem must be through explicitly imported interfaces. The main features of subsystems are:

Multiple 'instances' of the same type

The same template spring configuration can be initialized with different parameters in different instances. This enables, for example, the chaining of multiple authentication subsystems through property file edits.

Dynamic existence

JMX-based server configuration capabilities in Alfresco releases.

Own bean namespace

There is no problem of bean name uniqueness if you need multiple instances of the same subsystem. Again, this vastly simplifies the problem of building an authentication chain as there is no need to edit any template Spring configuration.

Clearly defined interfaces with the rest of the system

A subsystem's interfaces must be 'imported' in order to be used anywhere else in the system. This is done by mounting them as dynamic proxies.

Hidden implementation specifics

Implementation specifics are not visible because all of its beans are hidden in a private container.

Swapping of alternative implementations

To switch over from native Alfresco authentication to NTLM passthru authentication, you switch to a passthru authentication subsystem and the correct components are swapped in.

Separate product from configuration

A subsystem binds its configuration settings to properties and can even do this for composite data. There is no need to edit or extend prepackaged Spring configuration to configure a subsystem for your own needs.

Subsystem categories

Every subsystem has a category and a type.

- Category is a broad description of the subsystem's function, for example, Authentication.
- Type is a name for the particular flavor of implementation, where multiple alternative implementations exist, for example, `ldap`. Where a subsystem has only one implementation, you can use the default type name of `default`.

The Alfresco-supplied subsystem categories are:

ActivitiesFeed

Handles the activities notifications.

Audit

Handles the audit related functions.

Authentication

Handles all authentication related functions, including:

- Password-based authentication
- Single Sign-on (SSO) for WebClient, WebDAV, Web Scripts, and SharePoint Protocol
- CIFS and FTP authentication
- User registry export (LDAP only)

The subsystem is chained so that multiple instances of different types can be configured and used together.

ContentStore

Handles the properties for the encrypted and non-encrypted Content Stores.

email

Handles the outbound and inbound SMTP property settings.

fileServers

Handles the properties for the CIFS, FTP, and NFS servers.

googledocs

Handles the properties for Google Docs integration.

imap

Handles the properties for the IMAP service.

OOoDirect

Handles the settings for LibreOffice transformations. With this subsystem, the Alfresco server directly manages LibreOffice.

OOoJodconverter

Handles the JODConverter settings for LibreOffice transformations. With this subsystem, the JODConverter manages LibreOffice, including a pool of separate LibreOffice processes, automatic restart of crashed LibreOffice processes, automatic termination of slow LibreOffice operations, automatic restart of any LibreOffice process after a number of operations.

Replication

Handles the settings for the replication jobs tool.

Search

Handles the search mechanism for Alfresco, which can be set either to solr or solr4.

Subscriptions

Handles the settings for the activities feeds.

Synchronization

Performs regular synchronization of local user and group information with the user registry exporters (usually LDAP directories) in the authentication chain.

sysAdmin

Handles the properties for server administration.

thirdparty

Handles the properties for SWF Tools and ImageMagick content transformers.

Transformers

Handles the properties for the transformation server.

wcm_deployment_receiver

Handles the properties for WCM Deployment Receiver.

Subsystem configuration files

The prepackaged subsystem configuration files form part of the core product and should not be edited.

The prepackaged subsystems are found in the `<configRoot>/classes/alfresco/subsystems` directory.

Each subsystem directory should contain one or more Spring XML bean definition metadata files, with names matching the `*-context.xml` pattern. These files are loaded by the child application context that belongs to the subsystem instance.

The XML bean definitions can contain place holders for properties that correspond to configuration parameters of the subsystem. As per standard Spring conventions, these place holders begin with `${` and end with `}`. In the following example, the value of the `ooo.user` configuration parameter will be substituted into the bean definition when it is loaded:

```
<bean id="userInstallationURI" class="org.alfresco.util.OpenOfficeURI">
  <constructor-arg>
    <value>${ooo.user}</value>
  </constructor-arg>
</bean>
```

There is no need to declare a `PropertyPlaceholderConfigurer` bean. An appropriate one is added into the application context automatically.

Subsystem properties

A subsystem declares default values for all the properties it requires in one or more `.properties` files in its subsystem directory.

For example, there could be a property for the `ooo.user`, containing the following:

```
ooo.user=${dir.root}/oouser
```

Place holders are used for system-wide properties, such as `dir.root` in the `-context.xml` and `.properties` files, as the child application context will recursively expand place holders for its own properties and all the place holders recognized by its parent.

Properties files in the subsystem directory declare the configuration parameters and provide default values where these have not been supplied elsewhere. These files should not be edited in order to configure the subsystem.

Use the following methods to modify the subsystem properties:

- Subsystems and all their composite properties show under the `Alfresco:Type=Configuration` tree in JConsole.
- See [Modifying global properties](#) for more information on how to configure a prepackaged subsystem.
- `-D` options

Mounting a subsystem

A subsystem can be mounted, that is, its existence can be declared to the main server. To mount a subsystem, use the `ChildApplicationContextFactory` bean. This is an object that wraps the Spring application context that owns the subsystem and its beans. It initializes its application context as a child of the main Alfresco context with an appropriate `PropertyPlaceholderConfigurer` that will expand its configuration parameters.

 Any instances that you define should extend the `abstractPropertyBackedBean` definition. The identifier that you define for the bean automatically becomes the subsystem category and defines where the factory will look for configuration files, in the search paths.

1. Open the core `bootstrap-context.xml` file (the file that controls the startup of beans and their order).
2. Locate the following bean definition:

```
<!-- Third party transformer Subsystem -->
<bean id="thirdparty"
class="org.alfresco.repo.management.subsystems.ChildApplicationContextFactory"
```

```
parent="abstractPropertyBackedBean">
  <property name="autoStart">
    <value>true</value>
  </property>
</bean>
```

The `autoStart` property is set to `true`, meaning that the child application context will be refreshed when the server boots up, activating the beans it contains. For subsystems containing background processes or daemons (for example, the file server subsystem), it is very important to set this property, otherwise the subsystem will never activate.

3. Save your file.

Mounting a subsystem with composite properties

A subsystem is limited to flat property sets for its configuration, therefore it is difficult to allow structured data in this configuration. A composite property is a special property whose value is a list of beans.

- For example, the IMAP subsystem is mounted as:

```
<!-- IMAP Subsystem -->
<bean id="imap"
class="org.alfresco.repo.management.subsystems.ChildApplicationContextFactory"

parent="abstractPropertyBackedBean">
  <property name="autoStart">
    <value>true</value>
  </property>
  <property name="compositePropertyTypes">
    <map>
      <entry key="imap.server.mountPoints">

<value>org.alfresco.repo.imap.config.ImapConfigMountPointsBean</value>
      </entry>
    </map>
  </property>
</bean>
```

The subsystem declares a single composite property called `imap.server.mountPoints` with component type `org.alfresco.repo.imap.config.ImapConfigMountPointsBean`.

- The configured value of this composite property is materialized in the child application context as a `ListFactoryBean`. The bean's ID should match the name of the composite property. So, for example, in the IMAP subsystem configuration:

```
<!--The configurable list of mount points - actually a post-processed
composite property! -->
  <bean id="imap.server.mountPoints"
class="org.springframework.beans.factory.config.ListFactoryBean">
  <property name="sourceList">
    <list>
      <!-- Anything declared in here will actually be ignored
and replaced by the configured composite property value, resolved on
initialization -->
      <bean id="Repository_virtual"
class="org.alfresco.repo.imap.config.ImapConfigMountPointsBean">
        <property name="mode">
          <value>virtual</value>
        </property>
        <property name="store">
          <value>${spaces.store}</value>
        </property>
        <property name="path">
          <value>/${spaces.company_home.childname}</value>
        </property>
      </bean>
```

```

        <bean id="Repository_archive"
class="org.alfresco.repo.imap.config.ImapConfigMountPointsBean">
        <property name="mode">
            <value>archive</value>
        </property>
        <property name="store">
            <value>${spaces.store}</value>
        </property>
        <property name="path">
            <value>/${spaces.company_home.childname}</value>
        </property>
    </bean>
</list>
</property>
</bean>

```

Other beans in the subsystem application context can use `imap.server.mountPoints` as though it were a regular list of `ImapConfigMountPointsBeans`.

Extension classpath

The `alfresco-global.properties` file can only be used to define properties that are global to the whole system. You can also control the properties of subsystems that have multiple instances, for example, the Authentication subsystems. To do this, you need to target different values for the same properties, to each subsystem instance. You can use the extension classpath mechanism.

1. Add a property file to your application server's global classpath.
For example, under `$TOMCAT_HOME/shared/classes`.
2. Create the path to match the following pattern to override specific properties of a subsystem instance:

```
alfresco/extension/subsystems/<category>/<type>/<id>/*.properties
```

The `<id>` is the subsystem instance identifier, which will be default for single instance subsystems, or the provided identifier for chained subsystems.

For example, if your authentication chain looked like this:

```
authentication.chain=alfrescoNtlm1:alfrescoNtlm,ldap1:ldap
```

Then you could put property overrides for `alfrescoNtlm1` in the following file:

```
alfresco/extension/subsystems/Authentication/alfrescoNtlm/alfrescoNtlm1/
mychanges.properties
```

The default type and ID of non-chained subsystems is `default`, so you could put overrides for file server properties in the following file:

```
alfresco/extension/subsystems/fileServers/default/default/mychanges.properties
```

Setting up Alfresco authentication and security

The first time you access a vanilla Alfresco installation, you can identify yourself by entering a new user name and password in the **Login** screen. If you log in with the credentials of a user with administrator privileges, you can create additional users and assign them passwords.

In this out-of-the-box set up, you can manage the user base and their passwords manually from within Alfresco.

From here, there are a number of common customizations you might want to make to scale up to the needs of a larger enterprise. For example, you might want to:

- Enable automatic sign-on using operating system credentials or a Single Sign-On (SSO) server to remove the need for a **Login** page

- Delegate authentication responsibility to a central directory server to remove the need to set up users manually

Alfresco security

Alfresco security comprises a combination of authentication and authorization.

Authentication is about validating that a user or principal is who or what they claim to be. Alfresco normally refers to users. A user's credentials can take many forms and can be validated in a number of ways. For example, a password validated against an LDAP directory, or a Kerberos ticket validated against a Microsoft Active Directory Server.

Alfresco includes:

- An internal, password-based, authentication implementation
- Support to integrate with many external authentication environments
- The option to write your own authentication integration and to use several of these options simultaneously

Alfresco can integrate with LDAP, Microsoft Active Directory Server, the Java Authentication and Authorization Service (JAAS), Kerberos, and NTLM. A user ID can also be presented as an HTML attribute over HTTPS to integrate with web-based single-sign-on solutions.

Authorization determines what operations an authenticated user is allowed to perform. There are many authorization models. Popular ones include: Role Based Access Control (RBAC), UNIX-style Access Control Lists (ACLs) and extended ACLs, Windows-style ACLs, and many more. Authorization requirements for the management of records are more detailed and include additional requirements, for example, enforcing access based on security clearance or record state.

Alfresco authorization is based on UNIX-extended ACLs. Each node in the repository has an ACL that is used to assign permissions to users and groups. Operations, such as creating a new node, describe what permissions are required to carry out the operation. ACLs are then used to determine if a given user can execute the operation based on the permissions that have been assigned directly to the user or indirectly through a group. An operation in Alfresco is invoking a method on a public service bean. For example, creating a user's home folder requires invoking methods on several public services; to create the folder, set permissions, disable permission inheritance, and so on. Each public service method invocation will check that the user is allowed to execute the method.

By convention, public service beans are the beans whose names start with capital letters, such as the `NodeService`. You configure the security requirements for public service beans in XML. A given method on a particular service might be available to all users, all users in a specified group, all users with a specified role, or users who have particular permissions on specified arguments to the method or its return value. In addition, for methods that return collections or arrays, their content can be filtered based on user permissions. If the authorization requirements for a method call are not met, the method call will fail and it will throw an `AccessDeniedException`. Non-public beans, such as `nodeService`, do not enforce security; use these only when the enforcement of authorization is not required.

Permission assignments are made in *Access Control Lists* (ACLs), which are lists of *Access Control Entries* (ACEs). An ACE associates an authority (group or user) with a permission or set of permissions, and defines whether the permission is denied or allowed for the authority. Every node has a related ACL. When you create a node, it automatically inherits an ACL from its parent. You can alter this behavior after node creation by breaking inheritance or modifying the ACL.

The XML configuration for permissions also defines a context-free ACL for ACEs that apply to all nodes. For example, you could use this to assign everyone Read access to all nodes regardless

of what individual ACLs any node has set. (See the Permissions section in this chapter for more details on how to modify the permission model.)

```
<!-- Extension to alfresco\model\permissionDefinitions.xml -->
<globalPermission permission="Read" authority="GROUP_EVERYONE" />
```

A check that a user has Read permission for a node is done in two stages. First, the context-free ACL is checked to see if it allows access. If not, the ACL assigned or inherited by the node is checked. A user might be allowed to perform an operation because of permissions assigned to the context-free ACL, assigned to the node's ACL, inherited by the node from its parent, or a combination of all three.

Authentication subsystems

Authentication is one of the categories of the Alfresco subsystem. An authentication subsystem is a coordinated stack of compatible components responsible for providing authentication and identity-related functionality to Alfresco.

Alfresco offers multiple implementations of the authentication subsystem, each engineered to work with one of the different types of back-end authentication server that you have available in your enterprise.

An authentication subsystem provides the following functions to Alfresco:

- Password-based authentication for web browsing, Microsoft SharePoint protocol, FTP, and WebDAV
- CIFS and NFS file system authentication
- Web browser, Microsoft SharePoint protocol, and WebDAV Single Sign-On (SSO)
- User registry export (the automatic population of the Alfresco user and authority database)

The main benefits of the authentication subsystem are:

- Subsystems for all supported authentication types are pre-wired and there is no need to edit template configuration.
- There is no danger of compatibility issues between sub-components, as these have all been pre-selected. For example, your CIFS authenticator and authentication filter are guaranteed to be compatible with your authentication component.
- Common parameters are shared and specified in a single place. There is no need to specify the same parameters to different components in multiple configuration files.
- There is no need to edit the `web.xml` file. The `web.xml` file uses generic filters that call into the authentication subsystem. The `alfresco.war` file is a portable unit of deployment.
- You can swap from one type of authentication to another by activating a different authentication subsystem.
- Your authentication configuration will remain standard and, therefore, more manageable to support.
- Authentication subsystems are easily chained



Functions such as NTLM SSO and CIFS authentication can only be targeted at a single subsystem instance in the authentication chain. This is a restriction imposed by the authentication protocols themselves. For this reason, Alfresco targets these 'direct' authentication functions at the first member of the authentication chain that has them enabled.

Authentication subsystem types

A number of alternative authentication subsystem types exist for the most commonly used authentication protocols. These are each identified by a unique type name.

The following table shows the authentication subsystem types supplied with Alfresco and the optional features they support.

Type	Description	Single sign-on (SSO) support	CIFS authentication	User registry entry
alfrescoNtlm	Native Alfresco authentication	Yes, NTLM	Yes	No
ldap	Authentication and user registry export through the LDAP protocol (for example, OpenLDAP)	No	No	Yes
ldap-ad	Authentication and user registry export from Active Directory through the LDAP protocol	No	No	Yes
passthru	Authentication through a Windows domain server	Yes, NTLM	Yes	No
kerberos	Authentication through a Kerberos realm	Yes, SPNEGO	Yes	No
external	Authentication using an external SSO mechanism	Yes	No	No

 If you configure a single authentication subsystem of a type that does not support CIFS authentication (for example, LDAP), then the CIFS server will be automatically disabled. If you want CIFS and LDAP, then you must set up an authentication chain.

Authentication subsystem components

There are a number of main components in an authentication subsystem.

authentication component

Handles the specifics of talking to the back-end authentication system.

authentication Data Access Object (DAO)

Decides what user management functions are allowed, if any. For example, the ability to create a user.

authentication service

Wraps the authentication component and DAO with higher-level functions.

user registry export service (optional)

Allows Alfresco to obtain user attributes, such as email address, organization, and groups automatically.

authentication filters

Provide form or SSO-based login functions for the following:

- web client
- WebDAV
- web scripts
- SharePoint protocol

file server authenticators

Provide authentication functions for the following:

- CIFS protocol (optional)
- FTP protocol

Authentication chains

The authentication subsystem types allow you to integrate Alfresco with the authentication servers in your environment. However, if integrating Alfresco with only one of these systems is not sufficient, you might want to combine multiple authentication protocols against a collection of servers.

Authentication and identity management functionality is provided by a prioritized list, or chain, of configurable subsystems. The built-in authentication chain is a priority-ordered list of authentication subsystem instances. Alfresco composes together the functions of the subsystems in this list into a more powerful conglomerate.

An authentication subsystem provides the following functionality to Alfresco:

- Password-based authentication for web browsing, SharePoint, FTP, and WebDAV
- CIFS and NFS file system authentication
- Web browser and SharePoint Single Sign on (SSO)
- User register export (the automatic population of the Alfresco user and authority database)

Several alternative authentication subsystems exist for the most commonly used authentication protocols. These subsystems enable you to tie Alfresco to some of the most widely used authentication infrastructures. If you include more than one of these subsystems in the chain, you can create complex authentication scenarios.

Authentication chain functions

The functions of the chain are composed in two different ways: chained functions and pass-through functions.

Chained functions

Chained functions combine together functions of more than one subsystem.

For example, when a user logs in, Alfresco tries to match the user's credentials against each of the subsystems in the chain in order.

- If a chain member accepts the credentials, the log in succeeds
- If no chain member accepts, the log in fails

User registry export is also chained. During a synchronize operation, users and groups are exported from each member of the chain supporting user registry export (that is, those of type LDAP) and imported into Alfresco. Ordering in the chain is used to resolve conflicts between users and groups existing in the same directory.

Pass-through functions

Pass-through functions cannot be chained and instead pass through to a single member of the chain, which handles them directly.

Examples of pass-through functions are:

- NTLM / SPNEGO - based Single Sign-On (SSO)
- CIFS Authentication

Such pass-through functions are handled by the first member of the chain that supports that function and has it enabled.

 This means that only a subset of your user base might be able to use SSO and CIFS.

Configuring authentication

A number of examples demonstrate how to express various authentication configuration requirements in subsystem instances in the authentication chain. They also explain how the authentication chain integrates the functions of multiple subsystem instances into a more powerful conglomerate, letting you cater for even the most complex authentication scenarios. These examples demonstrate the flexibility and power of an Alfresco authentication chain. You can combine the strengths of a variety of different authentication protocols and keep the Alfresco user database synchronized almost transparently.

The authentication configuration examples adopt the following structured approach:

1. Decide the authentication chain composition (required subsystem types, instance names, order of precedence) and express this in the `alfresco-global.properties` file.
2. For each subsystem instance:
 - a. Locate the properties files for its subsystem type. These properties files define the configurable properties for that subsystem type and their default values.
 - b. Create a folder named after the subsystem instance under the Alfresco extension folders.
 - c. Copy the properties files into your new folder.
 - d. Edit the properties files to record the desired configuration of the subsystem instance.

Default authentication chain

The default product configuration has a simple chain with one member. This is an instance of the `alfrescoNtlm` subsystem type with an ID of `alfrescoNtlm1`.

This is expressed in the built-in defaults in the `repository.properties` file as:

```
authentication.chain=alfrescoNtlm1:alfrescoNtlm
```

You can configure the properties of `alfrescoNtlm1` using the global properties file.

 This subsystem instance does not have SSO enabled, by default.

To switch from password-based login to NTLM-based SSO, set the following property in the `alfresco-global.properties` file.

```
ntlm.authentication.sso.enabled=true
```

This basic use of NTLM requires Alfresco to store its own copies of your MD4 password hash, which means your user ID and password must be the same in both Alfresco and your Windows domain.

For direct authentication with a Windows domain server, without the need to synchronize accounts in Alfresco and the domain, use the pass-through (`passthru`) subsystem type.

Configuring the authentication chain

This section describes how you can add to or completely replace the default authentication chain.

The chain is controlled by the `authentication.chain` global property.

1. Open the `alfresco-global.properties` file.
2. Locate the `authentication.chain` global property.

This is a comma separated list of the form:

```
instance_name1:type1,...,instance_namen:typen
```

3. Set the property to the required values.

For example, set the property to the following value:

```
alfrescoNtlm1:alfrescoNtlm,ldap1:ldap
```

When you navigate to the

`Alfresco:Type=Configuration,Category=Authentication,id1=manager` MBean in global property overrides, then a new authentication subsystem instance called `ldap1` will be brought into existence and added to the end of the authentication chain.

4. Save the file.

The following example integrates Alfresco with Active Directory has the requirements:

- Built-in Alfresco users and Windows users should be able to log in, with Alfresco taking precedence
- The Windows domain server should handle CIFS authentication directly
- LDAP should be used to synchronize user and group details

To achieve these requirements, configure the following authentication chain:

```
alfrescoNtlm1:alfrescoNtlm,passthru1:passthru,ldap1:ldap
```

Next, deactivate SSO in order to activate chained password-based log in, target CIFS at `passthru1` and target synchronization (but not authentication) at `ldap1` by setting the properties as follows:

alfrescoNtlm1

```
ntlm.authentication.sso.enabled=false
alfresco.authentication.authenticateCIFS=false
```

passthru1

```
ntlm.authentication.sso.enabled=false
passthru.authentication.authenticateCIFS=true
```

ldap1

```
ldap.authentication.active=false
ldap.synchronization.active=true
```

Authentication chain example with JConsole

This section describes an example walk through of setting up an authentication chain using the JMX client, JConsole.

The default authentication within Alfresco is adequate for small-scale environments, however, you might prefer to choose an authentication method that will scale up to a production environment.

For example, you can:

- Enable automatic sign-on using operating system credentials or a single sign-on (SSO) server, which removes the need for a login page
- Delegate authentication responsibility to a central directory server, which removes the need to set up users manually with the **Users** tool.

Alfresco authentication chain

Authentication and identity management functionality is provided by a prioritized list, or chain, of configurable subsystems.

An authentication subsystem provides the following functionality to Alfresco:

- Password-based authentication for web browsing, SharePoint, FTP, and WebDAV
- CIFS and NFS file system authentication

- Web browser and Sharepoint Single Sign On (SSO)
- User register export (the automatic population of the Alfresco user and authority database)

Several alternative authentication subsystems exist for the most commonly used authentication protocols. These subsystems enable you to tie Alfresco to some of the most widely used authentication infrastructures. If you include more than one of these subsystems in the chain, you can create complex authentication scenarios.

Configuring `alfrescoNtlm`

`alfrescoNtlm` is the subsystem configured by default in the Alfresco authentication chain. It performs authentication based on user and password information stored in the Alfresco repository. It is capable of supporting both form-based login and NTLM-based Single Sign-On (SSO), as well as providing authentication for the CIFS server.

 The NTLM SSO functions are disabled by default, which means there are no assumptions about the availability of a Windows domain. You can activate SSO with a single property, without any changes to the `web.xml` file or further file server configuration.

Alfresco NTLM subsystem

The `alfrescoNtlm` subsystem supports optional NTLM Single Sign-On (SSO) functions for WebDAV.

 NTLM v2 is supported, which is more secure than the NTLM v1. If the client does not support NTLMv2, it will automatically downgrade to NTLMv1.

By using NTLM authentication to access Alfresco WebDAV sites, the web browser can automatically log in.

When SSO is enabled, Internet Explorer will use your Windows login credentials when requested by the web server. Firefox and Mozilla also support the use of NTLM but you need to add the URI to the Alfresco site that you want to access to `network.automatic-ntlm-auth.trusted-uris` option (available through writing `about:config` in the URL field) to allow the browser to use your current credentials for login purposes.

The Opera web browser does not support NTLM authentication. The browser is detected and will be sent to the usual Alfresco logon page.

In this configuration, Alfresco must still store its own copy of your MD4 password hash. In order to remove this need and authenticate directly with a Windows domain controller, consider using the pass-through subsystem.

`alfrescoNtlm` configuration properties

The `alfrescoNtlm` subsystem supports the following properties.

`ntlm.authentication.sso.enabled`

A Boolean that when true enables NTLM based Single Sign On (SSO) functionality in the Web clients. When false and no other members of the authentication chain support SSO, password-based login will be used.

`ntlm.authentication.mapUnknownUserToGuest`

Specifies whether unknown users are automatically logged on as the Alfresco guest user during Single Sign-On (SSO).

`alfresco.authentication.authenticateCIFS`

A Boolean that when true enables Alfresco-internal authentication for the CIFS server. When false and no other members of the authentication chain support CIFS authentication, the CIFS server will be disabled.

`alfresco.authentication.allowGuestLogin`

Specifies whether to allow guest access to Alfresco.



If you add extra administrator users in the `authority-services-context.xml` file and are using `alfrescoNtlm`, the extra users (other than the admin user) will no longer have administrator rights until you add them to the `ALFRESCO_ADMINISTRATORS` group.

Configuring Alfresco Share SSO to use NTLM

This section describes how to configure NTLM with Alfresco Share SSO.

Alfresco Share exists as a separate web application to the main Alfresco repository WAR file. It can run in the same application server instance on the same machine as the main web application, or it can run on a completely separate application server instance on a different machine. Share uses HTTP(S) to communicate with the configured Alfresco repository.

1. Locate the following configuration file:

```
<web-extension>\share-config-custom.xml
```

2. Edit the file, and then uncomment the following section:

```
<!--
    SSO authentication config for Share
    NOTE: change localhost:8080 below to appropriate alfresco server
    location if required
-->
<config evaluator="string-compare" condition="Remote">
  <remote>
    <connector>
      <id>alfrescoCookie</id>
      <name>Alfresco Connector</name>
      <description>Connects to an Alfresco instance using cookie-
based authentication</description>

      <class>org.alfresco.web.site.servlet.SlingshotAlfrescoConnector</class>
    </connector>

    <endpoint>
      <id>alfresco</id>
      <name>Alfresco - user access</name>
      <description>Access to Alfresco Repository WebScripts that
require user authentication</description>
      <connector-id>alfrescoCookie</connector-id>
      <endpoint-url>http://localhost:8080/alfresco/wcs</endpoint-
url>

      <identity>user</identity>
      <external-auth>true</external-auth>
    </endpoint>
  </remote>
</config>
```

3. Change the `<endpoint-url>http://localhost:8080/alfresco/wcs</endpoint-url>` value to point to your Alfresco server location.
4. Set the `maxThreads` option in the `<TOMCAT_HOME>/conf/server.xml` file.

```
<Connector port="8080" protocol="HTTP/1.1"
  connectionTimeout="20000"
  redirectPort="8443"
  maxThreads="200"
/>
```



If Share and Alfresco are installed on the same Tomcat, it is important to set the `maxThreads` option to $2 \times$ (expected number of concurrent requests). This is because each Share request spawns an Alfresco request.

5. Restart Share.

If you have configured `alfrescoNtlm` or `passthru` in your Alfresco authentication chain and enabled SSO, NTLM will be the active authentication mechanism.

Share SSO log in bypass

When configuring Share authentication as NTLM SSO, you can bypass the SSO authentication so that it is possible to log in as a different user than the one used in the Windows version.

To log in with another user credential to Share, use:

```
http://localhost:8080/share/page/type/login
```

To log out from Share back to the default NTLM credentials, use:

```
http://localhost:8080/share/page/dologout
```

Configuring pass-through

The pass-through (`passthru`) subsystem can be used to replace the standard Alfresco user database with a Windows server/domain controller, or list of servers, to authenticate users accessing Alfresco. This saves having to create user accounts within Alfresco.

The subsystem also supports optional NTLM Single Sign-On (SSO) functions for WebDav and Alfresco Share, and direct CIFS authentication for the CIFS server. This method of authentication is much more secure than simple LDAP-based authentication or form-based authentication.

 Only NTLM v1 is supported in this configuration. As NTLMv2 has been designed to avoid "man-in-the-middle" attacks, it would be impossible to use in this pass through style.

Pass-through configuration properties

The `passthru` subsystem supports domain level properties.

Also relevant are the configuration steps described in Alfresco Share SSO using NTLM if you want to enable NTLM-based Single Sign-On (SSO) for Alfresco Share.

Domain level properties

The following properties control the set of domain controllers used for authentication. The three properties are mutually exclusive. For example, to set the `passthru.authentication.servers` property, set `passthru.authentication.domain` to be empty and `passthru.authentication.useLocalServer` to be false.

passthru.authentication.useLocalServer

A Boolean that when true indicates that the local server should be used for pass through authentication by using loopback connections into the server.

passthru.authentication.domain

Sets the domain to use for pass through authentication. This will attempt to find the domain controllers using a network broadcast. Make sure that you use the Windows NetBIOS domain name, not the forest name. The network broadcast does not work in all network configurations. In this case use the `passthru.authentication.servers` property to specify the domain controller list by name or address.

passthru.authentication.servers

A comma delimited list of server names or addresses that are used for authentication. The pass through authenticator will load balance amongst the available servers, and can monitor server online/offline status.

- Each server name/address can be prefixed with a domain name using the format `<domain>\<server>`. If specifying this in `alfresco-global.properties`, remember that the backslash character must be escaped. For example


```
passthru.authentication.servers=DOMAIN1\\host1.com,DOMAIN2\\host2.com,host1.com
```
- If the client specifies a domain name in its login request, then the appropriate server will be used for the authentication. Domain mappings can also be specified to route authentication requests to the appropriate server.
- If a server handles authentication for multiple domains then multiple entries can be added in the server list prefixed with each domain name.
- There must be at least one entry in the server list that does not have a domain prefix. This is the catch all entry that will be used if the client domain cannot be determined from the NTLM request or using domain mapping.

Other pass-through properties

ntlm.authentication.sso.enabled

A Boolean that when true enables NTLM based Single Sign On (SSO) functionality in the Web clients. When false and no other members of the authentication chain support SSO, password-based login will be used.

ntlm.authentication.mapUnknownUserToGuest

Identifies whether unknown users are automatically logged on as the Alfresco guest user during Single Sign-On (SSO).

passthru.authentication.authenticateCIFS

A Boolean that when true enables pass-through authentication for the CIFS server. When false and no other members of the authentication chain support CIFS authentication, the CIFS server will be disabled.

passthru.authentication.authenticateFTP

A Boolean that when true enables pass-through authentication for the FTP server. The provided password is hashed and checked directly against the domain server securely using NTLM. When false and no other members of the authentication chain support FTP authentication, standard chained authentication will be used.

passthru.authentication.guestAccess

Identifies whether to allow guest access to Alfresco if the authenticating server indicates the login was allowed guest access.

passthru.authentication.defaultAdministratorUserNames

A comma separated list of user names who should be considered administrators by default. It is often useful to add the administrator user to this list.

passthru.authentication.connectTimeout

The timeout value when opening a session to an authentication server, in milliseconds. The default is 5000.

passthru.authentication.offlineCheckInterval

Specifies how often pass through servers that are marked as offline are checked to see if they are now online. The default check interval is 5 minutes. The check interval is specified in seconds.

passthru.authentication.protocolOrder

Specifies the type of protocols and the order of connection for pass through authentication sessions. The default is to use NetBIOS, if that fails then try to connect using native SMB/ port 445. Specify either a single protocol type or a comma delimited list with a primary and secondary protocol type. The available protocol types are NetBIOS for NetBIOS over TCP and TCPIP for native SMB.

Domain mappings

Domain mappings are used to determine the domain a client is a member of when the client does not specify its domain in the login request. If the client uses a numeric IP address to access the web server it will not send the domain in the NTLM request as the browser assumes it is an Internet address.

To specify the domain mapping rules that are used when the client does not supply its domain in the NTLM request you can use the `filesystem.domainMappings` composite property of the file server subsystem. Specify the file server subsystem settings in the `alfresco-global.properties` file.

There are two ways of defining a domain mapping, either by specifying an IP subnet and mask, or by specifying a range of IP addresses. The following example defines mappings for two domains: ALFRESCO and OTHERDOM.

```
filesystem.domainMappings=ALFRESCO,OTHERDOM
filesystem.domainMappings.value.ALFRESCO.subnet=192.168.1.0
filesystem.domainMappings.value.ALFRESCO.mask=192.168.1.0
filesystem.domainMappings.value.OTHERDOM.rangeFrom=192.168.1.0
filesystem.domainMappings.value.OTHERDOM.rangeTo=192.168.1.100
```

The mask value masks the IP address to get the subnet part, and in this example, the mask value is 192.168.1.0. An alternative is to use 255.255.255.0. A value of 255.255.255.0 will get the subnet, which is then checked against the subnet value. If there were two subnets, 192.168.1.0 and 192.168.2.0, then a mask value of 255.255.255.0 and subnet value of 192.168.1.0 would only match addresses in the 192.168.1.0 range.

The pass through subsystem can use the domain prefixed server name format of the `passthru.authentication.servers` property along with the domain mappings to route authentication requests to the appropriate server. A sample NTLM authentication component server list:

```
passthru.authentication.servers=ALFRESCO\\ADSERVER,OTHERDOM\\OTHERSRV
```

Example: customizing the pass-through subsystem

The authentication capabilities offered by the `ldap-ad` subsystem type cannot support CIFS and NTLM authentication. Instead, you would have to use form-based login for all users, and only Alfresco internal users could access CIFS. This is the compromise you would have to make if the directory server did not support any other authentication protocol. But for Active Directory, which also supports NTLM and Kerberos authentication, you can overcome this limitation by using either the Pass-through or the Kerberos subsystem types.

The Pass-through subsystem supports SSO, CIFS, and password authentication against a Windows domain server using the NTLM v1 protocol. Many prefer Kerberos for its enhanced security and you could consider it as an alternative.

1. Append an instance of `passthru` to the authentication chain.
2. Name the instance `passthru1`, and declare it by changing the `authentication.chain` property in `alfresco-global.properties` as follows:

```
alfresco.authentication.authenticateCIFS=false
```

 Functions such as NTLM SSO and CIFS authentication can only be targeted at a single subsystem instance in the authentication chain. This is a restriction imposed

by the authentication protocols themselves. For this reason, Alfresco targets these 'direct' authentication functions at the first member of the authentication chain that has them enabled. By disabling CIFS in `alfinst` earlier, `passthru1` has a chance to handle CIFS authentication for its larger user base. SSO is also left disabled in `alfinst`, which means that you can enable it in `passthru1`.

3. Stop `ldap1` from performing authentication.

You can leave that to `passthru1`, which will be authenticating against the same server using more secure protocols. This leaves the `ldap1` user registry export capabilities still active, which you still rely on for account synchronization.

4. Edit the `ldap.authentication.active` property in the `ldap-ad-authentication.properties` file located in your `ldap1` directory as follows:

```
ldap.authentication.active=false
```

5. Create the properties files to configure `passthru1`.

```
mkdir <installLocation>\tomcat\shared\classes\alfresco\extension
\subsystems\
Authentication\passthru\passthru1

cd /d <installLocation>\tomcat\shared\classes\alfresco\extension
\subsystems\
Authentication\passthru\passthru1

copy <installLocation>\tomcat\webapps\alfresco\WEB-INF\classes\alfresco
\subsystems\
Authentication\passthru\*.properties
```

After running the previous commands, two separate properties files should appear in your `passthru1` directory. These are:

- `passthru-authentication-context.properties`
- `ntlm-filter.properties`

Using a similar distinction to the `alfrescoNtlm` subsystem type, `passthru-authentication-context.properties` contains properties relating to core authentication capabilities, whereas `ntlm-filter.properties` groups those properties relating to automatic sign on. Unlike the `alfrescoNtlm` subsystem type, SSO is enabled by default in `passthru` subsystems so there is no need to edit `ntlm-filter.properties`.

The following lines show the set of properties you would need to edit and how to set them:

```
passthru.authentication.servers=DOMAIN1\\host1.com,DOMAIN2\
\host2.com,host1.com
passthru.authentication.domain=# Leave blank
passthru.authentication.guestAccess=false
passthru.authentication.defaultAdministratorUserNames=Administrator,alfresco
```

The following list is a summary of the settings that have been changed:

- `passthru.authentication.servers` — A comma-separated list of domain controller host names, each prefixed by the name of the NetBIOS domain they correspond to and a double backslash. The last member of the list should be a host name without a domain prefix, and this host will be used when a client does not include a domain name in an authentication request.
- `passthru.authentication.domain` — A property that is a less-reliable alternative to `passthru.authentication.servers` and should be left empty.
- `passthru.authentication.defaultAdministratorUserNames` — A list of user IDs who should be given Alfresco administrator privileges by default. Additional users can be made administrators by another administrator if they add those users to the `ALFRESCO_ADMINISTRATORS` group.

Applying the Pass-through example

Restart the Alfresco server.

The main differences to notice are:

- All Active Directory users can point their browser to the Alfresco server and be signed on automatically. (In Internet Explorer, this requires adding the Alfresco server to the Local Intranet security zone.)
- All Active Directory users can access Alfresco as a CIFS file system using their Active Directory credentials.

Configuring LDAP

An LDAP subsystem supports two main functions:

- user authentication - checking a user's ID and password using an LDAP bind operation
- user registry export - exposing information about users and groups to the synchronization subsystem

Either of these functions can be used in isolation or in combination. When LDAP authentication is used without user registry export, default Alfresco person objects are created automatically for all those users who successfully log in. However, they will not be populated with attributes without user registry export enabled. LDAP user registry export is most likely to be used without LDAP authentication when chained with other authentication subsystems. For example, Kerberos against Active Directory, pass-through against ActiveDirectory, and possibly Samba on top of OpenLDAP.

The user registry export function assumes that groups are stored in LDAP as an object that has a repeating attribute, which defines the distinguished names of other groups, or users. This is supported in the standard LDAP schema using the `groupOfNames` type. See the example LDIF file in [OpenLDAP tips](#).

LDAP configuration properties

Both the `ldap` and `ldap-ad` subsystem types support the following configurable properties.

-  The defaults for `ldap` are typical for Open LDAP, and the defaults for `ldap-ad` are typical for Active Directory.

ldap.authentication.active

This Boolean flag, when true enables use of this LDAP subsystem for authentication. It might be that this subsystem should only be used for user registry export, in which case this flag should be set to false and you would have to chain an additional subsystem such as `passthru` or `kerberos` to provide authentication functions.

ldap.authentication.java.naming.security.authentication

The mechanism to use to authenticate with the LDAP server. This should be set to one of the standard values listed here or one of the values supported by the LDAP provider. Sun's LDAP provider supports the following SASL mechanisms. The recommended values are:

simple

The basic LDAP authentication mechanism requiring the user name and password to be passed over the wire unencrypted. You might be able to add SSL for secure access, otherwise this should only be used for testing.

DIGEST-MD5

More secure RFC 2831 Digest Authentication. Note that with Active Directory, this requires your user accounts to be set up with reversible encryption, not the default setting.

ldap.authentication.java.naming.read.timeout

Specifies the read timeout in milliseconds for LDAP operations. If Alfresco cannot get a LDAP response within that period, it aborts the read attempt. The integer should be greater than zero. If the integer is less than or equal to zero, no read timeout is specified, which is equivalent to waiting for the response infinitely until it is received.

ldap.authentication.userNameFormat

Specifies how to map the user identifier entered by the user to that passed through to LDAP.

If set to an empty string (the default for the ldap subsystem), an LDAP query involving `ldap.synchronization.personQuery` and `ldap.synchronization.userIdAttributeName` will be performed to resolve the DN from the user ID dynamically. This allows directories to be structured and does not require the user ID to appear in the DN.

If set to a non-empty value, the substring `%s` in this value will be replaced with the entered user ID to produce the ID passed to LDAP. This restricts LDAP user names to a fixed format. The recommended format of this value depends on your LDAP server.

Active Directory

There are two alternatives:

User Principal Name (UPN)

These are generally in the format of `<sAMAccountName>@<UPN Suffix>`. If you are unsure of the correct suffix to use, use an LDAP browser, such as Softerra, to browse to a user account and find its `userPrincipalName` attribute. For example:

```
%s@domain
```

DN

This requires the user to authenticate with part of their DN, so might require use of their common name (CN) rather than their login ID. It also might not work with structured directory layouts containing multiple organization units (OUs). For example:

```
cn=%s,ou=xyz,dc=domain
```

OpenLDAP

The format used depends on the value chosen for `ldap.authentication.java.naming.security.authentication`.

simple

This must be a DN and would be something like the following:

```
uid=%s,ou=People,dc=company,dc=com
```

DIGEST-MD5

Use this value to pass through the entered value as-is:

```
%s
```

When authenticating against LDAP, users are not always in the same subtree of LDAP. In this situation, it is necessary to support authentication against multiple branches of LDAP. For example, some users who can authenticate using `cn=%s,ou=myCity,ou=myState,o=myCompany` but others can authenticate using `cn=%s,ou=ANOTHERCity,ou=myState,o=myCompany`. Set `ldap.authentication.userNameFormat` to be empty (the default), and then it will derive a query from your `personQuery` to look up a user by UID. This ensures that you can support users in any branch structure.

ldap.authentication.allowGuestLogin

Identifies whether to allow unauthenticated users to log in to Alfresco as the 'guest' user.

ldap.authentication.java.naming.factory.initial

The LDAP context factory to use. There is no need to change this unless you do not want to use the default Sun LDAP context factory.

ldap.authentication.java.naming.provider.url

The URL to connect to the LDAP server, containing its name and port. The standard ports for LDAP are 389 (and 636 for SSL). For example: `ldap://openldap.domain.com:389`

ldap.authentication.escapeCommasInBind

Escape commas in the entered user ID when authenticating with the LDAP server? Useful when using simple authentication and the CN is part of the DN and contains commas.

ldap.authentication.escapeCommasInUid

Escape commas in the entered user ID when deriving an Alfresco internal user ID? Useful when using simple authentication and the CN is part of the DN and contains commas, and the escaped `\`, is pulled in as part of a synchronize operation. If this option is set to true it will break the default home folder provider as space names cannot contain `\` (backslash character).

ldap.authentication.defaultAdministratorUserNames

A comma separated list of user names to be considered administrators by default. If you are using LDAP for all your users, this maps an LDAP user to be an administrator user. This administrator user can then configure the other admin users or groups by add users and/or groups to the `ALFRESCO_ADMINISTRATORS` group using the Share Admin Tools.

If you already have a group of administrators in LDAP, you can add the required LDAP group(s) to the `ALFRESCO_ADMINISTRATORS` group. This can be set without a server restart.

ldap.synchronization.active

This flag enables use of the LDAP subsystem for user registry export functions and decides whether the subsystem will contribute data to the synchronization subsystem. It might be that this subsystem should only be used for authentication, in which case this flag should be set to false.

ldap.synchronization.java.naming.security.authentication

The authentication mechanism used to connect to the LDAP server when performing user registry exports. In versions earlier than 3.4 versions, this property was the same as `ldap.authentication.java.naming.security.authentication`. The property should use one of the standard values covered in the Sun documentation <http://java.sun.com/javase/6/docs/technotes/guides/jndi/spec/jndi/properties.html#pgfId=999247> or one of the values supported by the LDAP provider. Sun's LDAP provider supports the SASL mechanisms documented in <http://java.sun.com/javase/6/docs/technotes/guides/jndi/jndi-ldap.html#SASL>. Recommended values are:

none

Use this option if your LDAP server supports connection without a password. Set to none to allow synchronization by using anonymous bind (note that you will not also need to set the following two properties).

simple

This option is the basic LDAP authentication mechanism requiring the user name and password to be passed over the wire unencrypted. You might be able to add SSL for secure access; otherwise, use this option for testing only.

DIGEST-MD5

This option provides a more secure [<ftp://ftp.isi.edu/in-notes/rfc2831.txt> RFC 2831] digest authentication. With Active Directory, this requires your user accounts to be set up with reversible encryption, not the default setting.

ldap.synchronization.java.naming.security.principal

The LDAP user to connect as for the export operation, if one is required by the `ldap.synchronization.java.naming.security.authentication` authentication mechanism. This should be in the same format as `ldap.authentication.userNameFormat` but with a real user ID instead of `%s`.

This is the default principal to use (only used for LDAP sync when `ldap.synchronization.java.naming.security.authentication=simple`):
`ldap.synchronization.java.naming.security.principal=cn\=Manager,dc
\=company,dc\=com`

ldap.synchronization.java.naming.security.credentials

The password for this user, if required. The password for the default principal (only used for LDAP sync when `ldap.synchronization.java.naming.security.authentication=simple`):
`ldap.synchronization.java.naming.security.credentials=secret`

ldap.synchronization.queryBatchSize

If set to a positive integer, this property indicates that RFC 2696 paged results should be used to split query results into batches of the specified size. This overcomes any size limits imposed by the LDAP server. The default value of 1000 matches the default result limitation imposed by Active Directory. If set to zero or less, paged results will not be used.

ldap.synchronization.groupQuery

The query to select all objects that represent the groups to export. This query is used in full synchronization mode, which by default is scheduled every 24 hours.

ldap.synchronization.groupDifferentialQuery

The query to select objects that represent the groups to export that have changed since a certain time. Should use the placeholder `{0}` in place of a timestamp in the format specified by `ldap.synchronization.timestampFormat`. The timestamp substituted will be the maximum value of the attribute named by `ldap.synchronization.modifyTimestampAttributeName` the last time groups were queried. This query is used in differential synchronization mode, which by default is triggered whenever a user is successfully authenticated that does not yet exist in Alfresco.

ldap.synchronization.personQuery

The query to select all objects that represent the users to export. This query is used in full synchronization mode, which by default is scheduled every 24 hours.

ldap.synchronization.personDifferentialQuery

The query to select objects that represent the users to export that have changed since a certain time. Should use the placeholder `{0}` in place of a timestamp in the format specified by `ldap.synchronization.timestampFormat`. The timestamp substituted will be the maximum value of the attribute named by `ldap.synchronization.modifyTimestampAttributeName` the last time users were queried. This query is used in differential synchronization mode, which by default is triggered whenever a user is successfully authenticated that does not yet exist in Alfresco.

ldap.synchronization.groupSearchBase

The DN below which to run the group queries.

ldap.synchronization.userSearchBase

The DN below which to run the user queries.

ldap.synchronization.modifyTimestampAttributeName

The name of the operational attribute recording the last update time for a group or user.

ldap.synchronization.timestampFormat

The timestamp format. This varies between directory servers.

Active Directory

`yyyyMMddHHmmss'.0Z'`

OpenLDAP

`yyyyMMddHHmmss'Z'`

ldap.synchronization.userIdAttributeName

The attribute name on people objects found in LDAP to use as the uid in Alfresco.

ldap.synchronization.userFirstNameAttributeName

The attribute on person objects in LDAP to map to the first name property in Alfresco.

ldap.synchronization.userLastNameAttributeName

The attribute on person objects in LDAP to map to the last name property in Alfresco.

ldap.synchronization.userEmailAttributeName

The attribute on person objects in LDAP to map to the email property in Alfresco.

ldap.synchronization.userOrganizationalIdAttributeName

The attribute on person objects in LDAP to map to the organizational ID property in Alfresco.

ldap.synchronization.defaultHomeFolderProvider

The default home folder provider to use for people created using LDAP import.

ldap.synchronization.groupIdAttributeName

The attribute on LDAP group objects to map to the group name in Alfresco.

ldap.synchronization.groupType

The group type in LDAP.

ldap.synchronization.personType

The person type in LDAP.

ldap.synchronization.groupMemberAttributeName

The attribute in LDAP on group objects that defines the DN for its members.

ldap.authentication.java.naming.security.protocol

This sets the security protocol to use for connecting with the LDAP server. This property has a single value of `ssl`. Set this property to `ssl` if the configuration of truststore is required. Leave this property unused if the truststore configuration is not required (the connection is not secured).

ldap.authentication.truststore.path

The path to the truststore file on the file system.

ldap.authentication.truststore.passphrase

The password for the truststore.

ldap.authentication.truststore.type

The type of the truststore.

Checking the supported SASL authentication mechanisms

1. Using an LDAP browser, such as the one from Softerra, check the values of the `supportedSASLMechanisms` attributes on the root node of your LDAP server.



The simple authentication method will not be reported because it is not a SASL mechanism.

2. If you use OpenLDAP, you can also query using `ldapsearch`. For example:

```
ldapsearch -h localhost -p 389 -x -b "" -s base -LLL
supportedSASLMechanisms
dn:
```

```
supportedSASLMechanisms: DIGEST-MD5
supportedSASLMechanisms: NTLM
supportedSASLMechanisms: CRAM-MD5
```

Example: authentication and synchronization with one ldap-ad subsystem

This example addresses the more advanced goal of delegating authentication responsibility to a centralized directory server. Most organizations maintain their user database in a directory server supporting the LDAP protocol, such as Active Directory or OpenLDAP.

When integrated with an LDAP server, Alfresco can delegate both the password checking and account setup to the LDAP server, thus opening up Alfresco to your entire enterprise. This avoids the need for an administrator to manually set up user accounts or to store passwords outside of the directory server.

To integrate Alfresco with a directory server, you simply need to include an instance of the ldap or ldap-ad subsystem types in the authentication chain. Both subsystem types offer exactly the same capabilities and should work with virtually any directory server supporting the LDAP protocol. Their only differences are the default values configured for their attributes. The ldap type is preconfigured with defaults appropriate for OpenLDAP, whereas ldap-ad is preconfigured with defaults appropriate for Active Directory.

There are two choices in this scenario: replace or add to the authentication chain.

- Replace the authentication chain

You could remove `alfinst` from the previous example and instead add an instance of `ldap-ad`. This would hand over all authentication responsibility to Active Directory and would mean that the built-in accounts, such as `admin` and `guest`, could not be used.

In this scenario, it would be important to configure at least one user who exists in Active Directory as an administrator and enable the `guest` account in Active Directory, if `guest` access were required. Furthermore, because `ldap-ad` cannot support CIFS authentication (as it requires an MD5 password hash exchange), it would rule out use of the CIFS server for all users and the CIFS server would be disabled.

- Add to the authentication chain

You could instead supplement the existing capabilities of `alfinst` by inserting an `ldap-ad` instance before or after `alfinst` in the chain. This means that you could use the built-in accounts alongside those accounts in the directory server. Furthermore, the built-in accounts could access Alfresco through the CIFS server, since `alfrescoNtlm` is able to drive CIFS authentication.

In this scenario, where you chose to position your `ldap-ad` instance in the chain determines how overlaps or collisions between user accounts are resolved. If an `admin` account existed in both Alfresco and Active Directory, then `admin` would be Alfresco if `alfinst` came first, or Active Directory if the `ldap-ad` instance came first.

This example uses an Active Directory server and configures an instance of the `ldap-ad` subsystem.

1. This example uses the second option to append an instance of `ldap-ad` to the authentication chain. This instance name is `ldap1` and is declared by changing the `authentication.chain` property in the `alfresco-global.properties` file. In addition to the `authentication.chain` property, you need to add the `ntlm.authentication.sso.enabled` property to the `alfresco-global.properties` file.
2. Undo any previous modifications to `alfinst` and disable NTLM-based SSO.

This is done because the `ldap-ad` and `ldap` subsystem types cannot participate in the NTLM handshake, so leaving SSO enabled would prevent any of the Active Directory users from logging in.

3. Disable SSO by opening the `alfresco-global.properties` file in a text editor and editing the `ntlm.authentication.sso.enabled` property as follows:

```
authentication.chain=alfinst:alfrescoNtlm,ldap1:ldap-ad

ntlm.authentication.sso.enabled=false

ldap.authentication.allowGuestLogin=false
ldap.authentication.userNameFormat=%s@domain.com
ldap.authentication.java.naming.provider.url=ldap://
domaincontroller.domain.com:389
ldap.authentication.defaultAdministratorUserNames=Administrator,alfresco
ldap.synchronization.java.naming.security.principal=alfresco@domain.com
ldap.synchronization.java.naming.security.credentials=secret
ldap.synchronization.groupSearchBase=ou=Security Groups,ou=Alfresco\
,dc=domain,dc=com

ldap.synchronization.userSearchBase=ou=User
Accounts,ou=Alfresco,dc=domain,dc=com
```

There are a large number of configurable properties for ldap-ad, which demonstrates the flexibility of Alfresco's LDAP infrastructure. Luckily, because ldap-ad already has sensible defaults configured for a typical Active Directory set up, there are only a few edits you must make to tailor the subsystem instance to your needs.

The following list is a summary of the settings that have been changed:

- `ldap.authentication.allowGuestLogin` — Enables / disables unauthenticated access to Alfresco
- `ldap.authentication.userNameFormat` — A template that defines how Alfresco user IDs are expanded into Active Directory User Principal Names (UPNs) containing a placeholder `%s`, which stands for the unexpanded user ID. A UPN generally consists of the user's account ID followed by an `@` sign and then the domain's UPN suffix. You can check the appropriate UPN suffix for your domain by connecting to the directory with an LDAP browser, browsing to a user account, and looking at the value of the `userPrincipalName` attribute.
- `ldap.authentication.java.naming.provider.url` — An LDAP URL containing the host name and LDAP port number (usually 389) of your Active Directory server
- `ldap.authentication.defaultAdministratorUserNames` — A list of user IDs who should be given Alfresco administrator privileges by default. Another administrator can include more users as administrators by adding those users to the `ALFRESCO_ADMINISTRATORS` group.
- `ldap.synchronization.java.naming.security.principal` — The UPN for an account with privileges to see all users and groups. This account is used by Alfresco to retrieve the details of all users and groups in the directory so that it can synchronize its internal user and authority database. Passwords are never compromised and remain in the directory server.
- `ldap.synchronization.java.naming.security.credentials` — The password for the previous account
- `ldap.synchronization.groupSearchBase` — The Distinguished Name (DN) of the Organizational Unit (OU) below which security groups can be found. You can determine the appropriate DN by browsing to security groups in an LDAP browser.
- `ldap.synchronization.userSearchBase` — The distinguished name (DN) of the Organizational Unit (OU) below which user accounts can be found. You can determine the appropriate DN by browsing to user accounts in an LDAP browser.

Applying the ldap-ad example

This example demonstrates how you can further delegate authentication responsibility to Active Directory, without the automatic sign-on and CIFS browsing capabilities that are available to internal Alfresco users.

1. Restart the Alfresco server.

If you watch the output from Tomcat in the `alfresco.log` in the installation directory, you will eventually see lines similar to the following:

```
13:01:31,225 INFO
[org.alfresco.repo.management.subsystems.ChildApplicationContextFactory]
Starting 'Synchronization' subsystem, ID: [Synchronization, default]

...

13:01:49,084 INFO
[org.alfresco.repo.security.sync.ChainingUserRegistrySynchronizer]
Finished synchronizing users and groups with user registry 'ldap1'

13:01:49,084 INFO
[org.alfresco.repo.security.sync.ChainingUserRegistrySynchronizer]
177 user(s) and 19 group(s) processed

13:01:49,131 INFO
[org.alfresco.repo.management.subsystems.ChildApplicationContextFactory]
Startup of 'Synchronization' subsystem, ID: [Synchronization, default]
complete
```

This is output is from the Synchronization subsystem, the Alfresco subsystem responsible for synchronizing the Alfresco internal user and authority database with all user registries in the authentication chain. Since the authentication chain now provides a user registry, the Synchronization subsystem has some work to do when Alfresco starts up.

2. From the example logs, notice that the Synchronization subsystem automatically created 177 users and 19 groups using attributes, such as email address and group memberships, retrieved from Active Directory through an LDAP query. This reduces the workload of the administrator user.

 The Synchronization subsystem uses an incremental timestamp-based synchronization strategy, meaning that it only queries for changes since the last synchronization run. So after the first start up, further synchronization runs can be almost instantaneous. Because synchronization runs are also triggered by a scheduled nightly job and whenever an unknown user successfully authenticates, you should find that Alfresco always stays synchronized with hardly any effort.

Now, if you enter the Alfresco URL: `http://localhost:8080/share/` into your browser, you can log in using the ID and password of any of the Active Directory users.

 Passwords are validated through an LDAP bind operation on Active Directory in real time. Passwords for Active Directory users are not stored locally.

3. Navigate to a user profile.

Notice that attributes such as email address were populated automatically from Active Directory.

Example: authentication and synchronization with two ldap-ad subsystems

This example uses one Active Directory server and shows authentication as well as user registry export (synchronization) from two ldap-ad subsystems.

The two ldap-ad subsystems used are `ad1` and `ad2`. Both these subsystems use the same Active Directory server but different locations within it (search bases).

1. Add the following properties to the `alfresco-global.properties` file.

```
authentication.chain=alfinst:alfrescoNtlm,ad1:ldap-ad,ad2:ldap-ad
ntlm.authentication.sso.enabled=false
```

2. Create the properties files to configure ad1:

```
mkdir <installLocation>\tomcat\shared\classes\alfresco\extension
\subsystems\
Authentication\ldap-ad\ad1

cd /d <installLocation>\tomcat\shared\classes\alfresco\extension
\subsystems\
Authentication\ldap-ad\ad1

copy <installLocation>\tomcat\webapps\alfresco\WEB-INF\classes\alfresco
\subsystems\
Authentication\ldap-ad\*.properties
```

A single file called `ldap-ad-authentication.properties` now appears in the `ad1` directory. You can edit this file to define your LDAP set up.

The following lines show the set of properties you will typically need to edit and how you might set them for a domain controller for a fictitious domain called `domain.com` for `ldap-ad` subsystem `ad1`.

```
ldap.authentication.allowGuestLogin=false
ldap.authentication.userNameFormat=%s@domain.com
ldap.authentication.java.naming.provider.url=ldap://
domaincontroller.domain.com:389
ldap.authentication.defaultAdministratorUserNames=Administrator,alfresco
ldap.synchronization.java.naming.security.principal=alfresco@domain.com
ldap.synchronization.java.naming.security.credentials=secret
ldap.synchronization.groupSearchBase=ou=ad1,ou=Alfresco\
,dc=domain,dc=com
ldap.synchronization.userSearchBase=ou=ad1,ou=Alfresco,dc=domain,dc=com
```

3. Create the properties files to configure ad2:

```
mkdir <installLocation>\tomcat\shared\classes\alfresco\extension
\subsystems\
Authentication\ldap-ad\ad2

cd /d <installLocation>\tomcat\shared\classes\alfresco\extension
\subsystems\
Authentication\ldap-ad\ad2

copy <installLocation>\tomcat\webapps\alfresco\WEB-INF\classes\alfresco
\subsystems\
Authentication\ldap-ad\*.properties
```

A single file called `ldap-ad-authentication.properties` now appears in your `ad2` directory. You can edit this file to define your LDAP set up.

The following lines show the set of properties you will typically need to edit and how you might set them for a domain controller for a fictitious domain called `domain.com` for `ldap-ad` subsystem `ad2`.

```
ldap.authentication.allowGuestLogin=false
ldap.authentication.userNameFormat=%s@domain.com
ldap.authentication.java.naming.provider.url=ldap://
domaincontroller.domain.com:389
ldap.authentication.defaultAdministratorUserNames=Administrator,alfresco
ldap.synchronization.java.naming.security.principal=alfresco@domain.com
ldap.synchronization.java.naming.security.credentials=secret
ldap.synchronization.groupSearchBase=ou=ad2,ou=Alfresco\
,dc=domain,dc=com
ldap.synchronization.userSearchBase=ou=ad2,ou=Alfresco,dc=domain,dc=com
```

Configuring Kerberos

The Java Authentication and Authorization Service (JAAS) is used within the Kerberos subsystem to support Kerberos authentication of user names and passwords. You can choose to use Kerberos against an Active Directory server in preference to LDAP or NTLM as it provides strong encryption without using SSL. It would still be possible to export user registry information using a chained LDAP subsystem.

The disadvantages of using LDAP authentication against Active Directory compared with JAAS/Kerberos are:

- the simplest approach is to use the SIMPLE LDAP authentication protocol, which should be used with SSL
- AD requires special set up to use digest MD5 authentication (reversible encryption for passwords), which might be difficult retrospectively
- LDAP can use GSSAPI and Kerberos which would be equivalent but this is more difficult to configure and has not been tested

For some pointers and background information on JAAS, the Java Authentication and Authorization Service, refer to the following web sites:

- <http://java.sun.com/products/jaas/>
- http://en.wikipedia.org/wiki/Java_Authentication_and_Authorization_Service

Kerberos configuration properties

To enable full Kerberos support in Alfresco requires that the CIFS server and the SSO authentication filters each have a Kerberos service ticket.

The Kerberos subsystem supports the following properties.

kerberos.authentication.realm

The Kerberos realm with which to authenticate. The realm should be the domain upper cased; an example is that if the domain is `alfresco.org` then the realm should be `ALFRESCO.ORG`.

kerberos.authentication.sso.enabled

A Boolean that when true enables SPNEGO/Kerberos based Single Sign On (SSO) functionality in the web client. When false and no other members of the authentication chain support SSO, password-based login will be used.

kerberos.authentication.authenticateCIFS

A Boolean that when true enables Kerberos authentication in the CIFS server. When false and no other members of the authentication chain support CIFS authentication, the CIFS server will be disabled.

kerberos.authentication.user.configEntryName

The name of the entry in the JAAS configuration file that should be used for password-based authentication. The default value `Alfresco` is recommended.

kerberos.authentication.cifs.configEntryName

The name of the entry in the JAAS configuration file that should be used for CIFS authentication. The default value `AlfrescoCIFS` is recommended.

kerberos.authentication.http.configEntryName

The name of the entry in the JAAS configuration file that should be used for web-based single-sign on (SSO). The default value `AlfrescoHTTP` is recommended.

kerberos.authentication.defaultAdministratorUserNames

A comma separated list of user names who should be considered administrators by default.

kerberos.authentication.browser.ticketLogons

Authentication with Alfresco using a ticket parameter in the request URL. Default is true. Note that WebDAV URLs always accept ticket parameters.

kerberos.authentication.stripUsernameSuffix

A Boolean which when true strips the @domain suffix from Kerberos authenticated user names in CIFS, SPP, WebDAV and the Web Client when false, it enables a multi-domain customer to use the @domain suffix.

For Kerberos to work with user names that contain non-ASCII characters, add the following option to JAVA_OPTS for the Share JVM:

```
-Dsun.security.krb5.msinterop.kstring=true
```

Configuring Kerberos against Active Directory

You can set up accounts for use by Alfresco on a Windows domain controller running Active Directory.

It is important to identify each of the servers in your Alfresco cluster that will be running one or both of the Alfresco repository tier (`alfresco.war`) and Alfresco Share (`share.war`) web applications. See [Configuring Share clustering](#) for supported cluster configurations. These instructions also apply to simple non-clustered installations, where a single `alfresco.war` and `share.war` run on a single host.

These instructions use the following naming conventions for the example server, `server1.alfresco.org`:

- `<host>` is the server host name (without domain name suffix). For example, `server1`.
 - `<hostnetbios>` is the resolved value of the `cifs.serverName` property if the server is part of the Active Directory domain (typically the host name with the letter 'A' appended) or the host name otherwise (without domain name suffix). For example, `server1A`.
 - `<domain>` is the DNS domain. For example, `alfresco.org`.
 - `<domainnetbios>` is the Windows domain NetBIOS name. For example, `alfresco`.
 - `<REALM>` is the DNS domain in upper case. For example, `ALFRESCO.ORG`.
1. On the Windows domain controller, create accounts for the Alfresco CIFS service by repeating the following steps for each server in the cluster that will be running the Alfresco repository tier web application (`alfresco.war`):
 - a. In the Active Directory Users and Computers application, navigate to the **Action > New > User** menu, then enter the full name as `CIFS <host>` and the user login name as `cifs<host>`.
 - b. Click **Next**.
 - c. Enter a password.
 - d. Enable **Password never expires** and disable **User must change password at next logon**.
 - e. Click **Next**.
 - f. Click **Finish**.
 - g. Right-click the new user account name, and then select **Properties**.
 - h. Select the **Account** tab and enable the **Do not require Kerberos preauthentication** option in the **Account Options** section.
 - i. From the command prompt, use the `ktpass` utility to generate key tables for this account as shown:

```
ktpass -princ cifs/<hostnetbios>.<domain>@<REALM> -pass <password> -mapuser
```

```
<domainnetbios>\cifs<host> -crypto RC4-HMAC-NT -ptype
KRB5_NT_PRINCIPAL -out
c:\temp\cifs<host>.keytab -kvno 0
```

- j. Create the Service Principal Names (SPN) for the account using the `setspn` utility.

```
setspn -a cifs/<hostnetbios> cifs<host>
setspn -a cifs/<hostnetbios>.<domain> cifs<host>
```



Remember that `ktpass` might already have added some of these SPNs automatically. You can list the existing SPNs for the account using:

```
setspn -l cifs<host>
```

2. Create accounts for the Alfresco SSO authentication filters by repeating the following steps for each server in the cluster that will be running either the Alfresco repository tier web application (`alfresco.war`) or the Share web application (`share.war`).

- In the Active Directory Users and Computers application, navigate to the **Action > New > User** menu, then enter the full name as `HTTP <host>` and the user log in name as `http<host>`.
- Click **Next**.
- Enter a password.
- Enable **Password never expires** and disable **User must change password at next logon**.
- Click **Next**.
- Click **Finish**.
- Right-click the new user account name, and then select **Properties**.
- Select the **Account** tab and enable the **Do not require Kerberos preauthentication** option in the **Account Options** section.
- From the command prompt, use the `ktpass` utility to generate key tables for this account as shown:

```
ktpass -princ HTTP/<host>.<domain>@<REALM> -pass <password> -mapuser
<domainnetbios>\http<host> -crypto RC4-HMAC-NT -ptype
KRB5_NT_PRINCIPAL -out
c:\temp\http<host>.keytab -kvno 0
```

- j. Create the Service Principal Names (SPN) for the account using the `setspn` utility.

```
setspn -a HTTP/<host> http<host>
setspn -a HTTP/<host>.<domain> http<host>
```

- In the Active Directory Users and Computers application, right click on the `http<host>` user and select **Properties**.
- Select the **Delegation** tab. If you cannot see the **Delegation** tab, do one or both of the following:
 - Check that you ran the specified `setspn` command correctly. Delegation is only intended to be used by service accounts, which should have registered SPNs, as opposed to a regular user account which typically does not have SPNs.
 - Raise the functional level of your domain to Windows Server 2003. To do this:
 - Open **Active Directory Domains and Trusts**.
 - In the console tree, right-click the applicable domain and then click **Raise Domain Functional Level**.
 - In **Select an available domain functional level**, click **Windows Server 2003**, and then click **Raise**.
- In the user **Delegation** tab, select the **Trust this user for delegation to any service (Kerberos only)** check box.

3. Copy the key table files created in steps 1 and 2 to the servers they were named after. Copy the files to a protected area, such as C:\etc\ or /etc.
4. On each server in the cluster that will be running either the Alfresco repository tier web application (alfresco.war) or the Share web application (share.war), repeat the following steps:

- a. Set up the Kerberos ini file to point to the Windows domain controller.

The default location is %WINDIR%\krb5.ini, where %WINDIR% is the location of your Windows directory, for example, C:\Windows\krb5.ini. In this example, our Windows domain controller host name is adsrv.alfresco.org.

```
[libdefaults]
default_realm = ALFRESCO.ORG
default_tkt_enctypes = rc4-hmac
default_tgs_enctypes = rc4-hmac

[realms]
ALFRESCO.ORG = {
    kdc = adsrv.alfresco.org
    admin_server = adsrv.alfresco.org
}

[domain_realm]
adsrv.alfresco.org = ALFRESCO.ORG
.adsrv.alfresco.org = ALFRESCO.ORG
```

 The realm should be specified in uppercase.

The Kerberos ini file for Linux is /etc/krb5.conf.

- b. Set up the Java log in configuration file.

For JBoss 5, open the \$JBASS_HOME/server/default/conf/login-config.xml file. Add entries such as the following inside the <policy> tag. Only include AlfrescoCIFS if the server is to run the Alfresco repository tier application (alfresco.war). Only include ShareHTTP if the server is to run the Alfresco Share web application (share.war).

```
<application-policy name="Alfresco">
  <authentication>
    <login-module
      code="com.sun.security.auth.module.Krb5LoginModule"
      flag="sufficient"/>
    </authentication>
  </application-policy>

<application-policy name="AlfrescoCIFS">
  <authentication>
    <login-module
      code="com.sun.security.auth.module.Krb5LoginModule" flag="required">
      <module-option name="debug">true</module-option>
      <module-option name="storeKey">true</module-option>
      <module-option name="useKeyTab">true</module-option>
      <module-option name="isInitiator">false</module-option>
      <module-option name="doNotPrompt">true</module-option>
      <module-option name="keyTab">C:/etc/cifs<host>.keytab</module-
option>
      <module-option name="principal">cifs/<hostnetbios>.domain</
module-option>
    </login-module>
  </authentication>
</application-policy>

<application-policy name="AlfrescoHTTP">
  <authentication>
```

```

    <login-module
code="com.sun.security.auth.module.Krb5LoginModule" flag="required">
    <module-option name="debug">true</module-option>
    <module-option name="storeKey">true</module-option>
    <module-option name="isInitiator">>false</module-option>
    <module-option name="useKeyTab">true</module-option>
    <module-option name="doNotPrompt">true</module-option>
    <module-option name="keyTab">C:/etc/http<host>.keytab</module-
option>
    <module-option name="principal">HTTP/<host>.<domain></module-
option>
    </login-module>
  </authentication>
</application-policy>

<application-policy name="ShareHTTP">
  <authentication>
    <login-module
code="com.sun.security.auth.module.Krb5LoginModule" flag="required">
    <module-option name="debug">true</module-option>
    <module-option name="storeKey">true</module-option>
    <module-option name="isInitiator">>false</module-option>
    <module-option name="useKeyTab">true</module-option>
    <module-option name="doNotPrompt">true</module-option>
    <module-option name="keyTab">C:/etc/http<host>.keytab</module-
option>
    <module-option name="principal">HTTP/<host>.<domain></module-
option>
    </login-module>
  </authentication>
</application-policy>

```

For other environments, in the `JRE\lib\security` folder (for example, `C:/Alfresco/java/jre/lib/security`), create a file named `java.login.config` with entries as shown. Only include `AlfrescoCIFS` if the server is to run the Alfresco repository tier application (`alfresco.war`). Only include `ShareHTTP` if the server is to run the Alfresco Share web application (`share.war`).

For Windows:

```

Alfresco {
  com.sun.security.auth.module.Krb5LoginModule sufficient;
};

AlfrescoCIFS {
  com.sun.security.auth.module.Krb5LoginModule required
  storeKey=true
  useKeyTab=true
  doNotPrompt=true
  keyTab="C:/etc/cifs<host>.keytab"
  principal="cifs/<hostnetbios>.<domain>";
};

AlfrescoHTTP
{
  com.sun.security.auth.module.Krb5LoginModule required
  storeKey=true
  useKeyTab=true
  doNotPrompt=true
  keyTab="C:/etc/http<host>.keytab"
  principal="HTTP/<host>.<domain>";
};

ShareHTTP
{
  com.sun.security.auth.module.Krb5LoginModule required
  storeKey=true

```

```

    useKeyTab=true
    doNotPrompt=true
    keyTab="C:/etc/http<host>.keytab"
    principal="HTTP/<host>.<domain>";
};

com.sun.net.ssl.client {
    com.sun.security.auth.module.Krb5LoginModule sufficient;
};

other {
    com.sun.security.auth.module.Krb5LoginModule sufficient;
};

```

- c. Enable the login configuration file by adding the following line to the main Java security configuration file, usually at `JRE\lib\security\java.security`.
`login.config.url.1=file:${java.home}/lib/security/java.login.config`
- d. If the Alfresco server is not part of the Active Directory domain, ensure that its clock is kept in sync with the domain controller's, for example, by configuring the domain controller as an NTP server.

Kerberos client configuration

To make Internet Explorer negotiate Kerberos authentication, rather than NTLM, ensure that:

- Alfresco web server is in the Local Intranet security zone.
 Check **Tools > Internet Options > Security > Local Intranet**, and then ensure that a pattern matching the protocol and domain name is included, for example, `http://server.com` or `http://*.company.com` (the IP address does not work).
 - Automatic log on is enabled.
 Check **Tools > Internet Options > Security > Custom Level**, and then ensure Automatic log on with the current user name and password is selected.
1. When using Firefox on Windows as your client, you need to add your Alfresco server name to the `network.negotiate-auth.trusted-uris` variable.
 Access the variable from the special URL: `about:config`.
 2. When using Firefox on Linux, you need to add your Alfresco server name to `network.negotiate-auth.trusted-uris` but you will need, in addition, to get a Kerberos ticket using the `kinit` command.

 The ticket can correspond to a different user than your Linux user name.

For example:

```
kinit user1
```

Where `user1` is an Active Directory user. If the client and the server are on the same machine, you will need to go to the `external` interface. The `loopback` interface will not be able to authenticate. You can view your tickets using `klist`.

Debugging Kerberos

You can debug Kerberos issues using the `log4j` properties in the `alfresco.log` file.

For example:

```
log4j.logger.org.alfresco.web.app.servlet.KerberosAuthenticationFilter=debug
log4j.logger.org.alfresco.repo.webdav.auth.KerberosAuthenticationFilter=debug
```

The following is a sample login output:

```
18:46:27,915 DEBUG [app.servlet.KerberosAuthenticationFilter] New Kerberos auth
request from 192.168.4.95 (192.168.4.95:38750)
```

```
18:46:28,063 DEBUG [app.servlet.KerberosAuthenticationFilter] User user1 logged on via Kerberos
```

Configuring Share Kerberos SSO

1. Configure the Alfresco server.
2. Configure Share.
 - a. Go to the Share `<web-extension>` directory.
 - b. Open the `share-config-custom.xml` file.
 - c. Replace the `realm` and `endpoint-spn` options with the correct values for the AlfrescoHTTP user (used to create the keytab files). The `realm` value should be capitalized.
 - d. Uncomment both the `<config evaluator="string-compare" condition="Remote">` sections.

```

<!-- example port config used to access remote Alfresco server
(default is 8080) -->

<config evaluator="string-compare" condition="Remote">
  <remote>
    <endpoint>
      <id>alfresco-noauth</id>
      <name>Alfresco - unauthenticated access</name>
      <description>Access to Alfresco Repository WebScripts that
do not require authentication</description>
      <connector-id>alfresco</connector-id>
      <endpoint-url>http://localhost:8080/alfresco/s</endpoint-
url>
      <identity>none</identity>
    </endpoint>

    <endpoint>
      <id>alfresco</id>
      <name>Alfresco - user access</name>
      <description>Access to Alfresco Repository WebScripts that
require user authentication</description>
      <connector-id>alfresco</connector-id>
      <endpoint-url>http://localhost:8080/alfresco/s</endpoint-
url>
      <identity>user</identity>
    </endpoint>

    <endpoint>
      <id>alfresco-feed</id>
      <name>Alfresco Feed</name>
      <description>Alfresco Feed - supports basic HTTP
authentication via the EndPointProxyServlet</description>
      <connector-id>http</connector-id>
      <endpoint-url>http://localhost:8080/alfresco/s</endpoint-
url>
      <basic-auth>true</basic-auth>
      <identity>user</identity>
    </endpoint>

    <endpoint>
      <id>activiti-admin</id>
      <name>Activiti Admin UI - user access</name>
      <description>Access to Activiti Admin UI, that requires
user authentication</description>
      <connector-id>activiti-admin-connector</connector-id>
      <endpoint-url>http://localhost:8080/alfresco/activiti-
admin</endpoint-url>
      <identity>user</identity>

```

```

        </endpoint>
    </remote>
</config>

<!--
    Overriding endpoints to reference an Alfresco server with
    external SSO enabled

    NOTE: If alfresco server location is not localhost:8080 then
    also combine changes from the
        "example port config" section below.
    *Optional* keystore contains SSL client certificate + trusted
    CAs.
    Used to authenticate share to an external SSO system such as
    CAS
    Remove the keystore section if not required i.e. for NTLM.

    NOTE: For Kerberos SSO rename the "KerberosDisabled" condition
    above to "Kerberos"

    NOTE: For external SSO, switch the endpoint connector to
    "AlfrescoHeader" and set
        the userHeader to the name of the HTTP header that the
    external SSO
        uses to provide the authenticated user name.
-->

<config evaluator="string-compare" condition="Remote">
    <remote>
        <keystore>
            <path>alfresco/web-extension/alfresco-system.p12</path>
            <type>pkcs12</type>
            <password>alfresco-system</password>
        </keystore>

        <connector>
            <id>alfrescoCookie</id>
            <name>Alfresco Connector</name>
            <description>Connects to an Alfresco instance using
            cookie-based authentication</description>

            <class>org.alfresco.web.site.servlet.SlingshotAlfrescoConnector</
            class>
        </connector>

        <connector>
            <id>alfrescoHeader</id>
            <name>Alfresco Connector</name>
            <description>Connects to an Alfresco instance using header
            and cookie-based authentication</description>

            <class>org.alfresco.web.site.servlet.SlingshotAlfrescoConnector</
            class>
            <userHeader>SsoUserHeader</userHeader>
        </connector>

        <endpoint>
            <id>alfresco</id>
            <name>Alfresco - user access</name>
            <description>Access to Alfresco Repository WebScripts that
            require user authentication</description>
            <connector-id>alfrescoCookie</connector-id>
            <endpoint-url>http://localhost:8080/alfresco/wcs/</
            endpoint-url>
            <identity>user</identity>
            <external-auth>>true</external-auth>
        </endpoint>
    </remote>
</config>

```

```
</remote>
</config>
```

- e. Locate the `<!-- Kerberos settings -->` section and replace `condition=KerberosDisabled` with `condition=Kerberos`.

```
<!-- Kerberos settings -->
  <!-- To enable kerberos rename this condition to "Kerberos" -->
  <config evaluator="string-compare" condition="Kerberos"
  replace="true">
    <kerberos>
```

- f. In the (Sun Java) `jre/lib/security/java.login.config` file, add a new section:

```
ShareHTTP {
  com.sun.security.auth.module.Krb5LoginModule required
  storeKey=true
  useKeyTab=true
  keyTab="/etc/keys/alfrescohttp.keytab"
  principal="HTTP/madona.example.foo";
};
```

- g. Restart the Alfresco server.

3. Configure Active Directory.

- Modify the `alfrescohttp` user created during the Alfresco Kerberos setup.
- In the user **Delegation** tab, tick the **Trust this user for delegation to any service (Kerberos only)** check box.

If you do not see the delegation tab, follow the `Allow a user to be trusted for delegation for specific services` instruction on the Microsoft <http://technet.microsoft.com> website.

- If you cannot see the **Delegation** tab, do one or both of the following:
 - Register a Service Principal Name (SPN) for the user account with the `Setspn` utility in the support tools on your CD. Delegation is only intended to be used by service accounts, which should have registered SPNs, as opposed to a regular user account which typically does not have SPNs.
 - Raise the functional level of your domain to Windows Server 2003.

To raise the domain functional level:

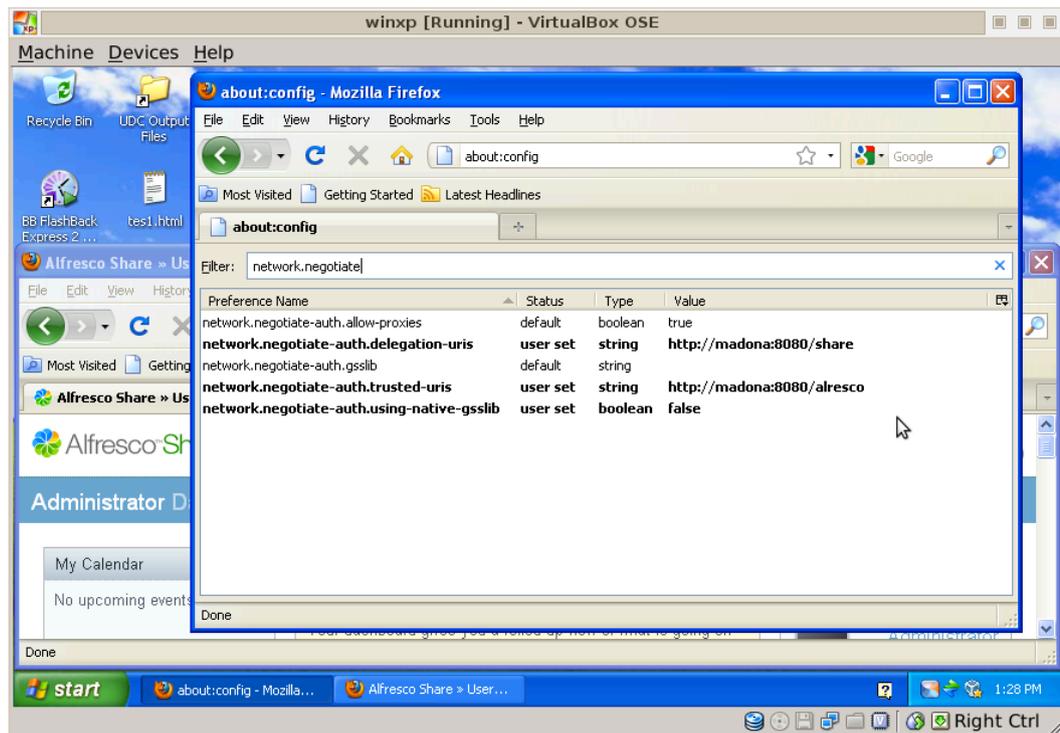
- Open **Active Directory Domains and Trusts**.
- In the console tree, right-click the domain for which you want to raise functionality, and then click **Raise Domain Functional Level**.
- In **Select an available domain functional level**, do one of the following:
 - To raise the domain functional level to Windows 2000 native, click **Windows 2000 native**, and then click **Raise**.
 - To raise domain functional level to Windows Server 2003, click **Windows Server 2003**, and then click **Raise**.

4. Configure the client.

- For Windows client configuration, Internet Explorer configured as described in [Kerberos client configuration](#) should work without modifications.
- To ensure that Firefox works with Windows on the share URL with Kerberos SSO, modify the following variables in the `about:config` special URL:

```
network.negotiate-auth.delegation-uris
network.negotiate-auth.trusted-uris
network.negotiate-auth.using-native-gsslib
```

For example:



Configuring external authentication

The external authentication subsystem can be used to integrate Alfresco with any external authentication system.

The external authentication system can be integrated with your application server in such a way that the identity of the logged-in user is passed to servlets by using the `HttpServletRequest.getRemoteUser()` method. As this is the standard way for application servers to propagate user identities to servlets, it should be compatible with a number of SSO solutions, including Central Authentication Service (CAS).

The subsystem also allows a proxy user to be configured, such that requests made through this proxy user are made in the name of an alternative user, whose name is carried in a configured HTTP request header. This allows, for example, the Share application and other Alfresco Surf applications to act as a client to an SSO-protected Alfresco application and assert the user name in a secure manner.

 Activating external authentication makes Alfresco accept external authentication tokens, make sure that no untrusted direct access to Alfresco's HTTP or AJP ports is allowed.

External configuration properties

The external subsystem supports the following properties.

external.authentication.enabled

A Boolean property that when true indicates that this subsystem is active and will trust remote user names asserted to it by the application server.

external.authentication.defaultAdministratorUserNames

A comma separated list of user names who should be considered administrators by default.

external.authentication.proxyUserName

The name of the remote user that should be considered the proxy user.

The default is `alfresco-system`. Requests made by this user will be made under the identity of the user named in the HTTP Header indicated by the `external.authentication.proxyHeader` property. If not set, then the HTTP Header indicated by the `external.authentication.proxyHeader` property is always assumed to carry the user name.

 This is not secure unless this application is not directly accessible by other clients.

external.authentication.proxyHeader

The name of the HTTP header that carries the name of a proxied user. The default is `x-Alfresco-Remote-User`, as used by Share.

external.authentication.userIdPattern

An optional regular expression to be used to extract a user ID from the HTTP header. The portion of the header matched by the first bracketed group in the regular expression will become the user name. If not set (the default), then the entire header contents are assumed to be the proxied user name.

Configuring Alfresco Share to use an external SSO

Alfresco Share can be configured to accept a user name from an HTTP header provided by an external authentication system for Single Sign on (SSO).

1. Go to the Share `<web-extension>` directory.
2. Open the `share-config-custom.xml` file.
3. Uncomment the second `<config evaluator="string-compare" condition="Remote">` section.

 There are multiple Remote configuration sections in this file. If you have multiple sections in a configuration file, then the last section is used.

4. Change the connector used by the endpoint in the second section to use `alfrescoHeader` rather than `alfrescoCookie`.
5. Set the name of the header used by the external SSO in the `userHeader` element of the `alfrescoHeader` connector.
6. Change the `endpoint-url` value to point to your Alfresco Server location.

```

<!--
    Overriding endpoints to reference an Alfresco server with
    external SSO
    enabled
    NOTE: If utilising a load balancer between web-tier and
    repository
    cluster, the "sticky sessions" feature of your load balancer must
    be used.

    NOTE: If alfresco server location is not localhost:8080 then also
    combine
    changes from the "example port config" section below.
    *Optional* keystore contains SSL client certificate + trusted
    CAs.
    Used to authenticate share to an external SSO system such as CAS
    Remove the keystore section if not required i.e. for NTLM.

    NOTE: For Kerberos SSO rename the "KerberosDisabled" condition
    above to
    "Kerberos"

    NOTE: For external SSO, switch the endpoint connector to
    "AlfrescoHeader"
  
```

```

        and set the userHeader to the name of the HTTP header
        that the external SSO uses to provide the authenticated
user name.
-->

<config evaluator="string-compare" condition="Remote">
  <remote>
    <keystore>
      <path>alfresco/web-extension/alfresco-system.p12</path>
      <type>pkcs12</type>
      <password>alfresco-system</password>
    </keystore>

    <connector>
      <id>alfrescoCookie</id>
      <name>Alfresco Connector</name>
      <description>Connects to an Alfresco instance using cookie-
based
                                authentication
      </description>

      <class>org.alfresco.web.site.servlet.SlingshotAlfrescoConnector</class>
    </connector>

    <connector>
      <id>alfrescoHeader</id>
      <name>Alfresco Connector</name>
      <description>Connects to an Alfresco instance using header
and
                                cookie-based authentication
      </description>

      <class>org.alfresco.web.site.servlet.SlingshotAlfrescoConnector</class>
      <userHeader>ExternalSsoSystemHeader</userHeader>
    </connector>

    <endpoint>
      <id>alfresco</id>
      <name>Alfresco - user access</name>
      <description>Access to Alfresco Repository WebScripts that
require user
                                authentication
      </description>
      <connector-id>alfrescoHeader</connector-id>
      <endpoint-url>http://localhost:8080/alfresco/wcs</endpoint-
url>
      <identity>user</identity>
      <external-auth>true</external-auth>
    </endpoint>
  </remote>
</config>

```

7. Set the `external.authentication.proxyHeader` property to the same value as the `userHeader` value.

This configures both Share and the repository to use the same HTTP header value.

```
external.authentication.proxyHeader=SsoUserHeader
```

8. Save the file and then restart Share.

Activating external authentication makes Alfresco accept external authentication tokens, make sure that no untrusted direct access to Alfresco's HTTP or AJP ports is allowed.

You have now configured Alfresco Share to use an external SSO.

Authorities

Authorities are people (or persons) or groups.

A group can contain people or other groups as members. The authorities assigned to a user at any time are the `userName` from their associated Person node, all of the groups in which the user is a direct or indirect member, and any appropriate dynamic authorities. Dynamic authorities are used for internal roles.

Dynamic authorities and roles

Alfresco uses some custom roles. To implement a custom role, you create a dynamic authority for that role and assign global permissions to it. The Alfresco internal roles have not been assigned any object-specific rights.

The internal roles are:

- `ROLE_ADMINISTRATOR` is assigned to the default administrators for the configured authentication mechanisms or members of the administration groups defined on the `AuthorityServiceImpl` bean. This role has all rights.
- `ROLE_OWNER` is assigned to the owner of a node. If there is no explicit owner, this role is assigned to the creator. This role has all rights on the owned node.
- `ROLE_LOCK_OWNER` is assigned to the owner of the lock on a locked node. This supports a lock owner's right to check in, cancel a check out, or unlock the node.

Alfresco Share supports the assignment of permissions only to the owner role. You can use such things as the Java API and scripting to make other assignments.



Hierarchical and zoned roles can be added to Alfresco in the future to avoid the hidden group implementation for true roles.

People and users

When a user logs in, Alfresco validates the user's identifier and password. Alfresco uses the identifier to look up the appropriate person details for the user, using the `userName` property on the Person type. You can configure this look-up to be case sensitive or case insensitive. The `userName` property on the matching Person node is used as the actual user authority; it might differ in case from the user identifier presented to the authentication system. After the Person node look-up, Alfresco is case sensitive when matching authorities to permissions, group membership, roles, and for all other authorization tests.

Any user, who authenticates by any mechanism, must have an associated person node in Alfresco. Person nodes can be:

- Explicitly created
- Created on demand with some default entries
- Created from LDAP synchronization

Person nodes are explicitly created when using Alfresco Share to manage users.

By default, person nodes are auto-created if not present. If an external authentication system is configured, such as NTLM, when any user authenticates, an appropriate person node might not exist. If a person node does not exist and auto-creation is enabled, a person node will then be created using the identifier exactly as presented by the user and validated by the authentication system. The auto-created Person node's `userName` will have the same case as typed by the user. LDAP synchronization will create person nodes with the `userName`, as provided from the LDAP server.

It is possible that LDAP synchronization can change the `userName` associated with a Person node. For example, this can happen with a system that uses NTLM authentication, LDAP

synchronization, or a system that creates person nodes on demand, or uses case-insensitive authentication. For example, Andy could log in as “Andy” and the associated Person node is created with the userName “Andy.” Later, the LDAP synchronization runs and changes the userName to “andy”.

Changes to Person node userNames will cause updates to other related data in Alfresco, such as ACL assignment.

Groups

Groups are collections of authorities with a name and display name.

Groups can include other groups or people. You can include a group in one or more other groups, as long as this inclusion does not create any cyclic relationships.

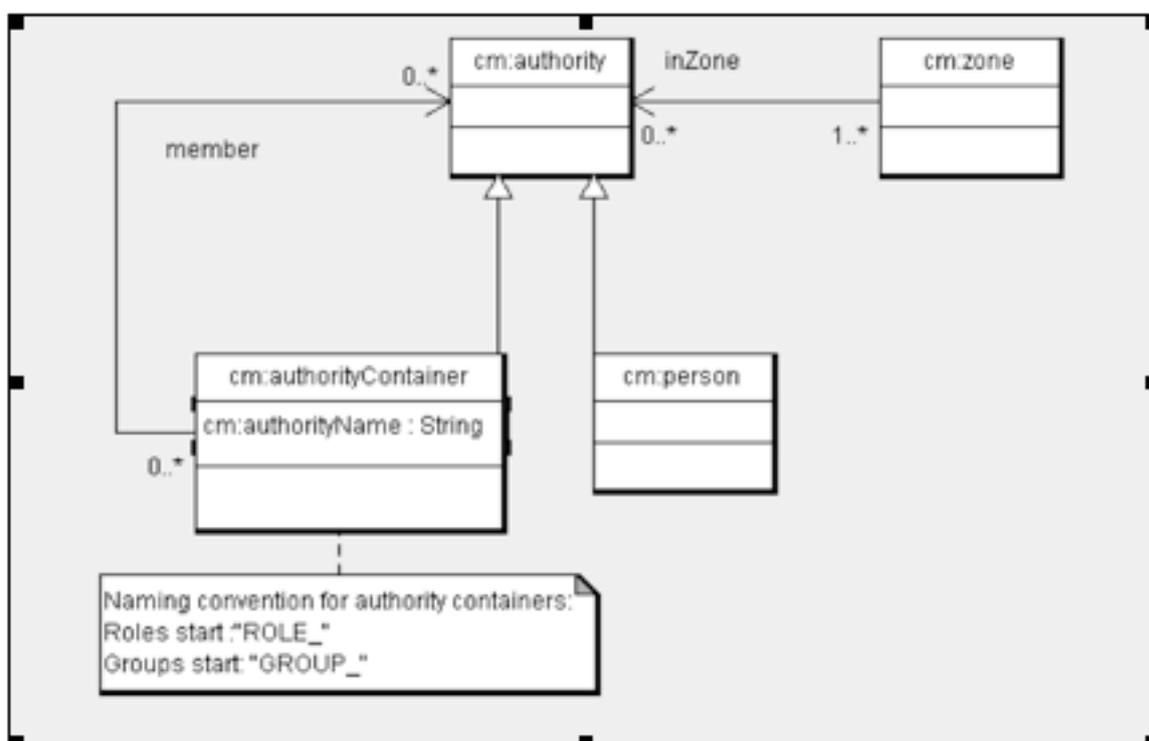
Zones

All person and group nodes are in one or more zones. You can use zones for any partitioning of authorities. For example, Alfresco synchronization uses zones to record from which LDAP server users and groups have been synchronized. Zones are used to hide some groups that provide Role Based Access Control (RBAC) role-like functionality from the administration pages of Alfresco Share. Examples of hidden groups are the roles used in Alfresco Share and Records Management (RM). Only users and groups in the default zone are shown for normal group and user selection on the group administration pages. Zones cannot be managed from the administration pages of Alfresco Share.

Zones are intended to have a tree structure defined by naming convention. Zones are grouped into two areas: Application-related zones and authentication-related zones.

Within a zone, a group is considered to be a root group if it is not contained by another group in the same zone.

Alfresco uses a model for persisting people, groups, and zones. A Person node represents each person, and an AuthorityContainer represents groups, which can be used for other authority groupings such as roles. AuthorityContainer and Person are sub-classes of Authority and as such can be in any number of Zones.



Application-related zones

Application-related zones, other than the default, hide groups that implement RBAC like roles. Application zones, by convention, start APP. and include:

- APP.DEFAULT is for person and group nodes to be found by a normal search. If no zone is specified for a person or group node, they will be a member of this default zone.
- APP.SHARE is for hidden authorities related to Alfresco Share.
- APP.RM will be added for authorities related to RM.

Authorization-related zones

Zones are also used to record the primary source of person and group information. They can be held within Alfresco or some external source. While authorities can be in many zones, it makes sense for an authority to be in only one authentication-related zone.

- AUTH.ALF is for authorities defined within Alfresco and not synchronized from an external source. This is the default zone for authentication.
- AUTH.EXT.<ID> is for authorities defined externally, such as in LDAP.

Defining permissions

Permissions and their groupings are defined in an XML configuration file.

The default file is found in the distribution configuration directory as [permissionDefinitions.xml](#). This configuration can be replaced or extended.

The following example uses the permission definitions related to the Ownable aspect.

```
<!-- ===== -->
<!-- Permissions associated with the Ownable aspect -->
<!-- ===== -->

<permissionSet type="cm:ownable" expose="selected">

    <!-- Permission control to allow ownership of the node to be taken from
others -->
    <permissionGroup name="TakeOwnership" requiresType="false"
expose="false">
        <includePermissionGroup permissionGroup="SetOwner" type="cm:ownable" />
    </permissionGroup>

    <permissionGroup name="SetOwner" requiresType="false" expose="false"/>

    <!-- The low level permission to control setting the owner of a node -->
    <permission name="_SetOwner" expose="false" requiresType="false">
        <grantedToGroup permissionGroup="SetOwner" />
        <requiredPermission on="node" type="sys:base" name="_WriteProperties" /
>
    </permission>
</permissionSet>
```

Permissions and permission groups are defined in a permission set, which is a sub-element of the permissions root element. A permission set is associated with a type or aspect and applies only to that type and sub-types, or aspect and sub-aspects.

A permission has a name. By convention, the names of permissions start with an underscore character. A permission, in its definition, can be granted to any number of permission groups. This means that those permission groups will include the permission. The permission might require that the type or aspect specified on the permission set be present on the node. If a permission is associated with an aspect and the requiresType property is set to true then if that aspect is not applied to a node, the permission does not apply to that node either. If an aspect-

related permission definition has the `requiresType` property set to `false`, the permission applies to any node, even if the aspect has not been applied to the node.

An aspect can be applied at any time and there are no restrictions as to which aspects can be applied to a type. A permission might also require other permissions be tested on the same node, its children, or its parent. In the example, `_SetOwner` requires `_WriteProperties`. This means you cannot set ownership on a node if you are not allowed to write to its properties. You can also use this to check that all children can be deleted before deleting a folder, or to enforce that you can only read nodes for which you can read all the parents; neither are normally required in Alfresco. The configuration to do this is present in the standard configuration file but is commented out. The `_DeleteNode` permission definition (as shown in the following code snippet) is an example. If permission A requires permission B and this requirement is implied (by setting the `implies` attribute of the `requiredPermission` element to `true`), assigning an authority permission A will also give them permission B (as opposed to checking they have permission B).

```
<permission name="_DeleteNode" expose="false" >
  <grantedToGroup permissionGroup="DeleteNode" />
  <!-- Commented out parent permission check ...
  <requiredPermission on="parent" name="_ReadChildren" implies="false"/>
  <requiredPermission on="parent" name="_DeleteChildren" implies="false"/>
  <requiredPermission on="node" name="_DeleteChildren" implies="false"/>
  -->
  <!-- Recursive delete check on children -->
  <!-- <requiredPermission on="children" name="_DeleteNode" implies="false"/>
  -->
</permission>
```

Permissions are normally hidden inside permission groups. Permission groups are made up of permissions and other permission groups. By convention, each permission has a related permission group. Permission groups can then be combined to make other permission groups. As for permissions, a permission group can be exposed by the administration pages of Alfresco Explorer and Alfresco Share and might require the presence of a type or aspect to apply to a particular node. In addition, a permission group can allow full control, which grants all permissions and permission groups. As a type or aspect can extend another, a permission group defined for a type or aspect can extend one defined for one of its parent types and be assigned more permissions, include more permission groups, or change what is exposed in the administration pages of the Alfresco Explorer and Alfresco Share web clients.

It is unusual to extend or change the default permission model unless you are adding your own types, aspects, and related public services or you wish to make minor modifications to the existing behavior. The following code snippets show how to extend and replace the default permission model.

```
<bean id='permissionsModelDAO'
class="org.alfresco.repo.security.permissions.impl.model.PermissionModel" init-
method="init">
  <property name="model">
<-- <value>alfresco/model/permissionDefinitions.xml</value> -->
<value>alfresco/extension/permissionDefinitions.xml</value>
  </property>
  <property name="nodeService">
    <ref bean="nodeService" />
  </property>
  <property name="dictionaryService">
    <ref bean="dictionaryService" />
  </property>
</bean>
```

The preceding code example shows how to replace the default permission model with one located in the `alfresco/extension` directory. The following code snippet shows how to extend the existing model.

```
<bean id="extendPermissionModel" parent="permissionModelBootstrap">
```

```
<property name="model" value="alfresco/extension/
permissionModelExtension.xml" />
</bean>
```

Controlling site creation permissions

By default, any authenticated user can create sites in Share. The creator of the new site is given the Site Manager role and they control who has access to the site and in what role. This topic gives on how to control site creation permissions in Alfresco.

The beans that enforce security to the repository services based on the currently authenticated user are defined in the [public-services-security-context.xml](#) file.

1. Copy the `SiteService_security` bean from the [public-services-security-context.xml](#) file.

```
<bean id="SiteService_security"
class="org.alfresco.repo.security.permissions.impl.acegi.MethodSecurityIntercepto
  <property name="authenticationManager"><ref
bean="authenticationManager" /></property>
  <property name="accessDecisionManager"><ref
local="accessDecisionManager" /></property>
  <property name="afterInvocationManager"><ref
local="afterInvocationManager" /></property>
  <property name="objectDefinitionSource">
  <value>

org.alfresco.service.cmr.site.SiteService.cleanSitePermissions=ACL_NODE.0.sys:bas
org.alfresco.service.cmr.site.SiteService.createContainer=ACL_ALLOW,AFTER_ACL_NO
  org.alfresco.service.cmr.site.SiteService.createSite=ACL_ALLOW
  org.alfresco.service.cmr.site.SiteService.deleteSite=ACL_ALLOW

org.alfresco.service.cmr.site.SiteService.findSites=ACL_ALLOW,AFTER_ACL_NODE.sys:
org.alfresco.service.cmr.site.SiteService.getContainer=ACL_ALLOW,AFTER_ACL_NODE.s
org.alfresco.service.cmr.site.SiteService.listContainers=ACL_ALLOW,AFTER_ACL_NODE
org.alfresco.service.cmr.site.SiteService.getMembersRole=ACL_ALLOW
org.alfresco.service.cmr.site.SiteService.getMembersRoleInfo=ACL_ALLOW
  org.alfresco.service.cmr.site.SiteService.resolveSite=ACL_ALLOW

org.alfresco.service.cmr.site.SiteService.getSite=ACL_ALLOW,AFTER_ACL_NODE.sys:ba
org.alfresco.service.cmr.site.SiteService.getSiteShortName=ACL_ALLOW,AFTER_ACL_NO
  org.alfresco.service.cmr.site.SiteService.getSiteGroup=ACL_ALLOW

org.alfresco.service.cmr.site.SiteService.getSiteRoleGroup=ACL_ALLOW
  org.alfresco.service.cmr.site.SiteService.getSiteRoles=ACL_ALLOW

org.alfresco.service.cmr.site.SiteService.getSiteRoot=ACL_ALLOW,AFTER_ACL_NODE.sy
  org.alfresco.service.cmr.site.SiteService.hasContainer=ACL_ALLOW

org.alfresco.service.cmr.site.SiteService.hasCreateSitePermissions=ACL_ALLOW
  org.alfresco.service.cmr.site.SiteService.hasSite=ACL_ALLOW
  org.alfresco.service.cmr.site.SiteService.isMember=ACL_ALLOW
  org.alfresco.service.cmr.site.SiteService.listMembers=ACL_ALLOW

org.alfresco.service.cmr.site.SiteService.listMembersInfo=ACL_ALLOW
org.alfresco.service.cmr.site.SiteService.listMembersPaged=ACL_ALLOW
org.alfresco.service.cmr.site.SiteService.listSites=ACL_ALLOW,AFTER_ACL_NODE.sys:
org.alfresco.service.cmr.site.SiteService.listSitesPaged=ACL_ALLOW,AFTER_ACL_NODE
org.alfresco.service.cmr.site.SiteService.removeMembership=ACL_ALLOW
```

```

    org.alfresco.service.cmr.site.SiteService.canAddMember=ACL_ALLOW
    org.alfresco.service.cmr.site.SiteService.setMembership=ACL_ALLOW
    org.alfresco.service.cmr.site.SiteService.updateSite=ACL_ALLOW

    org.alfresco.service.cmr.site.SiteService.countAuthoritiesWithRole=ACL_ALLOW
    org.alfresco.service.cmr.site.SiteService.isSiteAdmin=ACL_ALLOW
    org.alfresco.service.cmr.site.SiteService.*=ACL_DENY
  </value>
</property>
</bean>

```

2. Place it in a file called `<extension>/custom-model-context.xml`.
3. Modify the inserted `SiteService_security` bean to match your requirements. For example:

To give permission to only Administrators to create site, change:

```
org.alfresco.service.cmr.site.SiteService.createSite=ACL_ALLOW
```

to

```
org.alfresco.service.cmr.site.SiteService.createSite=ACL_METHOD.ROLE_ADMINISTRATOR
```

where, `ACL_ALLOW` executes a method that allows access to all users and `ACL_METHOD.ROLE_ADMINISTRATOR` executes a method that allows access to users who are members of the administrator group.

4. Save the file.
5. Restart Alfresco.

Access Control Lists

An Access Control List (ACL) is an ordered list of one or more Access Control Entries (ACE). An ACE associates a single authority to a single permission group or permission, and states whether the permission is to be allowed or denied. All nodes have an associated ACL.

There is one special, context-free, ACL defined in the XML configuration to support global permissions. An ACL specifies if it should inherit ACEs from a parent ACL. The parent ACL is associated with the primary parent node. When a new node is created it automatically inherits all ACEs defined on the parent within which it is created. Linking a node to a secondary parent has no effect on ACE inheritance; the node will continue to inherit permission changes from its primary parent (defined when it was first created).

By default, ACL inheritance is always from the primary parent. The underlying design and implementation does not mandate this. ACL inheritance does not have to follow the parent child relationship. It is possible to change this through the Java API.

There are several types of ACL defined in `ACLType`. The main types are:

- DEFINING
- SHARED
- FIXED
- GLOBAL

A node will be associated with an ACL. It will have a DEFINING ACL if any ACE has been set on the node. DEFINING ACLs include any ACEs inherited from the node's primary parent and above, if inheritance is enabled. All DEFINING ACLs are associated with one SHARED ACL. This SHARED ACL includes all the ACEs that are inherited from the DEFINING ACL. If the primary children of a node with a DEFINING ACL do not themselves have any specific ACEs defined then they can be assigned the related SHARED ACL. For the primary children of a node with a SHARED ACL that also have no specific ACEs set they can use the same SHARED ACL. A

single SHARED ACL can be associated with many nodes. When a DEFINING ACL is updated, it will cascade update any related ACLs by using the ACL relationships rather than walk the node structure. If a DEFINING ACL inherits ACEs, then these will come from the SHARED ACL related to another DEFINING ACL.

ACLs and nodes have two linked tree structures.

FIXED ACLs are not associated with a node but found by name. A node ACL could be defined to inherit from a fixed ACL. A GLOBAL ACL is a special case of a FIXED ACL with a well known name. It will be used to hold the global ACE currently defined in XML.

ACEs comprise an authority, a permission, and a deny/allow flag. They are ordered in an ACL.

ACL ordering and evaluation

The ACEs within an ACL are ordered and contain positional information reflecting how an ACE was inherited. DEFINING ACLs have entries at even positions; SHARED ACLs have entries at odd positions. For a DEFINING ACL, any ACEs defined for that ACL have position 0, any inherited from the parent ACL have position two, and so on. For a SHARED ACL, ACEs defined on the ACL from which it inherits will have position one.

When Alfresco makes permission checks, ACEs are considered in order with the lowest position first. Deny entries take precedence over allow entries at the same position. Once a deny entry is found for a specific authority and permission combination, any matching ACE, at a higher position from further up the inheritance chain, is denied. A deny for one authority does not deny an assignment for a different authority. If a group is denied Read permission, a person who is a member of that group can still be assigned Read permission using another group or directly with their person `userName`. However, if an authority is granted Read (made up of `ReadContent` and `ReadProperties`) and the same authority denied `ReadContent`, they will just be granted `ReadProperties` permission. The administration pages of Alfresco Share do not expose deny.

The default configuration is any deny denies. This is set by adding the following property to the `alfresco-global.properties` file:

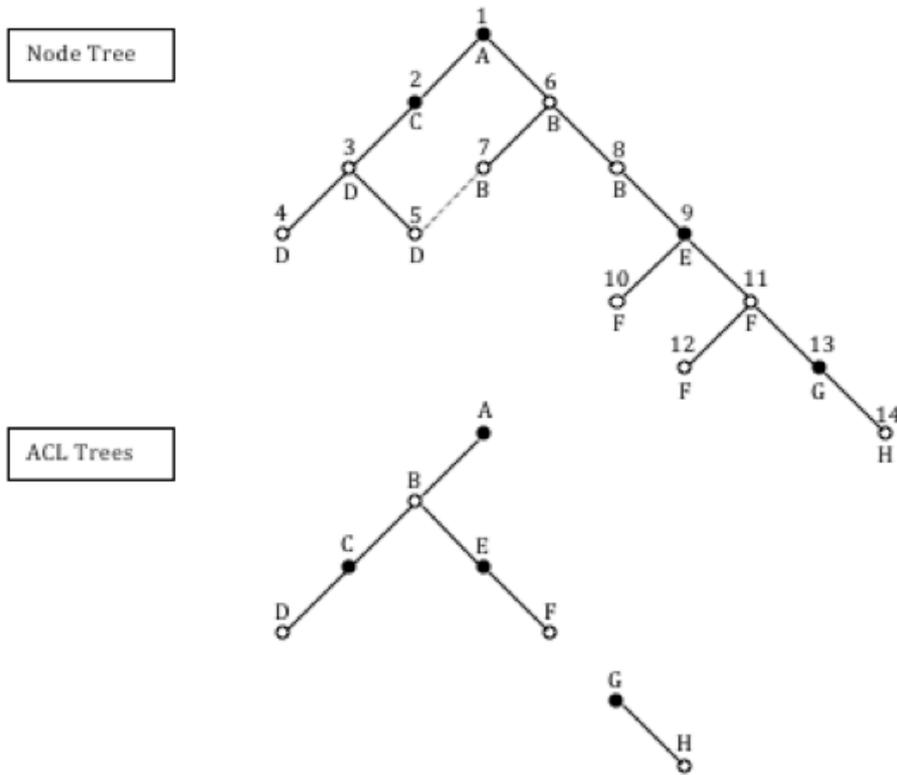
```
security.anyDenyDenies=true
```

You can alter the configuration to support any allow allows. This is set by adding the following property to the `alfresco-global.properties` file:

```
security.anyDenyDenies=false
```

An ACL example

This example relates a tree of nodes to two corresponding trees of ACLs. The nodes in the node tree are identified by number and are shown filled in black if they have any ACEs set, or white/clear if not. Primary child relationships are drawn as black lines and secondary child relationships as dashed lines. ACLs in the ACL trees are identified by letter, DEFINING ACLs are shown filled in black, and SHARED ACLs are shown as clear. Under each node on the node tree the related ACL is referenced.



The table describes the ACEs in each ACL and their position.

Table 1: ACL formats

ACL format	Authority	Permission	Allow/Deny	Position
ACL A (Defining, no inheritance)	All	Read	Allow	0
ACL B (Shared, inherits from ACL A)	All	Read	Allow	1
ACL C (Defining, inherits from ACL B)	All	Read	Allow	2
	ROLE_OWNER	All	Allow	0
	GROUP_A	Write	Allow	0
ACL D (Shared, inherits from ACL C)	GROUP_A	CreateChildren	Allow	0
	ALL	Read	Allow	3
	ROLE_OWNER	All	Allow	1
ACL E (Defining, inherits from ACL B)	GROUP_A	Write	Allow	1
	GROUP_A	CreateChildren	Allow	1
	All	Read	Allow	2
ACL F (Shared, inherits from ACL E)	Andy	All	Allow	0
	Bob	Write	Allow	0
	Bob	WriteContent	Deny	0
ACL G (Defining, no inheritance)	All	Read	Allow	3
	Andy	All	Allow	1
	Bob	Write	Allow	1
	Bob	WriteContent	Deny	1
	Bob	All	Allow	0

ACL format	Authority	Permission	Allow/Deny	Position
ACL H (Shared, inherits from ACL G)	Bob	All	Allow	1

ACL A, and any ACL that inherits from it, allows Read for everyone (All) unless permissions are subsequently denied for everyone (All). If ACL A is changed, all the ACLs that inherit from ACL A in the ACL tree will reflect this change. In the example, nodes 1-12 would be affected by such a change. Nodes 13 and 14 would not inherit the change due to the definition of ACL G.

ACL C adds Contributor and Editor permissions for any authority in GROUP_A.

 The GROUP_ prefix is normally hidden by the administration pages of Alfresco Share.

Anyone in GROUP_A can edit existing content or create new content. The owner ACE means that anyone who creates content then has full rights to it. The ACE assignment for owner is not normally required as all rights are given to node owners in the context-free ACL defined in the default permission configuration.

ACL E adds some specific user ACEs in addition to those defined in ACL A. As an example, it allows Bob Write but also denies WriteContent. Write is made up of WriteContent and WriteProperties. Bob will only be allowed WriteProperties.

ACL G does not inherit and starts a new ACL tree unaffected by any other ACL tree unless an inheritance link is subsequently made.

If a new node was created beneath node 13 or 14 it would inherit ACL H. If a new node was created beneath nodes 1, 6, 7, or 8 it would inherit ACL B.

If a node that has a shared ACL has an ACE set, a new defining ACL and a related shared ACL are inserted in the ACL tree. If a defining ACL has all its position 0 ACEs removed, it still remains a defining ACL: There is no automatic clean up of no-op defining ACLs.

Modifying access control

Modifying access control can involve changing definitions, adding services, defining types and aspects, or adding definitions to new or existing security interceptors.

Main functions include:

- Changing the definition of existing security interceptors to check for different conditions
- Adding new public services and related security interceptors
- Defining new types and aspects and their related permissions
- Adding new definitions to the security interceptor by implementing an ACEGI AccessDecisionVoter and/or AfterInvocationProvider (in extreme cases)

A few constraints and design patterns should be observed when modifying access control. Permissions apply to the node as whole. In particular, the same Read rights apply to all properties and content. You should check that methods can be executed and not that a user has a particular permission. The access control restrictions for a public service method can change. Follow the design pattern to implement RBAC roles.

When modifying access control, do not try to split ReadProperties and ReadContent. This does not make sense for search. A node and all of its properties, including content, are indexed as one entity. Splitting the evaluation of access for content and properties is not possible. Search would have to apply both criteria so as to not leak information. Other services, such as copy, might not behave as expected or might produce nodes in an odd state.

Permissions are assigned at the node level, not at the attribute level. Again, this makes sense with the search capabilities. Search results need to reflect what the user performing the search

can see. It makes sense that all properties have the same Read access as the node, as nodes are indexed for searching and not individual properties. Applying Read ACLs at the property level would require a change to the indexing implementation or a complex post analysis to work out how nodes were found by the search. If not, the values of properties could be deduced by how a readable node was found from a search on restricted properties.

Fine grain attribute permissions could be implemented by using children nodes to partition metadata. Queries would have to be done in parts and joined by hand, as there is no native support for SQL-like join.

Check that method execution is allowed and not that the user has a fixed permission. Rather than checking for Read permission in code, check that the appropriate method can be called using the `PublicServiceAccessService` bean. This avoids hard coding to a specific permission implementation and is essential if you intend to mix records management and the content repository. The access restrictions for public service methods can change. The `PublicServiceAccessService` bean allows you to test if any public service method can be invoked successfully with a given set of arguments. It checks all the entry criteria for the method and, assuming these have not changed, the method can be called successfully. The method call can still fail if the conditions for the returned object are not met or some security configuration has changed, such as an ACE is removed, a user is removed from a group, or the method fails for a non-authorization reason.

For those coming from an RBAC background, Alfresco has roles in the RBAC sense only for limited internal use. To implement RBAC in Alfresco, use zoned groups. These groups will not appear in the administration pages of Alfresco Share as normal groups (unless you also add them to the `APP.DEFAULT` zone) but can be used to assign users and groups to roles. This approach has been taken in Alfresco to support roles in Alfresco Share. To map RBAC terminology to Alfresco: operations map to method calls on public service beans, objects map to method arguments including nodes (folders, documents, and so on). Users and permissions/privileges map directly. Alfresco allows the assignment of permissions to users or groups.

By default, the owner of an object can manage any aspect of its ACL. Users with `ChangePermissions` rights for a node can also change its ACL. If users have the ability to alter the ACL associated with an object, they can allow other users to do the same. There is no restriction on the permissions they can assign. The Alfresco model supports liberal discretionary access control with multi-level grant. A user who can grant access can pass on this right without any restriction. In addition, anyone who can change permissions can carry out the revocation of rights: it is not restricted to the original granter. Normally, when someone can perform an operation you would not expect it is because they own the node and therefore have all permissions for that node.

Public services

Security is enforced around public services. Web services, web scripts, Alfresco Share, CIFS, WebDAV, FTP, CMIS, and more, all use public services, and therefore include security enforcement.

Public services are defined in [public-services-context.xml](#).

Access control allows or prevents users or processes acting on behalf of a user, from executing service methods on a particular object by checking if the current user, or any of the authorities granted to the current user, has a particular permission or permission group, or that the user has a particular authority.

For example, on the `NodeService` bean, the `readProperties` method checks that the current user has Read permission for the node before invoking the method and returning the node's properties. On the `SearchService` query method, the results are restricted to return only the nodes for which a user has Read permission.

Public services configuration

Security is enforced in the Spring configuration by defining proxies for each internal service implementation and adding a method interceptor to enforce security for each public service proxy. These interceptors also have other roles. When a method is called on a public service, the security interceptor is called before the method it wraps. At this stage, the interceptor can examine the function arguments to the method and check that the user has the appropriate rights for each argument in order to invoke the method. For example, a method `delete(NodeRef nodeRef)` exists on the node service. The security interceptor can see the `nodeRef` argument before the underlying `delete(...)` method is called. If configured correctly, the interceptor could check that the current user has "Delete" permission for the node. If they do not have the permission, a security exception is raised. If all the entry criteria are met, the method goes ahead.

In a similar manner, after a method has executed the interceptor can examine the returned object and decide if it should return it to the caller. For example, a search method could return a list of nodes. The security interceptor could filter this list for only those nodes for which the current user has Read permission.

It is also possible to configure a method so that it can be called by all users, only by users with the admin role, or only by specific users or groups. This can also be enforced by the security method interceptor.

Access control interceptor definitions for public services are included in `<installLocation>\tomcat\webapps\alfresco\WEB-INF\classes\alfresco\public-services-security-context.xml` along with any other supporting beans. This configuration file also defines the location from which the permission model is loaded. The interceptors are wired up to the public services in `<installLocation>\tomcat\webapps\alfresco\WEB-INF\classes\alfresco\public-services-context.xml`. The public services are the only Spring beans to have access control.

Method-level security definition

Method access is defined in the normal ACEGI manner with some additions.

The beans required to support Spring ACEGI-based security around method invocation are defined in [public-services-security-context.xml](#). This configures two Alfresco-specific beans: A voter that can authorize method execution based on the permissions granted to the current user for specific arguments to the method, and an after invocation provider to apply security to objects returned by methods.

For the following information detailing preconditions and postconditions, these factors are all relevant:

<authority>

Represents an authority (user name or group).

<#>

Represents a method argument index.

<permission>

Represents the string representation of a permission.

Preconditions take one of the following forms:

ACL_METHOD.<authority>

Restricts access to the method to those with the given authority in Alfresco. This could be a user name, role or group. Dynamic authorities are not supported.

ACL_NODE.<#>.<permission>

Restricts access control to users who have the specified permission for the node at the identified argument. If the argument is a `NodeRef`, it will be used; if it is a `StoreRef`, the root node for the store will be used; if it is a `ChildAssociationRef`, the child node will be used.

ACL_PARENT.<#>.<permission>

Restricts access control to users who have the specified permission for the parent of the node on the identified argument. If the argument is a NodeRef, the parent of the node will be used; if it is a ChildAssociationRef, the parent node will be used.

ROLE

Checks for an authority starting with ROLE_.

GROUP

Checks for an authority starting with GROUP_.

Here are some examples of method level security parameters:

- `ACL_METHOD.ROLE_ADMINISTRATOR`: Executes a method that allows access to users who are members of the administrator group.
- `ACL_ALLOW`: Executes a method that allows access to all users.

If more than one `ACL_NODE.<#>.<permission>`, `ACL_PARENT.<#>.<permission>`, or `ACL_METHOD.<permission>` entry is present, then all of the `ACL_NODE` and `ACL_PARENT` permissions must be present and any one of the `ACL_METHOD` restrictions, if present, for the method to execute.

Post-conditions take the forms:

AFTER_ACL_NODE.<permission>

Similar to `ACL_NODE.<#>.<permission>` but the restriction applies to the return argument.

AFTER_ACL_PARENT.<permission>

Similar to `ACL_PARENT.<#>.<permission>` but the restriction applies to the return argument.

The support return types are:

- StoreRef
- ChildAssociationRef
- Collections of StoreRef, NodeRef, ChildAssociationRef, and FileInfo
- FileInfo
- NodeRef
- Arrays of StoreRef, NodeRef, ChildAssociationRef, and FileInfo
- PagingLuceneResultSet
- QueryEngineResults
- ResultSet

The post-conditions will create access denied exceptions for return types such as NodeRef, StoreRef, ChildAssociationRef, and FileInfo. For collections, arrays, and result sets, their members will be filtered based on the access conditions applied to each member.

Continuing the example from the permissions defined for the Ownable aspect, the definition for the security interceptor for the related OwnableService is shown in the following code snippet.

```
<bean id="OwnableService_security"
  class="org.alfresco.repo.security.permissions.impl.acegi.MethodSecurityInterceptor">
  <property name="authenticationManager"><ref bean="authenticationManager"/></property>
  <property name="accessDecisionManager"><ref local="accessDecisionManager"/></property>
  <property name="afterInvocationManager"><ref local="afterInvocationManager"/></property>
  <property name="objectDefinitionSource">
    <value>
```

```

org.alfresco.service.cmr.security.OwnableService.getOwner=ACL_NODE.0.sys:base.ReadProp
org.alfresco.service.cmr.security.OwnableService.setOwner=ACL_NODE.0.cm:ownable.SetOwn
org.alfresco.service.cmr.security.OwnableService.takeOwnership=ACL_NODE.0.cm:ownable.T
org.alfresco.service.cmr.security.OwnableService.hasOwner=ACL_NODE.0.sys:base.ReadProp
    org.alfresco.service.cmr.security.OwnableService.*=ACL_DENY
    </value>
  </property>
</bean>

```

Security for the four methods on the `OwnableService` is defined. To invoke the `OwnableService` `getOwner()` method on a node, the invoker must have permission to read the properties of the target node. To set the owner of a node, a user must have been explicitly assigned the `SetOwner` permission or have all rights to the node. A user can have all rights to a node by using the context-free ACL or be assigned a permission, which grants all permission or includes `SetOwner`. With the default configuration, a user will own any node they create and therefore be able to give ownership to anyone else and possibly not have the right to take ownership back.

The last entry catches and denies access for any other method calls other than those listed. If any additional methods were added to this service and no security configuration explicitly defined for the new methods, these methods would always deny access.

Implementation and services

Alfresco enforces security services for managing authentication information. This section provides detailed information about the security services used in Alfresco and their implementation.

The following key services are involved in access control:

- `AuthenticationService`: responsible for authenticating user name and password.
- `PersonService`: responsible for obtaining a reference to the `Person` node for a given user name. It also creates, deletes and updates personal information.
- `AuthorityService`: responsible for managing authorities.
- `PermissionService`: responsible for managing ACLs and ACEs, and for checking if a user has been assigned a permission for a particular node.
- `OwnableService`: manages object ownership and is used in evaluation the dynamic `ROLE_OWNER` authority.

Let's consider a possible scenario to understand how the security services work. A user logs in to Alfresco using the *authentication service*, which determines the user's authorities, such as their user name (which is a `USER` authority). The *authority service* adds and manages the relevant groups and roles. The *permission service* maps those users, groups and roles to operations on particular nodes. It also controls the inheritance of permissions and provides a common set of default permissions. The *owner service* is related to the special `OWNER` role and it determines the owner of a node. The *person service* deals with the special case of person nodes, which identify users in Alfresco.

The protection of public services methods is implemented using the Spring method interceptors defined as part of the related ACEGI 0.8.2 security package. The Alfresco implementation adds new implementations of the ACEGI interfaces `AccessDecisionVoter` and `AfterInvocationProvider`, which support the configuration elements that have already been described (for example, `ACL_NODE.<#>.<permission>`). These extension classes make use of the key services.

Authentication service

This topic describes the features of authentication service and how to configure it.

The authentication service provides an API for:

- Authenticating using a user name and password
- Authenticating using a ticket
- Creating, updating and deleting authentication information
- Clearing the current authentication
- Invalidating a ticket
- Getting the user name for currently authenticated users
- Getting a ticket for subsequent re-authentication

The authenticated user name is used as the key to obtain other security information, such as group membership, the details about the person or to record a user as the owner of an object. It is one of the identifiers against which permissions can be assigned.

The authentication service does not provide any details about a user other than authentication. It stores authentication information on the calling thread. Application developers should ensure that this information is cleared.

The authentication service brings together three components:

- authentication component, which supports authentication;
- authentication DAO, which provides an API to create, delete and update authentication information; and
- ticket component, which manages and stores tickets that can be obtained after authentication and used in place of authentication.

The implementation and configuration for this service can be found in the `authentication-services-context.xml` file. This default implementation coordinates two service providers for `AuthenticationComponent` and `MutableAuthenticationDAO`. It also uses the permission service provider interface to clear up permissions as users are deleted. Tickets are supported using the ticket component.

Person service

This topic describes the features of person service and how to configure it.

The `PersonService` interface is the API by which nodes of the person type, as defined in [contentModel.xml](#), should be accessed.

The `PersonService` is responsible for all of the following:

- Obtaining a reference to the Person node for a given user name
- Determining if a person entry exists for a user
- Potentially creating missing people entries with default settings on demand
- Supplying a list of mutable properties for each person
- Creating, deleting, and altering personal information

The beans to support the `PersonService` and its configuration can be found in [authentication-services-context.xml](#). The principle configuration options are around how people are created on demand if users are managed by using NTLM or some other external user repository.

Authority service

This topic describes the features of authority service. It also describes how to configure it using the `authority-services-context.xml` file.

The authority service is responsible for:

- Creating and deleting authorities

- Querying for authorities
- Structuring authorities into hierarchies
- Supporting queries for membership
- Finding all the authorities that apply to the current authenticated user
- Determining if the current authenticated user has admin rights
- Managing zones and the assignment of authorities to zones

The default implementation allows a list of group names to define both administration groups and guest groups. Each authentication component defines its own default administrative user(s), which can also be set explicitly. The default service is defined in the [authority-services-context.xml](#) file.

Using `guestGroups` and `adminGroups` properties

The `authority-services-context.xml`, bean id `authorityService` provides the property configuration of the Authority Service implementation. This configuration also allows the designation of specific groups with `admin` or `guest` permissions in the system.

By listing a group under the `guestGroups` property (case insensitive), the users in that group will only be allowed `guest` permission. Likewise, by listing a group under the `adminGroups` property (case insensitive), the users in that group will be provided `admin` permission.

For example, assume that you are synchronizing users into Alfresco and you specifically want to specify some groups as only guest users in the system. You would override the `authority-services-context.xml` file adding those groups to the `guestGroups` list (case insensitive). As a result, users in those groups will have authenticated logins but limited to guest authorization. For details, see [Configuring `guestGroups` and `adminGroups` properties](#).

Configuring `guestGroups` and `adminGroups` properties

This topic describes how to configure the `guestGroups` and `adminGroups` properties.

1. Download the [authority-services-context.xml](#) file:
2. Paste this file into the `<extension>` directory.
3. Open the `authority-services-context.xml` file.
 - a. To specify some groups as only guest users in Alfresco, add them to the `guestGroups` property list.

```
<!-- A list of groups with guest rights. -->
<!-- -->
    <property name="guestGroups">
      <set>
      </set>
    </property>
```

- b. To assign admin rights to some groups in Alfresco, add them to the `adminGroups` property list.

```
<!-- A list of groups with admin rights. -->
<!-- -->
    <property name="adminGroups">
      <set>
        <value>ALFRESCO_ADMINISTRATORS</value>
      </set>
    </property>
```

4. Save the file and then restart the Alfresco server.

Permission service

This topic describes the features of permission service and how to configure it.

The permission service is responsible for:

- Providing well known permissions and authorities
- Providing an API to read, set, and delete permissions for a node
- Providing an API to query, enable, and disable permission inheritance for a node
- Determining if the current, authenticated user has a permission for a node

The `PermissionService` interface defines constants for well-known permissions and authorities.

The default implementation coordinates implementations of two service provider interfaces: a `ModelDAO` and a `PermissionsDAO`. A permission is simply a name scoped by the fully qualified name of the type or aspect to which it applies. The beans are defined and configured in [public-services-security-context.xml](#). This file also contains the configuration for security enforcement.

The `ModelDAO` interface defines an API to access a permissions model. The default permission model is in XML and defines permission sets, and their related permission groups and permissions. Global permissions are part of the permission model. There can be more than one permission model defined in XML; they are in practice merged into one permission model. A module can extend the permission model.

The available permissions are defined in the permission model. This is defined in [permissionDefinitions.xml](#). This configuration is loaded in a bean definition in [public-services-security-context.xml](#). This file also defines global permissions. The definition file is read once at application start-up. If you make changes to this file, you will have to restart the repository in order to apply the changes.

Ownable service

This topic describes the features of ownable service and how to configure it.

The idea of file ownership is present in both UNIX and Windows. In Alfresco, the repository has the concept of node ownership. This ownership is optional and is implemented as an aspect.

The owner of a node can have specific ACLs granted to them. Ownership is implemented as the dynamic authority, `ROLE_OWNER`, and is evaluated in the context of each node for which an authorization request is made. The Ownable aspect, if present, defines a node's owner by storing a `userName`; if the Ownable aspect is not present, the creator is used as the default owner. If the `userName` of the current user matches, including case, the `userName` stored as the owner of the node, the current user will be granted all permissions assigned to the authority `ROLE_OWNER`.

The `OwnableService` is responsible for all of the following:

- Determining the owner of a node
- Setting the owner of a node
- Determining if a node has an owner
- Allowing the current user to take ownership of a node

The `OwnableService` is supported by an Ownable aspect defined in `<installLocation>\tomcat\webapps\alfresco\WEB-INF\classes\alfresco\model\contentModel.xml`.

There are permissions and permission groups associated with the Ownable aspect in the permission model and related access controls applied to the methods on the public `OwnableService`.

Admin password in default authentication

This topic explains how the password for the Admin user is used by the default authentication system.

The Admin password for default authentication is set as a part of the initial bootstrap. This is located in `config\alfresco\bootstrap\alfrescoUserStore.xml`. The password is MD4 encoded, as required by NTLM.

How to reset the Admin password

If you happen to lose or forget the password for the Admin user, you can reset the password within the database using any one of the following ways:

- If you know the password of at least one user, then:
 1. Assign Admin rights to this known user by adding the following line in the `alfresco-global.properties` file.


```
alfresco_user_store.adminusername=username
```

where, `username` is the user name of the user whose password is known.
 2. Restart the repository.
 3. Log in as the know user.
 4. Reset the Admin user's password.
 5. Reset the configuration.
- You can reset the Admin password without knowing any user password by following the steps below:
 1. Configure the authentication component to accept all logins using `org.alfresco.repo.security.authentication.SimpleAcceptOrRejectAllAuthenticationCom`
 2. Login as a user with Admin rights.
 3. Reset the Admin user's password.
 4. Revert the configuration.
- You can also change the password directly in the database with the following instructions (for version 3.1 or later):
 1. Run the following command to find out the identifying parameters for how the Admin password is stored. Check that you have only one row in the output.

```
SELECT anp1.node_id,
       anp1.qname_id,
       anp1.string_value
FROM alf_node_properties anp1
     INNER JOIN alf_qname aq1 ON aq1.id = anp1.qname_id
     INNER JOIN alf_node_properties anp2 ON anp2.node_id = anp1.node_id
     INNER JOIN alf_qname aq2 ON aq2.id = anp2.qname_id
WHERE aq1.local_name = 'password'
AND aq2.local_name = 'username'
AND anp2.string_value = 'admin'
```

The output shows the current MD4 hashed password for the Admin user. Here's an example output:

```
+-----+-----+-----+
| node_id | qname_id | string_value |
+-----+-----+-----+
| 4 | 10 | 209c6174da490caeb422f3fa5a7ae634 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

2. To update the password, use the following command:

```
UPDATE alf_node_properties
SET string_value='209c6174da490caeb422f3fa5a7ae634'
WHERE
node_id=THENODEIDABOVE
and
```

```
qname_id=THEQNAMEVALUEABOVE
```

where you need to replace `THENODEIDABOVE` and `THEQNAMEVALUEABOVE` with the result values of `node_id` and `qname_id`, obtained in the previous step. In this example, it is 4 and 10, respectively.

-  Ensure that you use appropriate `AND` conditions in the `UPDATE` query.
-  The hashed password you use in the `UPDATE` statement must be in lower case. If you use a hash tool that returns a string with uppercase letters, change them all to lowercase.

3. Restart Alfresco.

Setting up clustering

This section describes how to implement multiple Alfresco instances in a clustered environment.

A cluster represents a collection of nodes. Clustering is implemented in Alfresco to provide high scalability and resilience. Improved performance is enhanced through redundant nodes that provide services when other nodes fail. When integrated with a load balancer, performance is enhanced by distributing, or balancing, server workload across a collection of nodes.

Clustering prerequisites when upgrading to Alfresco One 5.0

There are a number of prerequisites for upgrading from a version of Alfresco prior to Alfresco One 4.2 to Alfresco One 5.0 in a clustered environment.

Before upgrading, ensure that all files and configuration are backed up. Any customization(s) that you have made, for example, creation of custom caches, might need to be reapplied using the new Alfresco One 5.0 clustering infrastructure.

The following libraries are no longer used in Alfresco One 4.2 onwards, so any configuration related to these libraries should be removed before upgrading:

- JGroups
- EHCACHE

 You do not need to follow these steps if you are upgrading from Alfresco One 4.2 to Alfresco One 5.0. This information is only relevant if you are upgrading from any version of Alfresco prior to Alfresco One 4.2.

Follow the steps to remove the configuration not supported in version 5.0:

1. Browse to the `<classpathRoot>` directory.
For example, for Tomcat 6, browse to the `$TOMCAT_HOME/shared/classes/alfresco/extension/` directory.
2. Delete the `ehcache-custom.xml` file.
3. Browse to the `<classpathRoot>` directory.
For example, for Tomcat 6, browse to the `$TOMCAT_HOME/shared/classes/` directory.
4. Open the `alfresco-global.properties` file.
5. Remove the following legacy properties from the `alfresco-global.properties` file:
 - `alfresco.cluster.name`
 - `alfresco.ehcache.rmi.hostname`
 - `alfresco.ehcache.rmi.port`
 - `alfresco.ehcache.rmi.remoteObjectPort`

- alfresco.jgroups.defaultProtocol
 - alfresco.jgroups.bind_address
 - alfresco.jgroups.bind_interface
 - alfresco.tcp.start_port
 - alfresco.tcp.initial_hosts
 - alfresco.tcp.port_range
 - alfresco.udp.mcast_addr
 - alfresco.udp.mcast_port
 - alfresco.udp.ip_ttl
 - filesystem.cluster.enabled
 - filesystem.cluster.configFile
6. Browse to the `<classpathRoot>` directory.
For example, for Tomcat 6, browse to the `$TOMCAT_HOME/shared/classes/alfresco/extension` directory.
 7. Remove the Hazelcast configuration file, `hazelcastConfig.xml`, as a centralised configuration is now included within the `alfresco.war` deployment archive.
The `filesystem.cluster.configFile` property mentioned in Step 5 refers to the `hazelcastConfig.xml` file.
 8. After you have performed all the specified steps, if you want to initiate clustering, see [Setting up repository server cluster](#) for the instructions on installing an Alfresco 5.0 cluster.

Setting up Share cluster

This section provides information about cluster configuration for Share.

Configuring Hazelcast between Share instances

This section describes the configuration of Hazelcast clustering between instances of Share.

In a load balanced environment, Alfresco Share now uses Hazelcast to provide multicast messaging between the web-tier nodes. As a result, Share caches no longer need to be disabled for any node, simple cache invalidation message are sent to all nodes when appropriate. Each node functions practically as fast as a single Share instance, enhancing the overall performance of Share.

To enable Hazelcast clustering between Share instances, configure the `custom-slingshot-application-context.xml` file found at `< TOMCAT-HOME>/shared/classes/alfresco/web-extension`. This file is used to override the Spring application context beans for Share.

 An example `custom-slingshot-application-context.xml.sample` file is provided in the Alfresco distribution, which now includes this configuration.

To enable the Hazelcast cluster messaging, edit this section on each Share Tomcat instance:

```
<!--
    Hazelcast distributed messaging configuration - Share web-tier cluster
    config (3.4.8 and 4.0.1)
    - see http://www.hazelcast.com/docs.jsp
    - and specifically http://www.hazelcast.com/docs/1.9.4/manual/single_html/
    #SpringIntegration
-->
<!-- Configure cluster to use either Multicast or direct TCP-IP messaging -
    multicast is default -->
<!-- Optionally specify network interfaces - server machines likely to have
    more than one interface -->
```

```

<!-- The messaging topic - the "name" is also used by the persister config
below -->
<hz:topic id="topic" instance-ref="webframework.cluster.slingshot"
name="slingshot-topic"/>
<hz:hazelcast id="webframework.cluster.slingshot">
  <hz:config>
    <hz:group name="slingshot" password="alfresco"/>
    <hz:network port="5801" port-auto-increment="true">
      <hz:join>
        <hz:multicast enabled="true"
          multicast-group="224.2.2.5"
          multicast-port="54327"/>
        <hz:tcp-ip enabled="false">
          <hz:members></hz:members>
        </hz:tcp-ip>
      </hz:join>
      <hz:interfaces enabled="false">
        <hz:interface>192.168.1.*</hz:interface>
      </hz:interfaces>
    </hz:network>
  </hz:config>
</hz:hazelcast>

<bean id="webframework.slingshot.persister.remote"
class="org.alfresco.web.site.ClusterAwarePathStoreObjectPersister"
parent="webframework.sitedata.persister.abstract">
  <property name="store" ref="webframework.webapp.store.remote" />
  <property name="pathPrefix"><value>alfresco/site-data/${objectTypeIds}</
value></property>
  <property name="hazelcastInstance" ref="webframework.cluster.slingshot" />
  <property name="hazelcastTopicName"><value>slingshot-topic</value></
property>
</bean>

<bean id="webframework.factory.requestcontext.servlet"
class="org.alfresco.web.site.ClusterAwareRequestContextFactory"
parent="webframework.factory.base">
  <property name="linkBuilderFactory"
ref="webframework.factory.linkbuilder.servlet" />
  <property name="extensibilityModuleHandler"
ref="webscripts.extensibility.handler" />
  <property name="dependencyHandler" ref="dependency.handler" />
  <property name="clusterObjectPersister"
ref="webframework.slingshot.persister.remote" />
</bean>

```

This configuration enables the Hazelcast Spring integration, which in turn, starts the Hazelcast server. The Hazelcast server is easily configurable and can use either multicast (default) or TCP-IP direct, if preferred. For more information, see the [Hazelcast Documentation](#).

If this configuration is enabled, the Share instance becomes a cluster node and Hazelcast is started. If this configuration is disabled (such as, for a default install), then Hazelcast is not started. While using Share, only when any of the following actions occur, the cache invalidation messages will be sent from the affected node to other nodes in the cluster:

- an existing site/user dashboard layout is modified
- new site or user dashboard is created
- runtime application properties are changed



For activating the default set up, apply identical configuration to each Share node.

The following is a sample output that you get when you start Share:

```

INFO: /192.168.2.8:5801 [slingshot] Hazelcast 1.9.4.6 (20120105) starting at
Address[192.168.2.8:5801]
19-Jan-2012 13:58:57 com.hazelcast.system

```

```

INFO: /192.168.2.8:5801 [slingshot] Copyright (C) 2008-2011 Hazelcast.com
19-Jan-2012 13:58:57 com.hazelcast.impl.LifecycleServiceImpl
INFO: /192.168.2.8:5801 [slingshot] Address[192.168.2.8:5801] is STARTING
19-Jan-2012 13:58:59 com.hazelcast.impl.MulticastJoiner
INFO: /192.168.2.8:5801 [slingshot]
Members [1] {
    Member [192.168.2.8:5801] this
}
19-Jan-2012 13:58:59 com.hazelcast.impl.management.ManagementCenterService
INFO: /192.168.2.8:5801 [slingshot] Hazelcast Management Center started at port
5901.
19-Jan-2012 13:58:59 com.hazelcast.impl.LifecycleServiceImpl
INFO: /192.168.2.8:5801 [slingshot] Address[192.168.2.8:5801] is STARTED

```

The message shows that the configuration has successfully initialized Hazelcast between Share instances.

Configuring Share clustering

These steps are required for cluster configuration for Share. If you are using an HTTP load-balancing mechanism in front of a clustered installation, 'sticky' routing must be enabled for the HTTP requests made by the Share tier to the repository tier (the `/alfresco` application).

This can be achieved in one of two ways:

1. Hard-wire each `/share` instance to its own `/alfresco` instance, bypassing the load balancer.
This can be achieved by populating each `share-config-custom.xml` file with a host name and port number that is not behind your load balancing mechanism.
2. If NTLM or Kerberos authentication is enabled with SSO, then Share will use cookie-based sessions and you can configure your load balancer to use sticky routing using the `JSESSIONID` cookie.

To enable NTLM or Kerberos with SSO, refer to the instructions in [Configuring authentication](#) to configure `alfrescoNtlm`, `passthru`, or Kerberos authentication, and set either `ntlm.authentication.sso.enabled=true` or `kerberos.authentication.sso.enabled=true`).



If you are configuring a cluster, refer to [Setting up high availability systems](#).

Recommendations for split architecture

This topic describes the recommendations made for splitting the Alfresco architecture in a distributed or clustered environment.

Generally, there are two complementary purposes for distributing or clustering your installation.

- To achieve redundancy or high availability
- To provide high-performance and/or throughput

The main decisions are involved around when to split and how to split.

When to split

This topic provides indicators to help you decide when to split your architecture from a single node environment to a distributed node environment.

Some of the indicators to look for include:

- Low disk space
- CPU running out of memory
- High indexing load

How to split

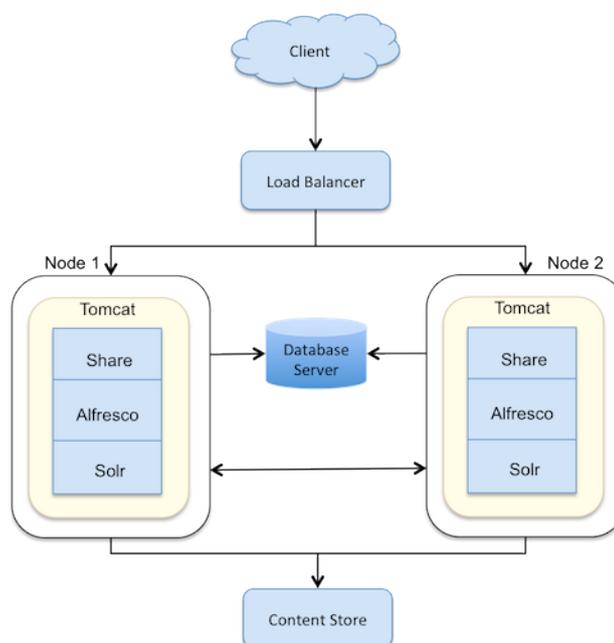
Once you have decided to upgrade from a single node environment to a distributed or clustered environment, you must find the most appropriate way to cluster Alfresco's architecture. This topic describes the clustered architecture based on scenarios.

Scenario: Clustering for redundancy

This is a scenario-based topic describing the clustering architecture for redundancy and high availability of Alfresco services.

This scenario shows a single repository database and content store, and two Tomcat nodes/web servers on two separate machines accessing the content simultaneously. The configuration does not guard against the content store or database failure, but allows multiple servers to share the web load, and provides redundancy in case of a server failure. Each server has local indexes (in the local content store).

This is the simplest cluster to set up and is ideal for small-scale installations. A cluster consisting of two or more machines working together provides a higher level of availability, reliability, and scalability than can be obtained from a single node.



A hardware load balancer balances the web requests among multiple servers. The load balancer must support 'sticky' sessions so that each client always connects to the same server during the session. The content store and database will reside on separate servers, which allows us to use alternative means for content store and database replication.

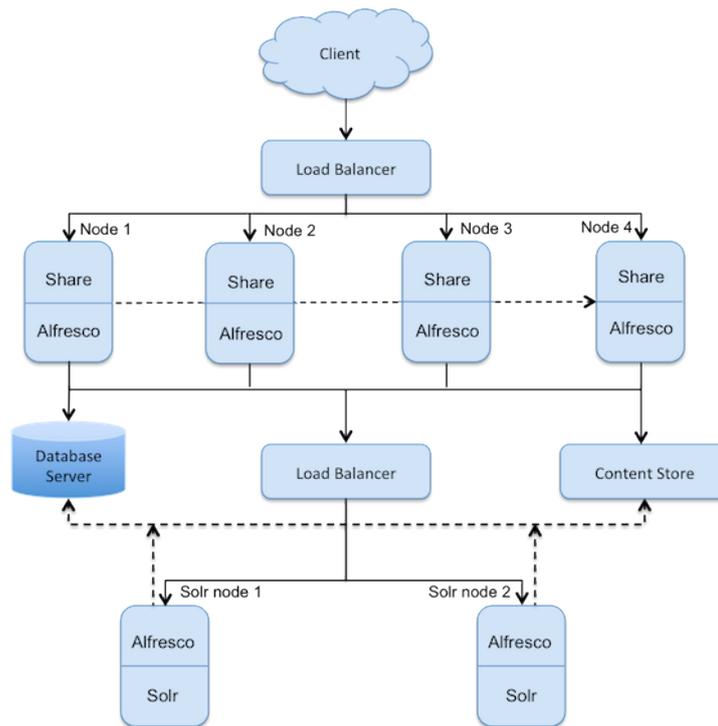
 All the servers in a cluster should have static IP addresses assigned to them.

Scenario: Clustering for high throughput

This is a scenario-based topic describing the clustering architecture for maximizing throughput of Alfresco services.

This scenario shows a single repository database and content store. There are four nodes with Alfresco/Share and two nodes with Solr search subsystem, all accessing the content simultaneously. This set up provides a higher level of availability, reliability, and scalability, thereby maximizing the throughput of various Alfresco services. Nodes in a cluster are positioned

behind a load balancer that delegates requests to cluster members based on any one member's ability/availability to handle the load.



Each Alfresco/Share instance is deployed into its own Tomcat servlet container. Alfresco services and CPU runtime footprint are optimized for high throughput under heavy concurrency with such a deployment. The load balancer fronts the cluster, and directs traffic to the member of the cluster most able to handle the current request.

 All the servers in a cluster should have static IP addresses assigned to them.

Setting up repository server cluster

This section describes how to implement an Alfresco One 5.0 repository server cluster.

The repository server cluster consists of the following components:

- Database server
- Content store, for example, NFS server
- Solr server
- Load balancer
- Hazelcast mancenter server (optional)

Setting up repository server cluster

This topic describes the steps for setting up a repository cluster.

By default, all Enterprise servers connected to the same database will form a repository cluster.

Follow these steps for each server in the cluster:

1. Install and configure repository server. See [Installing the Alfresco WAR](#) for deploying Alfresco (`alfresco.war`). In addition, ensure that:
 - The content store is available to all members in the cluster. For example, an NFS server mounted locally can be referred to by the `dir.root` property.

- Each cluster member must be set up to access the same database, using the same database properties as in `alfresco-global.properties`.
 - Deploy a Solr indexing server for use across the cluster and configure the properties of each cluster member to utilize this Solr server.
2. Ensure that port number 5701 (the default clustering port) is accessible on each repository server by all the other repository servers in the cluster.
 3. Specify a wildcard (for example, `10.50.*.*`) or exact (for example, `192.168.1.100`) IP address of the network interface for clustering to use.

The advantage of using a wildcard IP address is that the configuration can be used on multiple servers without local changes. The java property name to use is `alfresco.cluster.interface` (optional).

4. Set the following java property to activate Hazelcast's own JMX reporting (optional).

```
hazelcast.jmx=true
```

5. For security reasons, set the cluster password with the following java property:

```
alfresco.hazelcast.password
```

Starting the repository server cluster

This topic describes the process of starting the repository server cluster.

In most cases, it is not necessary to apply any clustering-specific configuration - just starting the servers will result in a cluster.

Let's suppose you have two cluster members on IP addresses, 10.244.50.101 and 10.244.50.102. Upon starting the first member, you should see the log message similar to the one shown:

```
2013-08-05 17:06:31,794 INFO [cluster.core.ClusteringBootstrap] [Thread-3]
Cluster started,
    name: MainRepository-2c0aa5c6-e38a-4f64-bd29-1a7cf9894350
2013-08-05 17:06:31,797 INFO [cluster.core.ClusteringBootstrap] [Thread-3]
Current cluster members:
10.244.50.101:5701 (hostname: repo1.local)
```

This shows that a cluster name has been automatically generated, based on the repository name (`MainRepository`) and a UUID (a random/ unique identifier). Finally, the cluster has been started and the cluster members are listed. As shown in the log message, only one cluster member is present currently.

Upon starting the second member, you should see the log message similar to the one shown:

```
2013-08-05 17:06:58,350 INFO [cluster.core.ClusteringBootstrap] [Thread-3]
Cluster started,
    name: MainRepository-2c0aa5c6-e38a-4f64-bd29-1a7cf9894350
2013-08-05 17:06:58,353 INFO [cluster.core.ClusteringBootstrap] [Thread-3]
Current cluster members:
10.244.50.102:5701 (hostname: repo2.local)
10.244.50.101:5701 (hostname: repo1.local)
```

This log message shows that both the servers are now members of the same cluster.

Testing the cluster

The quickest and easiest way to test the cluster is by using the Admin Console.

Ensure that the Alfresco server is running.

1. Enter the following URL in a browser window:

```
http://<your-host-name>:8080/alfresco/service/enterprise/admin
```

Where `<your-host-name>` is the host name where you are running the Alfresco server.

An **Authentication Required** prompt displays, showing the IP address or name and the port number of the Alfresco server.

2. Enter your Alfresco user name and password.

Your user name and password must be for an account with administrator permissions.

The Admin Console displays in a browser window. The first page you see is the [System Summary](#).

3. In the **Repository Services** section, click **Repository Server Clustering**.

You see the **Repository Server Clustering** page.

This page displays information regarding the current cluster members under the **Cluster Members** section.

4. Click **Validate Cluster** to start a quick test to check that communication is available between each pair of cluster members.

You see the **Cluster Validation** page. This page displays the result in a matrix form showing cluster communication as either Success or Failure.

Clustering properties

This information describes the most common clustering-related properties.

 These properties are optional.

Clustering property	Example setting	What is it?
alfresco.cluster.enabled	true	This enables clustering.
alfresco.cluster.interface	10.256.*.*	This specifies a particular network interface to use for clustering. It might be a wildcard value, such as 10.256.*.*, which means an attempt is made to bind with an interface having an IP address beginning with 10.256.
alfresco.cluster.nodetype	Repository Server	This specifies the human-friendly description of the cluster member. It is useful to give a name to the non-clustered servers, such as a transformation server that it attached to the same database as the cluster, but not participating in it (for example, <code>alfresco.cluster.enabled=false</code>).
alfresco.hazelcast.password	alfrescocluster	This specifies the password used by the cluster members to access or join the Hazelcast cluster.
alfresco.hazelcast.port	5701	This specifies the port to use for clustering.
alfresco.hazelcast.autoinc.port	false	This enables Hazelcast to make several attempts to find a free port starting at the value of <code>alfresco.hazelcast.port</code> .  Alfresco recommends that you do not use this property.
alfresco.hazelcast.mancenter.enabled	enabled	This enables the server to push stats and other useful information to Hazelcast's mancenter (management center) dashboard application. See Setting up Hazelcast dashboard .

Clustering property	Example setting	What is it?
<code>alfresco.hazelcast.mancenter.url</code>	<code>http://localhost:8080/mancenter</code>	This specifies the URL where the Hazelcast mancenter application can be found. Note that <code>alfresco.hazelcast.mancenter.enabled</code> must be set to <code>true</code> for this property to be valid.
<code>alfresco.hazelcast.max.no.heartbeat.seconds</code>	<code>15</code>	This specifies the maximum timeout of heartbeat in seconds for a node to assume it is dead.

Setting up Hazelcast dashboard (mancenter)

The Hazelcast management center (mancenter) enables you to monitor and manage your servers running hazelcast. Additionally, mancenter enables you to monitor the overall state of your clusters, and analyze and browse your data structures in detail. This topic describes the instructions for setting up a Hazelcast dashboard (mancenter).

The Hazelcast diagnostics and reporting application is a useful addition to an Alfresco repository cluster. It can be installed on any servlet container.

1. Install a servlet container, for example Tomcat. See [Installing Tomcat application server](#) for more information.
2. Deploy the `mancenter.war` file to the servlet container.
3. Specify the location of the data directory by setting the java property, `hazelcast.mancenter.home`. To do so, add the following property to `CATALINA_OPTS` environment variable.

```
-Dhazelcast.mancenter.home=/home/tomcat7/mancenter_data
```

The data directory where the servlet container is running must be writeable by the user.

4. Set the repository property to enable mancenter use.

```
alfresco.hazelcast.mancenter.enabled=true
```
5. Set the repository property, `alfresco.hazelcast.mancenter.url` to point to the mancenter web application.

```
alfresco.hazelcast.mancenter.url=http://mancenter.example.com:8080/mancenter
```
6. Ensure that the repository servers are able to access the mancenter server at the URL specified in Step 5. The cluster members will push any cluster information updates to this URL. Remember to configure appropriate firewall rules.

Tracking clustering issues

This section describes how to track clustering issues in Alfresco One 5.0.

- The main clustering debug information can be customised using the following log4j setting (default value is `INFO`):

```
log4j.logger.org.alfresco.enterprise.repo.cluster=info
```

- For a better control and more detailed clustering debug information, the following category can be configured:

```
org.alfresco.enterprise.repo.cluster.core.ClusteringBootstrap
```

This controls clustering initialisation and shutdown. It provides `INFO` level startup and shutdown messages. It also provides `WARN` level messages, if clustering is disabled or an invalid 5.0 license is installed.

Here is an example output:

```
12:38:38,769
INFO [org.alfresco.enterprise.repo.cluster.core.ClusteringBootstrap] Cluster
started, name:
    MainRepository-35ee3b27-0276-4224-9613-3fd8089c6e11
12:38:38,776
INFO [org.alfresco.enterprise.repo.cluster.core.ClusteringBootstrap] Current
cluster
members:
    10.248.10.205:5701 (hostname: node1.alf.example.com)
    10.208.63.40:5701 (hostname: node2.alf.example.com)
```

- When a cluster member leaves or joins, the following class generates an informative INFO level message:

```
org.alfresco.enterprise.repo.cluster.core.MembershipChangeLogger
```

Here is an example output:

```
12:38:47,560
INFO [org.alfresco.enterprise.repo.cluster.core.MembershipChangeLogger] Member
joined:
    10.65.41.64:5701 (hostname: node1.alf.example.com)
12:38:47,569
INFO [org.alfresco.enterprise.repo.cluster.core.MembershipChangeLogger]
Current cluster
members:
    10.208.63.40:5701 (hostname: solr.alf.example.com)
    10.248.10.205:5701 (hostname: node2.alf.example.com)
    10.65.41.64:5701 (hostname: node1.alf.example.com)
```

- An important aspect of clustering is caching. To log cache creation (for example, increase the cache related logging to DEBUG level), enable the following log categories:

```
log4j.logger.org.alfresco.enterprise.repo.cluster.cache=DEBUG
log4j.logger.org.alfresco.repo.cache=DEBUG
```

- The underlying clustering technology, Hazelcast, is configured in Alfresco to use log4j for logging. Therefore, you can configure logging for the whole Hazelcast top-level package, as shown:

```
log4j.logger.com.hazelcast=info
```

To increase logging from Hazelcast's member joining mechanism, enable the following log category:

```
log4j.logger.com.hazelcast.impl.TcpIpJoiner=debug
```

Configuring search

From Alfresco One 5.0, Solr 4 is the default search subsystem. This section provides an overview on the Solr 4 search service and describes how to configure it.



In this documentation, we refer to **Solr 1.4** search subsystem as **Solr**.



The Lucene search subsystem is not supported in Alfresco One 5.0.

Configuring search in Alfresco Share

The following sections describe how to configure search in Alfresco Share.

Controlling permissions checking on search results in Share

You can limit time that Alfresco spends on ensuring that the user executing the search has the necessary permissions to see each result. Setting this limit increases search speed and reduces the use of resources.

You can limit both the time spent and the number of documents checked before Alfresco returns a search query using the `system.acl.maxPermissionCheckTimeMillis` and the `system.acl.maxPermissionChecks` properties. The default values are 10000 and 1000 respectively.

1. Open the `<classpathRoot>/alfresco-global.properties` file.
2. Set the `system.acl.maxPermissionCheckTimeMillis` property.
For example, `system.acl.maxPermissionCheckTimeMillis=20000`.
3. Set the `system.acl.maxPermissionChecks` property.
For example, `system.acl.maxPermissionChecks=2000`.



- If you increase these values and have a query that returns a very large number of results, (a) the search results will take longer to be returned to the user, and (b) the system will spend longer to check permissions, leading to the possibility of performance degradation.
- If you set these values to a low number, you run the risk of inconsistent search results every time you run the same search.

Controlling search results in Share

This topic provides instructions on controlling the maximum number of items that a Share search returns.

By default, the Share search feature returns a maximum of 250 search results. You can extend this number of search results to return more than 250 entries.

1. Download the [share-config.xml](#) file.
2. Open the `share-config.xml` file and copy the `<config evaluator="string-compare" condition="Search">` section.
3. Open the `<web-extension>\share-config-custom.xml` file and then paste the copied section.
4. Locate the `<max-search-results>250</max-search-results>` property and then edit the value to your preferred number of search results.
5. For the changes to take effect, refresh the Alfresco web scripts. To refresh the web scripts:
 - a. Navigate to the Alfresco web scripts Home page.
For example, go to: `http://<your-host>:8080/share/page/index`.
 - b. Click on **Refresh Web Scripts**.
You have now refreshed the web scripts and set a limit to the number of items a search in Share returns.



Custom searches and searches from the node browser use the `solr.query.maximumResultsFromUnlimitedQuery` property to control search results. For more information, see [Solr core configuration properties](#).

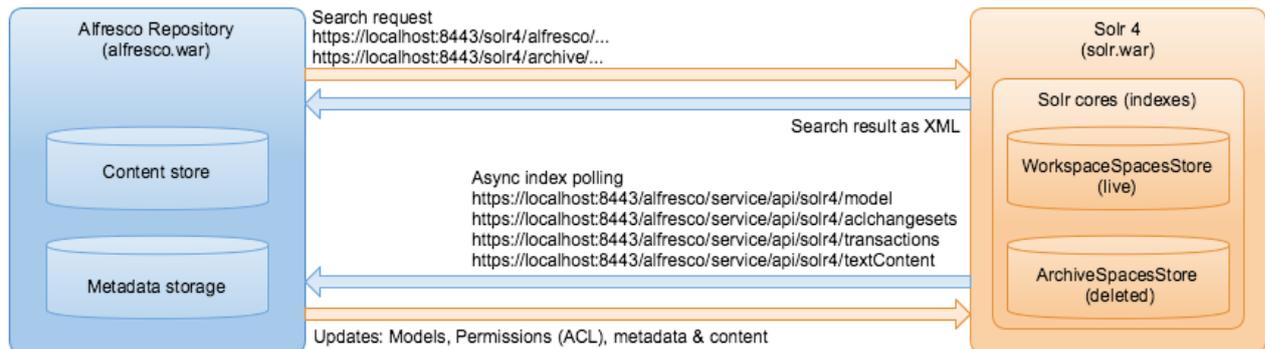
Solr overview

Alfresco supports use of the Solr 4 search platform for searching within the Alfresco repository.

Solr is an open source enterprise search platform that uses Lucene as an indexing engine. Solr is written in Java and runs as a standalone search server within the Tomcat application server. Alfresco sends HTTP and XML input to Solr and searches for content. Solr updates the cores or indexes and returns the result of the query in XML or JSON format.

There are two cores or indexes in Solr version 4:

- **WorkspaceStore:** used for searching all live content stored at `alfresco/solr4` within the Solr 4 search server.
- **ArchiveStore:** used for searching content that has been marked as deleted at `alfresco/solr4` within the Solr 4 search server.



 Solr 4 is the default search mechanism for new installations installed with the Setup Wizard. Also, the Solr 4 server is supported only when running in a Tomcat application server. Therefore, if you are running Alfresco within a different application server and you wish to use Solr 4 search, you must install Tomcat.

Advantages of Solr 4 over Solr 1.4 search

Solr 4 search server brings improvements and new features over Solr 1.4 with respect to scalability, performance, and flexibility.

In particular, Solr 4 offers:

- More compact disc formats
- Faster index rebuilding
- Simpler and faster wildcard querying
- Use of doc values for faceting and ordering
- More accurate results and facet count
- Integrated Solr date math for `d:date` and `d:datetime` types
- Use of primitive types
- Support for spell checking and suggestion
- Support for site shortnames using `SITE` in queries and faceting using `TAG`
- Special tag support in queries and faceting

Eventual consistency

Alfresco One 5.0 introduces the concept of eventual consistency to overcome the scalability limitations of in-transaction indexing.

Alfresco One 5.0 with the Solr 4 subsystem does not include any transactional indexing. In other words, Alfresco removes the need to have the database and indexes in perfect sync at any given

time and relies on an index that gets updated at configurable intervals (default: 15s) by Solr 4 itself.

The index tracker takes care of polling Alfresco for new transactions and proceeds to update its index. In this sense, indexes will eventually be consistent with the database.

Configure Solr 4 search service

The way that you configure Alfresco to use Solr 4 depends on how you have installed Alfresco. If you install Alfresco using the setup wizard, Solr 4 is installed and enabled automatically. Solr 4 is installed in the same Tomcat container as Alfresco, and the connection URL is unchanged from the default. The Solr 4 home is within the Alfresco home directory.

This topic provides information on configuring the Solr 4 search subsystem. It also outlines the Solr 4 directory structure, configuration files, and properties.

Installing and configuring Solr 4

The Solr 4 search subsystem is installed by default when you install Alfresco One 5.0 using the setup wizards (installer), and therefore, you do not need to do these steps. If you install Alfresco manually using the distribution zip, you will need to configure Solr 4 separately on the existing Alfresco installation using Tomcat.

The installation contains the following artifacts:

- a template SOLR home directory containing `solr.xml`, which is expected by Solr 4
- Solr 4 WAR file
- an example context to wire up in Tomcat
- two Solr core configurations: one to track the live SpacesStore and one to track items archived from the SpacesStore

You can install Solr 4 either to the same Tomcat application server as Alfresco or a separate Tomcat. The Solr 4 server indexes data in Alfresco by periodically tracking the changes made to Alfresco. It does so by calling a RESTful API that describe the latest transactions to it. The Alfresco server performs searches through the Solr 4 server by issuing queries through a special API. For this reason, there needs to be two-way communication between the Alfresco server and the Solr 4 server. For security reasons, the communication channel between the Alfresco server and Solr 4 server must be secured by means of https encryption and mutual client certificate authentication.

The following instructions use `<ALFRESCO_TOMCAT_HOME>` to refer to the tomcat directory where Alfresco is installed and `<SOLR4_TOMCAT_HOME>` to the tomcat directory where Solr 4 is installed. These can be the same or different directories, depending on whether you have chosen to install Solr 4 on a standalone server.

1. Extract the `alfresco-enterprise-5.0.zip` file to a location. For example, `<EXTRACTED-ARCHIVE>`.
2. Copy the `<SOLR4-ARCHIVE>`.
3. Copy the `<ALFRESCO_HOME>/solr4/context.xml` file to `apache-tomcat-7.0.53\conf\Catalina\localhost\solr4.xml`.
4. Edit `solr4/home` in XML to point to `<SOLR4-ARCHIVE>`.

For example:

```
<?xml version="1.0" encoding="utf-8"?>
<Context debug="0" crossContext="true">
  <Environment name="solr4/home" type="java.lang.String" value="<SOLR4-ARCHIVE>" override="true"/>
  <Environment name="solr4/model/dir" type="java.lang.String"
value="@@ALFRESCO_SOLR4_MODEL_DIR@" override="true"/>
```

```
<Environment name="solr4/content/dir" type="java.lang.String"
value="@@ALFRESCO_SOLR4_CONTENT_DIR@" override="true"/>
</Context>
```

where:

- `@@ALFRESCO_SOLR4_MODEL_DIR@@` should point to the location of the Solr 4 model directory. For example, `<ALFRESCO_HOME>/alf_data/solr4/model`.
 - `@@ALFRESCO_SOLR4_CONTENT_DIR@@` should point to the location of the Solr 4 content directory. For example, `<ALFRESCO_HOME>/alf_data/solr4/content`.
5. For each core, edit the `solrcore.properties` file:
- `archive-SpacesStore/conf/solrcore.properties`
 - `workspace-SpacesStore/conf/solrcore.properties`

Set the `data.dir.root` property to the location where the Solr 4 indexes will be stored. You can set the same value for the both cores, and the cores will create the sub-directories.

6. Ensure that Alfresco has already been started at least once and the `<ALFRESCO_TOMCAT_HOME>/webapps/alfresco/WEB-INF` directory exists.
7. Create and populate a keystore directory for the Alfresco and Solr 4 servers. By default, the keystore directory is created in `<ALFRESCO_HOME>/alf_data/keystore`. Note that at this stage the keystore directory will just be a template, containing standard keys. To secure the installation, you must follow the steps to generate new keys as explained in the [Generating Secure Keys for Solr 4 Communication](#) section.

 If you turn off SSL between Solr 4 and Alfresco, you must generate and install your own certificates to secure the installation.

For example:

For Unix:

```
mkdir -p <ALFRESCO_HOME>/alf_data/keystore

cp <ALFRESCO_TOMCAT_HOME>/webapps/alfresco/WEB-INF/classes/alfresco/
keystore/* <ALFRESCO_HOME>/alf_data/keystore
```

For Windows:

```
mkdir <ALFRESCO_HOME>\alf_data\keystore

copy <ALFRESCO_TOMCAT_HOME>\webapps\alfresco\WEB-INF\classes\alfresco
\keystore\* <ALFRESCO_HOME>\alf_data\keystore
```

8. Configure the Alfresco and Solr 4 tomcat application servers to use the keystore and truststore for `https` requests by editing the specification of the connector on port 8443 in `<ALFRESCO_TOMCAT_HOME>/conf/server.xml` and `<SOLR4_TOMCAT_HOME>/conf/server.xml` as shown:

 Remember to replace `<ALFRESCO_HOME>/alf_data/keystore` with the full path to your keystore directory.

For example:

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11Protocol"
SSLEnabled="true" maxThreads="150" scheme="https"
keystoreFile="<ALFRESCO_HOME>/alf_data/keystore/ssl.keystore"
keystorePass="kT9X6oe68t" keystoreType="JCEKS" secure="true"
connectionTimeout="240000"
truststoreFile="<ALFRESCO_HOME>/alf_data/keystore/ssl.truststore"
truststorePass="kT9X6oe68t" truststoreType="JCEKS"
clientAuth="false" sslProtocol="TLS"/>
```

9. Configure Alfresco to use the keystore and truststore for client requests to Solr 4 by specifying `dir.keystore` in `<ALFRESCO_TOMCAT_HOME>/shared/classes/alfresco-global.properties`.

 Remember to replace `<ALFRESCO_HOME>/alf_data/keystore` with the full path to your keystore directory.

For example:

```
dir.keystore=<ALFRESCO_HOME>/alf_data/keystore
```

10. Configure an identity for the Alfresco server. In `<SOLR4_TOMCAT_HOME>/conf/tomcat-users.xml`, add the following:

 Remember, you can choose a different user name, such as the host name of the Alfresco server, but it must match the `REPO_CERT_DNAME` that you will later specify in the keystore in the [Generating Secure Keys for Solr 4 Communication](#) section.

For example:

```
<user username="CN=Alfresco Repository, OU=Unknown, O=Alfresco Software Ltd., L=Maidenhead, ST=UK, C=GB" roles="repository" password="null"/>
```

11. Configure an identity for the Solr 4 server. In `<ALFRESCO_TOMCAT_HOME>/conf/tomcat-users.xml`, add the following:

 Remember, you can choose a different user name but it must match the `SOLR_CLIENT_CERT_DNAME` that you will later specify in the keystore in the [Generating Secure Keys for Solr 4 Communication](#) section.

For example:

```
<user username="CN=Alfresco Repository Client, OU=Unknown, O=Alfresco Software Ltd., L=Maidenhead, ST=UK, C=GB" roles="reproclient" password="null"/>
```

12. To complete the installation, it is necessary to secure the two-way communication between Alfresco and Solr 4 by generating your own keys. See the [Generating Secure Keys for Solr 4 Communication](#) topic.

Generating secure keys for Solr 4 communication

This task describes how to replace or update the keys used to secure communication between Alfresco and Solr 4, using secure keys specific to your Alfresco installation.

The following instructions assume that Solr 4 has been extracted and a `keystore` directory has already been created, either automatically by the Alfresco installer or manually by following the instructions in the [Configuring Solr 4](#) section.

If you are applying these instructions to a clustered installation, the steps should be carried out on a single host and then the generated `.keystore` and `.truststore` files must be replicated across all other hosts in the cluster.

1. Obtain the file `generate_keystores.sh` (for Linux and Solaris) or `generate_keystores.bat` (for Windows) from the [Alfresco Customer Support](#) website under **Online Resources > Downloads > Alfresco Enterprise 5.0 > <Alfresco Version> generate_keystores.x**.
2. Edit the environment variables at the beginning of the file to match your environment.
 - a. If you are updating an environment created by the Alfresco installer, you only need to edit `ALFRESCO_HOME` to specify the correct installation directory.
 - b. For manual installations, carefully review `ALFRESCO_KEYSTORE_HOME`, `SOLR_HOME`, `JAVA_HOME`, `REPO_CERT_DNAME` and `SOLR_CLIENT_CERT_DNAME` and edit as appropriate.
3. Run the edited script.

You should see the message “Certificate update complete” and another message reminding you what `dir.keystore` should be set to in the `alfresco-global.properties` file.

Solr 4 directory structure

After you have installed Alfresco, several directories and configuration files related to Solr 4 will be available in the Alfresco home directory. This section explains the Solr 4 directory structure.

`alfresco\solr4`

This is the Solr 4 home directory. It contains the Solr cores: `archive-SpacesStore` (for deleted content) and `workspace-SpacesStore` (for live content). It also contains two configurations files: `context.xml` and `solr.xml`.

The Solr 4 directory contains the following sub-folders and files:

- `alfrescoModels`: This directory contains all the content models that come out of the box with Alfresco. Any new custom content model added to Alfresco are synced to this directory so that Solr 4 knows about it.
- `archive-SpacesStore`: This is the configuration directory for the archive core.
- `workspace-SpacesStore`: This is the configuration directory for the workspace core.
- `context.xml`: This configuration file specifies the Solr 4 web application context template to use when installing Solr 4 in separate tomcat server.
- `log4j-solr.properties`: This is the configuration file for Solr 4-specific logging.
- `solr.xml`: This configuration file specifies the cores to be used by Solr 4.

`alfresco\alf_data\solr4\`

The Solr 4 directory contains the following sub-folders:

- `content`: This directory contains a copy of every Solr 4 document added to the index.
- `index`: This directory contains the actual Lucene index.
- `model`: This directory contains the models that are synced from Alfresco.

`alfresco\alf_data\solr4Backup\`

This directory stores the Solr 4 backup. It contains the `alfresco` and `archive` sub-directories.

Solr 4 configuration files

When you install Alfresco One 5.0, several Solr 4 configuration files are made available to you. The section lists the Solr 4 configuration files, their location in the Alfresco directory structure and description.

Configuration File	Location	Description
<code>repository.properties</code>	<code>alfresco\tomcat\webapps\alfresco\WEB-INF\classes\alfresco\</code>	This file specifies the Solr 4-related properties in how Alfresco connects to the Solr 4 server. As the Solr 4 server runs in the same Tomcat instance as Alfresco, the connection properties are setup to connect to a locally running Solr 4 server. The properties of this file are managed by an Alfresco Administrator user.

Configuration File	Location	Description
schema.xml	alfresco \solr4\ <code><core></code> \conf \, where <code><core></code> is the location of core's configuration directory. For example alfresco \solr4\workspace-SpacesStore \conf or alfresco \solr4\archive-SpacesStore\conf	This file defines the schema for the index including field type definitions with associated analyzers. It contains details about the fields that you can include in your document and also describes how those fields can be used when adding documents to the index or when querying those fields. The properties of this file are managed by an expert user.
solr4.xml	alfresco\tomcat \conf\catalina \localhost\	This file defines the Solr 4 web application context. It specifies the location of the Solr 4 war file and sets up the Solr 4 home directory.
solr.xml	alfresco\alf_data \solr4	This file specifies the cores to be used by Solr 4.
core.properties	<ALFRESCO_HOME>/ solr4/archive-SpacesStore/ core.properties or <ALFRESCO_HOME>/ solr4/workspace-SpacesStore/ core.properties	This file specifies the cores to be used by Solr 4.
solrconfig.xml	alfresco \solr4\workspace-SpacesStore \conf or alfresco \solr4\archive-SpacesStore\conf	This file specifies the parameters for configuring Solr 4. Also, the Solr 4 search components are added to this file. The properties of this file are managed by an Alfresco expert Administrator user.
solrcore.properties	alfresco \solr4\workspace-SpacesStore \conf or alfresco \solr4\archive-SpacesStore\conf	This is the property configuration file for a core. Solr 4 supports system property substitution, so properties that need substitution can be put in to this file. There is one <code>solrcore.properties</code> file in each core's configuration directory. For details, see the Solr 4 core configuration properties topic. The properties of this file are managed by an Alfresco Administrator user.
context.xml	alfresco\solr4	This file specifies the Solr 4 web application context template to use when installing Solr 4 in separate tomcat server.
ssl.repo.client.keystore	alfresco \solr4\workspace-SpacesStore \conf or alfresco \solr4\archive-SpacesStore\conf	This keystore contains the Solr 4 public/private RSA key pair.
ssl-keystore-passwords.properties	alfresco \solr4\workspace-SpacesStore \conf or alfresco \solr4\archive-SpacesStore\conf	This file contains the password information for <code>ssl.repo.client.keystore</code> .

Configuration File	Location	Description
ssl.repo.client.truststore	alfresco \solr4\workspace- SpacesStore \conf or alfresco \solr4\archive- SpacesStore\conf	This keystore contains the trusted Alfresco Certificate Authority certificate (which has been used to sign both the repository and Solr 4 certificates)
ssl-truststore- passwords.properties	alfresco \solr4\workspace- SpacesStore \conf or alfresco \solr4\archive- SpacesStore\conf	This file contains the password information for ssl.repo.client.truststore.

Solr 4 core configuration properties

The `solrcore.properties` configuration file is the property configuration file for a Solr 4 core. There is one `solrcore.properties` file in each core's configuration directory. This section lists the properties of this file, their description, and the default value.

Property Name	Description	Default Value
data.dir.root	This property specifies the top level directory path for the indexes managed by Solr 4.	C:/Alfresco/alf_data/solr4/index
data.dir.store	This property specifies the directory relative to <code>data.dir.root</code> where the data for this core is stored.	workspace/SpacesStore
enable.alfresco.tracking	This property instructs Solr 4 if it should index Alfresco content in the associated Alfresco repository store or not.	true
max.field.length	This property specifies the maximum number of tokens to include for each field. By default, all tokens are added.	2147483647
alfresco.version	This property specifies the Alfresco version installed.	5.0
alfresco.host	This property specifies the host name for the Alfresco instance that Solr 4 should track and index. In a default installation, both Alfresco and Solr 4 runs in the same Tomcat instance and on the same host, so host would be set to local host.	localhost
alfresco.port	This property specifies the HTTP port for the Alfresco instance that Solr 4 should track and index.	8080
alfresco.port.ssl	This property specifies the HTTPS port for the Alfresco instance that Solr 4 should track and index.	8443

Property Name	Description	Default Value
<code>alfresco.cron</code>	This property specifies the cron expression that instructs Solr 4 how often to track Alfresco and index new or updated content. The default value indicates that Solr 4 tracks Alfresco every 15 seconds.	<code>0/15 * * * * ? *</code>
<code>alfresco.stores</code>	This property specifies the Alfresco repository store that this core should index.	<code>workspace://SpacesStore</code>
<code>alfresco.baseUrl</code>	This property configures the base URL to Alfresco web project. If you need to change the <code>baseUrl</code> value, see Deploying Alfresco with a different context path on page 102 for configuring information.	<code>/alfresco</code>
<code>alfresco.lag</code>	When Solr 4 tracking starts, it aims to get up to date to the current time (in seconds), less this lag.	<code>1000</code>
<code>alfresco.hole.retention</code>	Each track will revisit all transactions from the timestamp of the last in the index, less this value, to fill in any transactions that might have been missed.	<code>3600000</code>
<code>alfresco.batch.count</code>	This property indicates the number of updates that should be made to this core before a commit is executed.	<code>1000</code>
<code>alfresco.secureComms</code>	This property instructs Solr 4 if it should talk to Alfresco over HTTP or HTTPS. Set to <code>none</code> if a plain HTTP connection should be used.	<code>https</code>
<code>alfresco.encryption.ssl.keyStoreType</code>	This property specifies the CLIENT keystore type.	<code>JCEKS</code>
<code>alfresco.encryption.ssl.keyStoreProvider</code>	This property specifies the Java provider that implements the <code>type</code> attribute (for example, JCEKS type). The provider can be left unspecified and the first provider that implements the keystore type specified is used.	
<code>alfresco.encryption.ssl.keyStoreLocation</code>	This property specifies the CLIENT keystore location reference. If the keystore is file-based, the location can reference any path in the file system of the node where the keystore is located.	<code>ssl.repo.client.keystore</code>

Property Name	Description	Default Value
<code>alfresco.encryption.ssl.keyLocation</code>	This property specifies the location of the file containing the password that is used to access the CLIENT keystore, also the default that is used to store keys within the keystore.	<code>ssl-keystore-passwords.properties</code>
<code>alfresco.encryption.ssl.truststoreType</code>	This property specifies the CLIENT truststore type.	JCEKS
<code>alfresco.encryption.ssl.truststoreProvider</code>	This property specifies the Java provider that implements the type attribute (for example, JCEKS type). The provider can be left unspecified and the first provider that implements the truststore type specified is used.	
<code>alfresco.encryption.ssl.truststoreLocation</code>	This property specifies the CLIENT truststore location reference. If the truststore is file-based, the location can reference any path in the file system of the node where the truststore is located.	<code>ssl.repo.client.truststore</code>
<code>alfresco.encryption.ssl.truststoreKeyLocation</code>	This property specifies the location of the file containing the password that is used to access the CLIENT truststore, also the default that is used to store keys within the truststore.	<code>ssl-truststore-passwords.properties</code>
<code>alfresco.corePoolSize</code>	This property specifies the pool size for multi-threaded tracking. It is used for indexing nodes.	3
<code>alfresco.maximumPoolSize</code>	This property specifies the maximum pool size for multi-threaded tracking.	-1
<code>alfresco.keepAliveTime</code>	This property specifies the time (in seconds) to keep non-core idle threads in the pool.	120
<code>alfresco.threadPriority</code>	This property specifies the priority that all threads must have on the scale of 1 to 10, where 1 has the lowest priority and 10 has the highest priority.	5
<code>alfresco.threadDaemon</code>	This property sets whether the threads run as daemon threads or not. If set to <code>false</code> , shut down is blocked else it is left unblocked.	<code>true</code>
<code>alfresco.workQueueSize</code>	This property specifies the maximum number of queued work instances to keep before blocking against further adds.	-1

Property Name	Description	Default Value
alfresco.maxTotalConnections	This property is used for HTTP client configuration.	40
alfresco.maxHostConnections	This property is used for HTTP client configuration.	40
alfresco.socketTimeout	This property specifies the amount of time Solr 4 tracker will take to notice if the Alfresco web app shuts down first, if Alfresco and Solr 4 are running on the same web application.	60000
solr.filterCache.size	This property specifies the maximum number of entries in the Solr 4 filter cache.	64
solr.filterCache.initialSize	This property specifies the initial capacity (number of entries) of the Solr 4 filter cache.	64
solr.queryResultCache.size	This property configures the Solr 4 result cache.	1024
solr.queryResultCache.initialSize	This property configures the Solr 4 result cache.	1024
solr.documentCache.size	This property configures the Solr 4 document cache.	64
solr.documentCache.initialSize	This property configures the Solr 4 document cache.	64
solr.queryResultMaxDocsCache	This property configures the Solr 4 result cache.	2000
solr.authorityCache.size	This property configures the Solr 4 result cache.	64
solr.authorityCache.initialSize	This property configures the Solr 4 result cache.	64
solr.pathCache.size	This property configures the Solr 4 result cache.	64
solr.pathCache.initialSize	This property configures the Solr 4 result cache.	64
solr.ownerCache.size	This property configures the Solr 4 result cache.	4096
solr.ownerCache.initialSize	This property configures the Solr 4 result cache.	1024
solr.readerCache.size	This property configures the Solr 4 result cache.	4096
solr.readerCache.initialSize	This property configures the Solr 4 result cache.	1024
solr.deniedCache.size	This property configures the Solr 4 result cache.	4096
solr.deniedCache.initialSize	This property configures the Solr 4 result cache.	1024

Property Name	Description	Default Value
<code>solr.nodeBatchSize</code>	This property configures the Solr 4 result cache.	10
<code>solr.filterCache.autowarmCount</code>	This property configures the Solr 4 result cache.	128
<code>solr.authorityCache.autowarmCount</code>	This property configures the Solr 4 result cache.	0
<code>solr.pathCache.autowarmCount</code>	This property configures the Solr 4 result cache.	128
<code>solr.deniedCache.autowarmCount</code>	This property configures the Solr 4 result cache.	0
<code>solr.readerCache.autowarmCount</code>	This property configures the Solr 4 result cache.	0
<code>solr.ownerCache.autowarmCount</code>	This property configures the Solr 4 result cache.	0
<code>solr.queryResultCache.autowarmCount</code>	This property configures the Solr 4 result cache.	0
<code>solr.documentCache.autowarmCount</code>	This property configures the Solr 4 result cache.	0
<code>solr.queryResultWindowSize</code>	This property configures the Solr 4 result cache.	200
<code>alfresco.doPermissionCheck</code>	This property configures the Solr 4 result cache.	true
<code>alfresco.metadata.skipDescriptionTypes</code>	This property configures the Solr 4 result cache.	false
<code>alfresco.metadata.ignore.default</code>	This property configures the Solr 4 result cache.	cm:person
<code>alfresco.metadata.ignore.default</code>	This property configures the Solr 4 result cache.	app:configurations
<code>solr.maxBooleanClauses</code>	This property specifies the number of Boolean clauses in a query. It can affect range or wildcard queries that expand to big Boolean queries.	10000
<code>alfresco.transactionDocsBatchSize</code>	This property is used for batch fetching updates during tracking.	100
<code>alfresco.changeSetAclsBatchSize</code>	This property is used for batch fetching updates during tracking.	100
<code>alfresco.aclBatchSize</code>	This property is used for batch fetching updates during tracking.	10
<code>alfresco.index.transformContent</code>	If this property is set to false, the index tracker will not transform any content and only the metadata will be indexed.	false

Solr 4 subsystem

Search is contained in a subsystem, and it has an implementation `solr4`.

The following properties in the `alfresco-global.properties` file are related to Solr 4 and are setup as follows, by default:

```
### Solr indexing ###
index.subsystem.name=solr4
dir.keystore=${dir.root}/keystore
solr.port.ssl=8443
```

Activating Solr 4

This information describes how to activate the Solr 4 search mechanism in a manual Alfresco installation or when upgrading from a previous version.

Global properties file

1. Open the `<classpathRoot>\alfresco-global.properties` file.
2. Set the following properties:

Property	Description
<code>index.subsystem.name</code>	The subsystem type value. The default value is <code>solr4</code> .
<code>solr.host</code>	The host name where the Solr 4 instance is located.
<code>solr.port</code>	The port number on which the Solr 4 instance is running.
<code>solr.port.ssl</code>	The port number on which the Solr 4 SSL support is running.

For example, some example properties for activating Solr 4 are:

```
index.subsystem.name=solr4
solr.host=localhost
solr.port=8080
solr.port.ssl=8443
```

3. Save the global properties file and restart the Alfresco server.

Share Admin Console

1. Open the Share Admin Console.
2. Edit the following properties:

Property	Description
<code>index.subsystem.name</code>	Select the subsystem type value as <code>solr4</code> .
<code>solr.host</code>	The host name where the Solr 4 instance is located.
<code>solr.port</code>	The port number on which the Solr 4 instance is running.
<code>solr.port.ssl</code>	The port number on which the Solr 4 SSL support is running.

3. Click Save.

JMX client

1. Navigate to **MBeans > Alfresco > Configuration > Search**.
2. Set the manager `sourceBeanName` to `solr4`.

The subsystems have their own related properties. The `managed - solr4` instance exposes the `solr.base.url` property.

3. These can now be configured live and the subsystem redeployed.

Solr 4 security

Communications between Alfresco repository and Solr 4 are protected by SSL with mutual authentication.

Both the repository and Solr 4 have their own public/private key pair, signed by an Alfresco Certificate Authority, which are stored in their own respective keystores. These keystores are bundled with Alfresco. You can also create your own keystores. For an overview on how to create an SSL public/private key and certificate for the repository, see [Generating Repository SSL Keystores](#).

Keystores are used also to protect repository and Solr 4 communications using encryption and mutual authentication. To do this, keystores store RSA keys and certificates.

It is assumed that the keystore files are stored in `alf_data`. Place the keystore files from the directory `repository/config/alfresco/keystore` in the `$ALF_DATA/keystore` directory.

Repository SSL key stores

This section describes the key stores used by the repository for SSL.

The repository has two key stores it uses for SSL:

- `ssl keystore` contains a public/private RSA key pair for the repository
- `ssl truststore` contains the trusted Alfresco Certificate Authority certificate (which has been used to sign both the repository and Solr 4 certificates)

These key stores can be stored in any location.

Update the following key store properties in the `alfresco-global.properties` file to specify the location of the key stores:

`ssl keystore`

Property	Description
<code>encryption.ssl.keystore.location</code>	Specifies the key store location.
<code>encryption.ssl.keystore.provider</code>	Specifies the key store provider.
<code>encryption.ssl.keystore.type</code>	Specifies the key store type.
<code>encryption.ssl.keystore.keyMetaData.location</code>	Specifies the key store metadata file location.

`ssl truststore`

Property	Description
<code>encryption.ssl.truststore.location</code>	Specifies the trust store location.
<code>encryption.ssl.truststore.provider</code>	Specifies the trust store provider.
<code>encryption.ssl.truststore.type</code>	Specifies the trust store type.
<code>encryption.ssl.truststore.keyMetaData.location</code>	Specifies the trust store metadata file location.

Solr 4 SSL key stores

This section describes the key stores used by Solr 4 for SSL.

Solr 4 core has two key stores it uses for SSL:

- `ssl.repo.client.keystore` contains a Solr 4 public/private RSA key pair
- `ssl.repo.client.truststore` contains the trusted Alfresco Certificate Authority certificate (which has been used to sign both the repository and Solr 4 certificates)

Connecting to the SSL-protected Solr 4 web application

The Solr 4 Admin Web interface allows you to view Solr 4 configuration details, run queries, and analyze document fields.

All Solr 4 URLs, which are bundled within Alfresco, are protected by SSL. To use these URLs from a browser, you need to import a browser-compatible key store to allow mutual authentication and decryption to work. The following steps describe how to import the key store into your browser (these relate to Firefox, other browsers will have a similar mechanism):

1. Open the FireFox **Certificate Manager** by selecting **Firefox > Preferences... > Advanced > Certificates > View Certificates > Your Certificates**.
2. Import the browser keystore `browser.p12` that is located in your `<ALFRESCO_HOME>/alf_data/keystore` directory.
3. Enter the password `alfresco`.
A window displays showing that the key store has been imported successfully. The **Certificate Manager** now contains the imported key store with the Alfresco repository certificate under the **Your Certificates** tab.
4. Close the **Certificate Manager** by clicking **OK**.
5. In the browser, navigate to a Solr 4 URL, <http://localhost:8080/solr4>.
The browser displays an error message window to indicate that the connection is untrusted. This is due to the Alfresco certificate not being tied to the server IP address. In this case, view the certificate and confirm that it is signed by the Alfresco Certificate Authority.
6. Expand **I understand the risks**.
7. Select **Add Exception**.
8. Click **View**.
This displays the certificate.
9. Confirm that the certificate was issued by Alfresco Certificate Authority, and then confirm the **Security Exception**.

Access to Solr 4 is granted and the Solr 4 Admin page is displayed. It is divided into two parts.

The left-side of the screen is a menu under the Solr logo that provides navigation through various screens. The first set of links are for system-level information and configuration and provide access to Logging, Core Admin and Java Properties. At the end of this information is a list of Solr cores configured for this instance of Alfresco.

The center of the screen shows the detail of the Solr core selected, such as statistics, summary report, and so on.

The screenshot shows the Apache Solr Admin interface. On the left is a navigation sidebar with options like Dashboard, Logging, Core Admin, Java Properties, Thread Dump, Overview, Analysis, Documents, Files, Ping, Plugins / Stats, Query, Replication, and Schema Browser. The main content area is divided into several sections:

- Statistics:** Shows system metrics such as Last Modified (6 days ago), Num Docs (903), Max Doc (931), Heap Memory Usage (1093688), Deleted Docs (28), Version (4.1), Segment Count (9), and an 'optimize now' button.
- Instance:** Displays instance details including CID (/Applications/alfresco-5.0), Instance (/Applications/alfresco-5.0/solr4/workspace-SpacesStore), Data (/Applications/alfresco-5.0/alf_data/solr4/index/workspace/SpacesStore), Index (/Applications/alfresco-5.0/alf_data/solr4/index/workspace/SpacesStore/index), and Impl (org.apache.solr.core.NRTCachingDirectoryFactory).
- Replication (Master):** A table showing replication status for Master (Searching) and Master (Replicable).
- Admin Extra:** Includes buttons for 'Update All' (to update Summary and FTS Status reports) and 'Alfresco Core - Summary Report'.
- Alfresco Core - FTS Status Report:** Shows FTS Status Clean (163), FTS Status Dirty (0), and FTS Status New (0). It also includes a note that the full report can take some time to generate.
- Other Links:** Provides links to Solr Errors, Solr Exception Messages, and Solr Exceptions Stack.

At the bottom of the page, there are links for Documentation, Issue Tracker, IRC Channel, Community forum, and Solr Query Syntax.

Generating repository SSL keystores

This task describes how to create an SSL public/private keystore and a certificate for the repository.

The following instructions create an RSA public/private key pair for the repository with a certificate signed by the Alfresco Certificate Authority (CA). It also creates a truststore for the repository containing the CA certificate that is used to authenticate connections to specific repository URLs from Solr 4. The instructions assume the existence of the Alfresco CA key and certificate to sign the repository certificate. However, for security reasons these might not be available. You can either generate your own CA key and certificate or use a recognised Certificate Authority, such as Verisign. To generate your own CA key and certificate, see [Generating CA key and certificate](#).



<store password> is the keystore password. The file `C:\Alfresco\alf_data\keystore\ssl-keystore-passwords.properties` contains passwords for the SSL keystore, whereas, the file `C:\Alfresco\alf_data\keystore\ssl-truststore-passwords.properties` contains passwords for the SSL truststore.

1. Generate the repository public/private key pair in a keystore.

```
$ keytool -genkey -alias repo -keyalg RSA -keystore ssl.keystore -
storetype JCEKS -storepass <store password>
Enter keystore password:
Re-enter new password:
What is your first and last name?
 [Unknown]:  Alfresco Repository
What is the name of your organizational unit?
 [Unknown]:
What is the name of your organization?
 [Unknown]:  Alfresco Software Ltd.
What is the name of your City or Locality?
 [Unknown]:  Maidenhead
What is the name of your State or Province?
 [Unknown]:  UK
What is the two-letter country code for this unit?
 [Unknown]:  GB
Is CN=Alfresco Repository, OU=Unknown, O=Alfresco Software Ltd.,
L=Maidenhead, ST=UK, C=GB correct?
 [no]:  yes
```

```
Enter key password for <repo>
(RETURN if same as keystore password):
```

2. Generate a certificate request for the repository key.

```
$ keytool -keystore ssl.keystore -alias repo -certreq -file repo.csr -
storetype JCEKS -storepass <store password>
```

3. Alfresco CA signs the certificate request and creates a certificate that is valid for 365 days.

```
$ openssl x509 -CA ca.crt -CAkey ca.key -CAcreateserial -req -in repo.csr
-out repo.crt -days 365
Signature ok
subject=/C=GB/ST=UK/L=Maidenhead/O=Alfresco Software Ltd./OU=Unknown/
CN=Alfresco Repository
Getting CA Private Key
Enter pass phrase for ca.key:
```

4. Import the Alfresco CA key into the repository keystore.

```
$ keytool -import -alias AlfrescoCA -file ca.crt -keystore ssl.keystore -
storetype JCEKS -storepass <store password>
Enter keystore password:
Owner: CN=Alfresco CA, O=Alfresco Software Ltd., L=Maidenhead, ST=UK,
C=GB
Issuer: CN=Alfresco CA, O=Alfresco Software Ltd., L=Maidenhead, ST=UK,
C=GB
Serial number: 805ba6dc8f62f8b8
Valid from: Fri Aug 12 13:28:58 BST 2011 until: Mon Aug 09 13:28:58 BST
2021
Certificate fingerprints:
MD5: 4B:45:94:2D:8E:98:E8:12:04:67:AD:AE:48:3C:F5:A0
SHA1: 74:42:22:D0:52:AD:82:7A:FD:37:46:37:91:91:F4:77:89:3A:C9:A3
Signature algorithm name: SHA1withRSA
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 08 42 40 DC FE 4A 50 87 05 2B 38 4D 92 70 8E 51 .B@..JP..+8M.p.Q
0010: 4E 38 71 D6 N8q.
]
]

#2: ObjectId: 2.5.29.19 Criticality=false
BasicConstraints:[
CA:true
PathLen:2147483647
]

#3: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
0000: 08 42 40 DC FE 4A 50 87 05 2B 38 4D 92 70 8E 51 .B@..JP..+8M.p.Q
0010: 4E 38 71 D6 N8q.
]
]

[CN=Alfresco CA, O=Alfresco Software Ltd., L=Maidenhead, ST=UK, C=GB]
SerialNumber: [ 805ba6dc 8f62f8b8]
]

Trust this certificate? [no]: yes
Certificate was added to keystore
```

5. Import the CA-signed repository certificate into the repository keystore.

```
$ keytool -import -alias repo -file repo.crt -keystore ssl.keystore -
storetype JCEKS -storepass <store password>
```

```
Enter keystore password:
Certificate reply was installed in keystore
```

6. Convert the repository keystore to a pkcs12 keystore (for use in browsers, such as Firefox). Specify the keystore password for pkcs12 keystore as 'alfresco'.

```
keytool -importkeystore -srckeystore ssl.keystore -srcstorepass <keystore
password> -srcstoretype JCEKS -srcalias
repo -srckeypass kT9X6oe68t -destkeystore firefox.p12 -deststoretype
pkcs12 -deststorepass alfresco -destalias repo
-destkeypass alfresco
```

7. Create a repository truststore containing the Alfresco CA certificate.

```
keytool -import -alias AlfrescoCA -file ca.crt -keystore ssl.keystore -
storetype JCEKS -storepass <store password>
```

8. Copy the keystore and truststore to the repository keystore location defined by the property `dir.keystore`.
9. Update the SSL properties (properties starting with the prefixes `alfresco.encryption.ssl.keystore` and `alfresco.encryption.ssl.truststore`).

Generating a Certificate Authority (CA) key and certificate

This task describes how to create Alfresco CA key and certificate to sign the repository certificate.

1. Generate the CA private key.

```
$ openssl genrsa -des3 -out ca.key 1024
Generating RSA private key, 1024 bit long modulus
.....++++++
..++++++
e is 65537 (0x10001)
Enter pass phrase for ca.key:
Verifying - Enter pass phrase for ca.key:
```

2. Generate the CA self-signed certificate.

```
$ openssl req -new -x509 -days 3650 -key ca.key -out ca.crt
Enter pass phrase for ca.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:GB
State or Province Name (full name) [Some-State]:UK
Locality Name (eg, city) []:Maidenhead
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Alfresco
Software Ltd.
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:Alfresco CA
Email Address []:
```

Solr 4 monitoring and troubleshooting

This section provides help for monitoring and resolving any Solr 4 index issues that might arise as a result of a transaction.

Reindexing a Solr 4 transaction

This task describes how to reindex a Solr 4 transaction.

This task assumes you are using only one Solr 4 instance for all nodes in the Alfresco cluster. If not, then you need to repeat process on each Solr 4 instance used in the cluster.

1. Confirm the location of the Solr 4 core directories for `archive-SpacesStore` and `workspace-SpacesStore` cores. This can be determined from the `solrcore.properties` file for both the cores. By default, the `solrcore.properties` file can be found at `<ALFRESCO_HOME>/solr4/workspace-SpacesStore/conf` or `<ALFRESCO_HOME>/solr4/archive-SpacesStore/conf`. The Solr 4 core location is defined in the `solrcore.properties` file as:

For Solr 4, the default `data.dir.root` path is:

```
data.dir.root=<ALFRESCO_HOME>/alf_data/solr4/indexes/
```

2. Shut down Alfresco (all nodes, if clustered).
3. Shut down Solr 4 (if running on a separate application server).
4. Delete the contents of the index data directories for each Solr core at `${data.dir.root}/${data.dir.store}`.

- `<ALFRESCO_HOME>/alf_data/solr4/index/workspace/SpacesStore`
- `<ALFRESCO_HOME>/alf_data/solr4/index/archive/SpacesStore`

5. Delete all the Alfresco models for each Solr 4 core at `${data.dir.root}`.

```
<ALFRESCO_HOME>/alf_data/solr4/model
```

6. Delete the contents of the `<ALFRESCO_HOME>/alf_data/solr4/content` directory.
7. Start up the application server that runs Solr 4.
8. Start up the Alfresco application server (if not same as Solr 4 application server).
9. Monitor the application server logs for Solr. You will get the following warning messages on bootstrap:

```
WARNING: [alfresco] Solr index directory '<ALFRESCO_HOME>/alf_data/solr/
workspace/SpacesStore/index' doesn't exist. Creating new index...
09-May-2012 09:23:42
org.apache.solr.handler.component.SpellCheckComponent inform
WARNING: No queryConverter defined, using default converter
09-May-2012 09:23:42 org.apache.solr.core.SolrCore initIndex
WARNING: [archive] Solr index directory '<ALFRESCO_HOME>/alf_data/solr/
archive/SpacesStore/index' doesn't exist. Creating new index...
```

10. Use the Solr 4 administration console to check the health of the Solr 4 index.



The process of building the Solr 4 indexes can take some time depending on the size of the repository. To monitor reindexing progress, use the Solr 4 administration console and check the logs for any issues during this activity. Also, it is advised not to use Alfresco application as the query result set might not be correct until full reindex is complete.

Unindexed Solr 4 Transactions

You can check the status of the Solr 4 index to identify the nodes to a transaction that failed to index.

To generate a general report for Solr 4, including the last transaction indexed and the time, use:

```
http://localhost:8080/solr4/admin/cores?action=REPORT&wt=xml
```

The `REPORT` parameter compares the database with the index and generates an overall status report with the following details:

- DB transaction count: indicates the transaction count on the database
- DB acl transaction count: indicates the ACL transaction count on the database
- Count of duplicated transactions in the index: indicates the number of transactions that appear more than once in the index. The value of this parameter should be zero. If not, there is an issue with the index.

- `Count of duplicated acl transactions in the index`: indicates the number of ACL transactions that appear more than once in the index. The value of this parameter should be zero. If not, there is an issue with the index.
- `Count of transactions in the index but not the database`: indicates the number of transactions in the index but not in the database. This count includes empty transactions that have been purged from the database. The value of this parameter should be zero. If not, there might be an issue with the index.
- `Count of acl transactions in the index but not the DB`: indicates the number of ACL transactions in the index but not in the database. The value of this parameter should be zero. If not, there is an issue with the index. Note that empty ACL transactions are not purged from the database.
- `Count of missing transactions from the Index`: indicates the number of transactions in the database but not in the index. The value of this index should be zero when the index is up-to-date.
- `Count of missing acl transactions from the Index`: indicates the number of ACL transactions in the database but not in the index. The value of this index should be zero when the index is up-to-date.
- `Index transaction count`: indicates the number of transactions in the index.
- `Index acl transaction count`: indicates the number of ACL transactions in the index.
- `Index unique transaction count`: indicates the number of unique transactions in the index.
- `Index unique acl transaction count`: indicates the number of unique ACL transactions in the index.
- `Index leaf count`: indicates the number of docs and folders in the index.
- `Count of duplicate leaves in the index`: indicates the number of duplicate docs or folders in the index. The value of this parameter should be zero. If not, there is an issue with the index.
- `Last index commit time`: indicates the time stamp for the last transaction added to the index. It also indicates that transactions after this time stamp have not yet been indexed.
- `Last Index commit date`: indicates the time stamp as date for the last transaction added to the index. It also indicates that transactions after this date have not yet been indexed.
- `Last TX id before holes`: indicates that transactions after this ID will be checked again to make sure they have not been missed. This is computed from the index at start up time. By default, it is set an hour after the last commit time found in the index. Solr 4 tracking, by default, goes back an hour from the current time to check that no transactions have been missed .
- `First duplicate` : indicates if there are duplicate transactions in the index. It returns the ID of the first duplicate transaction.
- `First duplicate acl tx`: indicates if there are duplicate ACL transactions in the index. It returns the ID of the first duplicate ACL transaction.
- `First transaction in the index but not the DB`: if the related count is > 0 , it returns the ID of the first offender.
- `First acl transaction in the index but not the DB`: if the related count is > 0 , it returns the ID of the first offender.
- `First transaction missing from the Index`: if the related count is > 0 , it returns the ID of the first offender.
- `First acl transaction missing from the Index`: if the related count is > 0 , it returns the ID of the first offender.

- First duplicate leaf in the index: if the related count is > 0, it returns the ID of the first offender.

To generate a summary report for Solr 4, use:

<http://localhost:8080/solr4/admin/cores?action=SUMMARY&wt=xml>

With multi-threaded tracking, you can specify additional tracking details and tracking statistics:

- `detail=true`: provide statistics per tracking thread
- `hist=true`: provides a histogram of the times taken for tracking operations for each tracking thread
- `reset=true`: resets all tracking statistics
- `values=true`: reports (by default) the last 50 values recorded for each tracking operation for each thread

The `SUMMARY` parameter provides the status of the tracking index and reports the progress of each tracking thread. It generates a report with the following details:

- `Active`: indicates the tracker for the core active.
- `Last Index Commit Time`: indicates the time stamp for the last transaction that was indexed.
- `Last Index Commit Date`: indicates the time stamp as a date for the last transaction that was indexed. Changes made after this time are not yet in the index.
- `Lag`: indicates the difference in seconds between the last transaction time stamp on the server and the time stamp for the last transaction that was indexed.
- `Duration`: indicates the time lag as an XML duration.
- `Approx transactions remaining`: indicates the approximate number of transactions to index in order to bring the index up-to-date. It is calculated as the last transaction ID on the server minus the last transaction ID indexed. It includes all the missing and empty transactions.
- `Approx transaction indexing time remaining`: it is based on `Approx transactions remaining`, the average number of nodes per transaction and the average time to index a node (how long the index will take to be up-to-date). The estimate is in the most appropriate scale, for example, seconds, minutes, hours and days.
- `Model sync times (ms)`: indicates summary statistics for model sync time. It supports additional information with `&detail=true`, `&hist=true` and `&value=true`.
- `Acl index time (ms)`: indicates summary statistics for ACL index time. It supports additional information with `&detail=true`, `&hist=true` and `&value=true`.
- `Node index time (ms)`: indicates summary statistics for node index time. It supports additional information with `&detail=true`, `&hist=true` and `&value=true`.
- `Acl tx index time (ms)`: indicates the summary statistics for ACL transaction index time. It supports additional information with `&detail=true`, `&hist=true` and `&value=true`.
- `Tx index time (ms)`: indicates summary statistics for transaction index time. It specifies the estimated time required to bring the index up-to-date.
- `Docs/Tx`: indicates summary statistics for the number of documents per transaction. It supports additional information with `&detail=true`, `&hist=true` and `&value=true`.
- `Doc Transformation time (ms)`: indicates summary statistics for document transformation time. It supports additional information with `&detail=true`, `&hist=true` and `&value=true`.

Troubleshooting Solr 4 Index

This section describes how to repair a transaction that failed to index.

 The default URL for the Solr4 index is `http://localhost:8080/solr4/...`

To repair an unindexed or failed transaction (as identified by the `REPORT` option in the [Unindexed Solr 4 Transactions](#) section), run the following report:

```
http://localhost:8080/solr4/admin/cores?action=FIX
```

The `FIX` parameter compares the database with the index and identifies any missing or duplicate transactions. It then updates the index by either adding or removing transactions.

Use the `PURGE` parameter to remove transactions, acl transactions, nodes and acls from the index. It can also be used for testing wrong transactions and then to fix them.

```
http://localhost:8080/solr4/admin/cores?
action=PURGE&txid=1&acltxid=2&nodeid=3&aclid=4
```

Use the `REINDEX` parameter to reindex a transaction, acl transactions, nodes and acls.

```
http://localhost:8080/solr4/admin/cores?
action=REINDEX&txid=1&acltxid=2&nodeid=3&aclid=4
```

Use the `INDEX` parameter to create entries in the index. It can also be used to create duplicate index entries for testing.

```
http://localhost:8080/solr4/admin/cores?
action=INDEX&txid=1&acltxid=2&nodeid=3&aclid=4
```

Use the `RETRY` parameter to retry indexing any node that failed to index and was skipped. In other words, it enables the users to attempt to fix documents that failed to index in the past and appear in the solr report (`http://localhost:8080/solr/admin/cores?action=REPORT&wt=xml`) with the field **Index error count**.

```
http://localhost:8080/solr4/admin/cores?action=RETRY
```

Use the following setting to specify an option core for the report. If it is absent, a report is produced for each core. For example:

```
&core=alfresco
&core=archive
```

You can also fix index issues, check the index cache and backup individual indexes by using JMX. The status of the index can be checked using the JMX client on the **JMX MBeans > Alfresco > solrIndexes > <store>** tabs. The default view is the Solr 4 core summary. The operations run the same consistency checks that are available by URL.

Solr 4 troubleshooting for SSL configurations

When you have an Alfresco installation that requires an SSL configuration, you might encounter connection issues.

If Solr 4 search and/or the Solr 4 tracking is not working properly, you might see this message on the Tomcat console:

```
Aug 22, 2011 8:19:21 PM org.apache.tomcat.util.net.jsse.JSSESupport handShake
WARNING: SSL server initiated renegotiation is disabled, closing connection
```

This message indicates that one side of the SSL connection is trying to renegotiate the SSL connection. This form of negotiation was found to be susceptible to man-in-the-middle attacks and it was disabled in the Java JSEE stack until a fix could be applied.

Refer to the following link for more information: <http://www.oracle.com/technetwork/java/javase/documentation/tlsreadme2-176330.html>.

Refer also to the following links: http://www.gremwell.com/enabling_ssl_tls_renegotiation_in_java and <http://tomcat.apache.org/tomcat-6.0-doc/config/http.html>.

If your version of Java does not have the fix, you need to re-enabled renegotiation by performing the following steps:

1. Add the `-Dsun.security.ssl.allowUnsafeRenegotiation=true` option to `JAVA_OPTS`.
2. Add the `allowUnsafeLegacyRenegotiation="true"` option to the Tomcat SSL connector.

Solr 4 backup and restore

This section describes the process for backing up and restoring the Solr 4 server.

Your backup strategy must be tested end-to-end, including restoration of backups that were taken previously. Ensure that you have adequately tested your backup scripts prior to deploying Alfresco to production.

Backing up Solr 4

There are a number of ways to back up the Solr 4 indexes.

You can set the Solr 4 indexes backup properties either by using the Admin Console in Share or by editing the `alfresco-global.properties` file or by using a JMX client, such as JConsole.

Set up Solr 4 backup properties using Share Admin Console

You can only see the Admin Console if you are an administrator user.

1. Launch the Admin Console. For information, see [Launching the Admin Console](#).
2. In the **Repository Services** section, click **Search Service**.
You see the **Search Service** page.
3. Scroll down to the **Backup Settings** section.

Backup Settings	
<p>Main Store</p> <p>Backup Location: <input type="text" value="\${dir.root}/solr4Backup/alfresco"/> <small>The location where the index backup is stored on the Solr server.</small></p> <p>Backup Cron Expression: <input type="text" value="0 0 2 * * ?"/> <small>A unix-like expression, using the same syntax as the cron command, that defines when backups occur. The default value is 0 0 2 * * ? meaning the backup is performed daily at 02.00.</small></p> <p>Backups To Keep: <input type="text" value="3"/> <small>The number of backups to keep (including the latest backup).</small></p>	<p>Archive Store Properties</p> <p>Backup Location: <input type="text" value="\${dir.root}/solr4Backup/archive"/> <small>The location where the index backup is stored on the Solr server.</small></p> <p>Backup Cron Expression: <input type="text" value="0 0 4 * * ?"/> <small>A unix-like expression, using the same syntax as the cron command, that defines when backups occur. The default value is 0 0 4 * * ? meaning the backup is performed daily at 04.00.</small></p> <p>Backups To Keep: <input type="text" value="3"/> <small>The number of backups to keep (including the latest backup).</small></p>

Here, you can specify the backup location and edit backup properties for each core of the Solr 4 index: **Main Store** and **Archive Store**.

- **Backup Cron Expression:** Specifies a Quartz cron expression that defines when backups occur. Solr 4 creates a timestamped sub-directory for each index back up you make.
 - **Backup Location:** Specifies the full-path location for the backup to be stored.
 - **Backups To Keep:** Specifies the maximum number of index backups that Solr 4 should store.
4. Click **Edit**.

5. Specify the full location path on the Solr 4 server file system to store the index backup in the **Backup Location** text box.
6. Click **Save**.

Specifying Solr 4 backup directory by using `alfresco-global.properties` file

This task shows how to specify the Solr 4 backup directory by using `alfresco-global.properties` file.

- To set the Solr 4 backup directory using the `alfresco-global.properties` file, set the value of the following properties to the full path where the backups should be kept:

```
solr4.backup.archive.remoteBackupLocation=  
solr4.backup.alfresco.remoteBackupLocation=
```



The values set on a sub-system will mean that the property values from configuration files may be ignored. Use the Share Admin Console or JMX client to set the backup location.

Back up Solr 4 indexes using JMX client

You can use the JMX client, JConsole to backup Solr 4 indexes, edit Solr 4 backup properties and setup the backup directory.

- You can set the backup of Solr 4 indexes using the JMX client, such as JConsole on the **JMX MBeans > Alfresco > Schedule > DEFAULT > MonitoredCronTrigger > search.alfrescoCoreBackupTrigger > Operations > executeNow** tab. The default view is the Solr 4 core summary. Alternatively, navigate to **MBeans > Alfresco > SolrIndexes > coreName > Operations > backUpIndex** tab. Type the directory name in the **remoteLocation** text box and click **backUpIndex**.
- Solr 4 backup properties can be edited using the JMX client on the **JMX MBeans > Alfresco > Configuration > Search > managed > solr > Attributes** tab. The default view is the Solr 4 core summary.
- To use JMX client to setup Solr 4 backup directory, navigate to **MBeans tab > Alfresco > Configuration > Search > managed > solr > Attributes** and change the values for `solr.backup.alfresco.remoteBackupLocation` and `solr.backup.archive.remoteBackupLocation` properties.

Restoring Solr 4 indexes

This section describes the process for restoring the Solr 4 indexes.

During the recovery process, Solr 4 queries Alfresco to update the indexes restored from a backup. To restore Solr 4 indexes, use the following steps:

1. Stop the Solr 4 server.
2. Copy a backup index to the data directory for each core.
Remember to use a backup created from the same Alfresco instance.
3. Restart the Solr 4 server. Solr 4 will start to track the indexes based on the state of the restored index.

Full text search configuration properties for Solr 4 index

The `repository.properties` file defines the properties that influence how all indexes behave. This topic describes the Solr 4 index's full text search properties, which are contained in the `repository.properties` file.

The main index and deltas all use the same configuration. The data dictionary settings for properties determine how individual properties are indexed.



The following properties are set in the `repository.properties` file. However, if you wish to change them, we recommend that you add the relevant property to the `alfresco-global.properties` file and then make the changes. No changes should be done in the `repository.properties` file.

Solr 4 index properties

solr.host=localhost

The host name where the Solr instance is located.

solr.port=8080

The port number on which the Solr instance is running.

solr.port.ssl=8443

The port number on which the Solr SSL support is running.

solr.solrUser=solr

The Solr user name.

solr.solrPassword=solr

The Solr password.

solr.secureComms=https

The HTTPS connection.

solr.solrConnectTimeout=5000

The Solr connection timeouts in ms.

solr.solrPingCronExpression=0 0/5 * * * ? *

The cron expression defining how often the Solr Admin client (used by JMX) pings Solr 4 if it goes away.

Data dictionary options

The indexing behavior of each property can be set in the content model. By default, they are indexed atomically. The property value is not stored in the index, and the property is tokenized when it is indexed.

The following example shows how indexing can be controlled.

Enabled="false"

If this is false, there will be no entry for this property in the index.

Atomic="true"

If this is true, the property is indexed in the transaction, if not the property is indexed in the background.

facetable="true"

If true, the property will be used for faceting and if false, you cannot use it for faceting.

Tokenised="true"

If "true", the string value of the property is tokenized before indexing.

if "false", it is indexed "as is" as a single string.

if "both" then both specified forms are in the index.

The tokenizer is determined by the property type in the data dictionary. This is locale sensitive as supported by the data dictionary, so you could switch to tokenize all your content in German. At the moment you cannot mix German and English tokenization.

```
<type name="cm:content">
  <title>Content</title>
  <parent>cm:cobject</parent>
  <properties>
```

```

    <property name="cm:content">
      <type>d:content</type>
      <mandatory>>false</mandatory>
      <index enabled="true">
        <facetable>>true</facetable>
        <atomic>>false</atomic>
        <tokenised>>true</tokenised>
      </index>
    </property>
  </properties>
</type>

```

Indexing defaults

The effective indexing defaults for all properties are as follows:

```

<index enabled="true">
  <atomic>>true</atomic>
  <stored>>false</stored>
  <tokenised>>true</tokenised>
</index>
...

```

Using Filtered search

This section gives an overview on the application of filtered search capability in Alfresco Share along with its configuration details. It also describes how to define your own custom filters.

Filtered search within Alfresco is a powerful search feature that allows users to filter and customize their results by applying multiple filters to their search results in a navigational way. Filtered search breaks up search results into multiple categories, typically showing counts for each, and allows the user to drill down or further restrict their search results based on those filters.

 Filtered search uses the Solr 4 search subsystem and is enabled in Alfresco One 5.0, by default. For more information on migrating from your existing search subsystem to Solr 4, see the [Solr 4 migration documentation](#).

Configuring filtered search

You can configure filtered search either by using the [configuration files](#) or by using the [Share Search Manager](#).

Filtered search configuration file and default properties

There are a number of default filtered search configuration properties defined. This topic lists the default filtered search properties along with a description.

The following example shows how the Alfresco default filters are defined:

```

#
# Alfresco default facets
# Note: If you have changed the filter's default value(s) via Share, then any
# subsequent changes of those default values won't be applied to the filter on
# server startup.
#
# Field-Facet-Qname => cm:content.mimetype
default.cm\:content.mimetype.filterID=filter_mimetype
default.cm\:content.mimetype.displayName=faceted-search.facet-
menu.facet.formats
default.cm\:content.mimetype.displayControl=alfresco/search/FacetFilters
default.cm\:content.mimetype.maxFilters=5
default.cm\:content.mimetype.hitThreshold=1
default.cm\:content.mimetype.minFilterValueLength=4
default.cm\:content.mimetype.sortBy=DESCENDING

```

```

default.cm\:content.mimetype.scope=ALL
default.cm\:content.mimetype.scopedSites=
default.cm\:content.mimetype.isEnabled=true

# Field-Facet-Qname => cm:creator
default.cm\:creator.filterID=filter_creator
default.cm\:creator.displayName=faceted-search.facet-menu.facet.creator
default.cm\:creator.displayControl=alfresco/search/FacetFilters
default.cm\:creator.maxFilters=5
default.cm\:creator.hitThreshold=1
default.cm\:creator.minFilterValueLength=4
default.cm\:creator.sortBy=ALPHABETICALLY
default.cm\:creator.scope=ALL
default.cm\:creator.scopedSites=
default.cm\:creator.isEnabled=true

# Field-Facet-Qname => cm:modifier
default.cm\:modifier.filterID=filter_modifier
default.cm\:modifier.displayName=faceted-search.facet-menu.facet.modifier
default.cm\:modifier.displayControl=alfresco/search/FacetFilters
default.cm\:modifier.maxFilters=5
default.cm\:modifier.hitThreshold=1
default.cm\:modifier.minFilterValueLength=4
default.cm\:modifier.sortBy=ALPHABETICALLY
default.cm\:modifier.scope=ALL
default.cm\:modifier.scopedSites=
default.cm\:modifier.isEnabled=true

# Field-Facet-Qname => cm:created
default.cm\:created.filterID=filter_created
default.cm\:created.displayName=faceted-search.facet-menu.facet.created
default.cm\:created.displayControl=alfresco/search/FacetFilters
default.cm\:created.maxFilters=5
default.cm\:created.hitThreshold=1
default.cm\:created.minFilterValueLength=4
default.cm\:created.sortBy=INDEX
default.cm\:created.scope=ALL
default.cm\:created.scopedSites=
default.cm\:created.isEnabled=true

# Field-Facet-Qname => cm:modified
default.cm\:modified.filterID=filter_modified
default.cm\:modified.displayName=faceted-search.facet-menu.facet.modified
default.cm\:modified.displayControl=alfresco/search/FacetFilters
default.cm\:modified.maxFilters=5
default.cm\:modified.hitThreshold=1
default.cm\:modified.minFilterValueLength=4
default.cm\:modified.sortBy=INDEX
default.cm\:modified.scope=ALL
default.cm\:modified.scopedSites=
default.cm\:modified.isEnabled=true

# Field-Facet-Qname => cm:content.size
default.cm\:content.size.filterID=filter_content_size
default.cm\:content.size.displayName=faceted-search.facet-menu.facet.size
default.cm\:content.size.displayControl=alfresco/search/FacetFilters
default.cm\:content.size.maxFilters=5
default.cm\:content.size.hitThreshold=1
default.cm\:content.size.minFilterValueLength=4
default.cm\:content.size.sortBy=INDEX
default.cm\:content.size.scope=ALL
default.cm\:content.size.scopedSites=
default.cm\:content.size.isEnabled=true

```

Filter property description

An example of a filter is `cm:modified`. It specifies the name of the filter field. It is the field on which you want to do a filtered search.

filterID

Specifies a unique name to identify the filter. Before adding a new filter, check the existing filters (via [Search Manager](#)) to ensure that the `filterID` does not already exist.

displayName

Specifies the display name of the filter.

displayControl

Enables the user to decide the user interface control or how the filter is displayed on the **Search** page. The default option is **Check box**.

maxFilters

Enables the user to select the maximum number of filters shown for search results. You can select to show more than one filter.

hitThreshold

Enables the user to select the minimum number of matches a filter result must have to be shown on the **Search** page.

minFilterValueLength

Specifies the minimum length of characters that a filter value must have to be displayed. This can be useful in hiding common short words.

sortBy

Enables the user to select the order in which the filter results must be shown on the **Search** page.

scope

Enables the user to select the sites where the filter will be available.

scopedSites

Displays a list of sites where the filter will be available.

isEnabled

Specifies if the filter is enabled for inclusion on the search results page. Disabled filters are not displayed. Only the filters you create via Share console can be deleted; default filters must be disabled to hide them.



You cannot delete or modify any of the default filters, however you can disable them. To define your own custom filters, see [Defining custom search filters](#).

Defining custom search filters using configuration file

This topic describes how you can define and create your own custom filters for being displayed on the search result page.

You can define custom filters in the `solr-facets-config-custom.properties` file. You can also use this file to override the default filter properties.

1. Navigate to the `<classpathRoot>/alfresco/extension` directory.
2. Create the `solr-facets-config-custom.properties` file.
3. Open the `solr-facets-config-custom.properties` file and specify your custom filter properties.

Here's an example of custom filter configuration:

```
custom.cm\:description.filterID=filter_newFilter
custom.cm\:description.displayName=faceted-search.facet-
menu.facet.description
custom.cm\:description.displayControl=alfresco/search/FacetFilters
custom.cm\:description.maxFilters=3
custom.cm\:description.hitThreshold=1
custom.cm\:description.minFilterValueLength=2
custom.cm\:description.sortBy=DESCENDING
custom.cm\:description.scope=SCOPED_SITES
```

```
custom.cm\:description.scopedSites=
custom.cm\:description.isEnabled=true
```

-  The values specified in the custom filters will overwrite the default filter's value. However, if you change the filter's default value(s) via Share, then any subsequent changes made to the filter values via the configuration files, won't be applied to the filter on server startup.

Setting Solr 4 log4j values

You can set different debug logging levels for Alfresco-Solr 4 components using the Solr 4 log4j properties.

1. Locate the `<solrRootDir>/log4j-solr.properties` file.
2. Edit it to add your required logging setting. For example:


```
log4j.logger.org.alfresco.solr.tracker.CoreTracker=DEBUG
```
3. Changes to the `log4j-solr.properties` file will be re-read by Solr 4 when it starts up. If you need to make changes to the logging level while the system is running, going to the following URL (either in a browser or for example, using curl) will cause Solr 4 to re-load the properties file.

```
https://<solrHostName>:<solrPort>/solr4/admin/cores?
action=LOG4J&resource=log4j-solr.properties
```

Calculate the memory needed for Solr 4 nodes

Solr 4 can have high memory requirements. You can use a formula to calculate the memory needed for the Alfresco internal data structures used in Solr 4 for PATH queries and read permission enforcement.

By default, there are two cores in Solr 4: `WorkspaceSpacesStore` and `ArchiveSpacesStore`. Normally, each core has one searcher but can have a maximum of two searchers.

In the calculation:

- N = refers to the number of nodes in the store. Each core's value is calculated separately. If there are more than two cores, you will need to add additional queries to calculate the value for that core (as shown in the example code block).
- T = refers to the number of transactions in the repository and this is same for each core
- A = refers to the number of ACLs in the repository and this is same for each core
- X = refers to the number of ACL transactions in the repository and this is same for each core

The values for N, T, A and X come from the database. Use the following commands to derive these values:

```
select * from
(select count( * ) N_Alfresco from alf_node where store_id = (select id from
alf_store where protocol = 'workspace' and identifier = 'SpacesStore')) as
N1 ,
(select count( * ) N_Archive from alf_node where store_id = (select id from
alf_store where protocol = 'archive' and identifier = 'SpacesStore')) as N2 ,
(select count( * ) T from alf_transaction ) as T,
(select count( * ) A from alf_access_control_list ) as A,
(select count( * ) X from alf_acl_change_set ) as X;
```

For example, if there are three cores, include additional queries to calculate the value for that core, as shown:

```
select * from
(select count( * ) N_Alfresco from alf_node where store_id = (select id from
alf_store where protocol = 'workspace' and identifier = 'SpacesStore')) as
N1 ,
```

```
(select count( * ) N_Archive from alf_node where store_id = (select id from
alf_store where protocol = 'archive' and identifier = 'SpacesStore')) as N2 ,
(select count( * ) N_Version2 from alf_node where store_id = (select id from
alf_store where protocol = 'workspace' and identifier = 'version2Store'))as
N3 ,
(select count( * ) T from alf_transaction ) as T,
(select count( * ) A from alf_access_control_list ) as A,
(select count( * ) X from alf_acl_change_set) as X;
```

Memory calculation for the Alfresco data structures associated with one searcher

For a store containing 100 MB nodes, 100 MB transactions, 100 MB ACLs and 100 MB ACL transactions, 20.1 GB of memory is needed. Assuming there are not many ACLs or ACL changes, for 100 MB nodes, you will need 12 GB to 16 GB of memory depending on the number of transactions. This calculation is based on the following formula: $120N + 32(T + A + X)$ bytes.

Memory calculation for the Solr 4 caches associated with one searcher

The Solr 4 cache will use up to $(2N + T + A + X)/8$ bytes for an entry in any cache.

The formula to calculate the total memory needed for the caches for a single core is:

```
(solr.filterCache.size + solr.queryResultCache.size + solr.authorityCache.size
+ solr.pathCache.size) * (2N + T + A + X)/8 bytes
```

So, for 100M documents and 100M transactions, 150G of memory is needed using the out of box configuration.

```
(512 + 1024 + 512 + 512)(500M)/8 = 150G
```

The default cache values needs to change to accommodate a large repository. So, for 100M documents, 100M transactions and reduced cache size, 19G of memory is needed.

```
(64 + 128 + 64 + 64)(500M)/8 = 19G
```

Solr 4 memory planning

For the Alfresco JVM, the most important parameter is `-Xmx`, which controls the heap. The specified formula helps to evaluate the memory required by Solr 4 and for capacity planning. Solr 4 memory requirements increase with the size of the repository but also with the amount of memory you allocate to the Solr 4 caches. Decreasing the Solr 4 cache parameters can dramatically lower the memory requirements, with the drawback of hitting the disk more often. You can set these parameters to different values for the each of the stores.

```
solr.filterCache.size
solr.queryResultCache.size
solr.authorityCache.size
solr.pathCache.size
```

Overall Solr 4 memory use

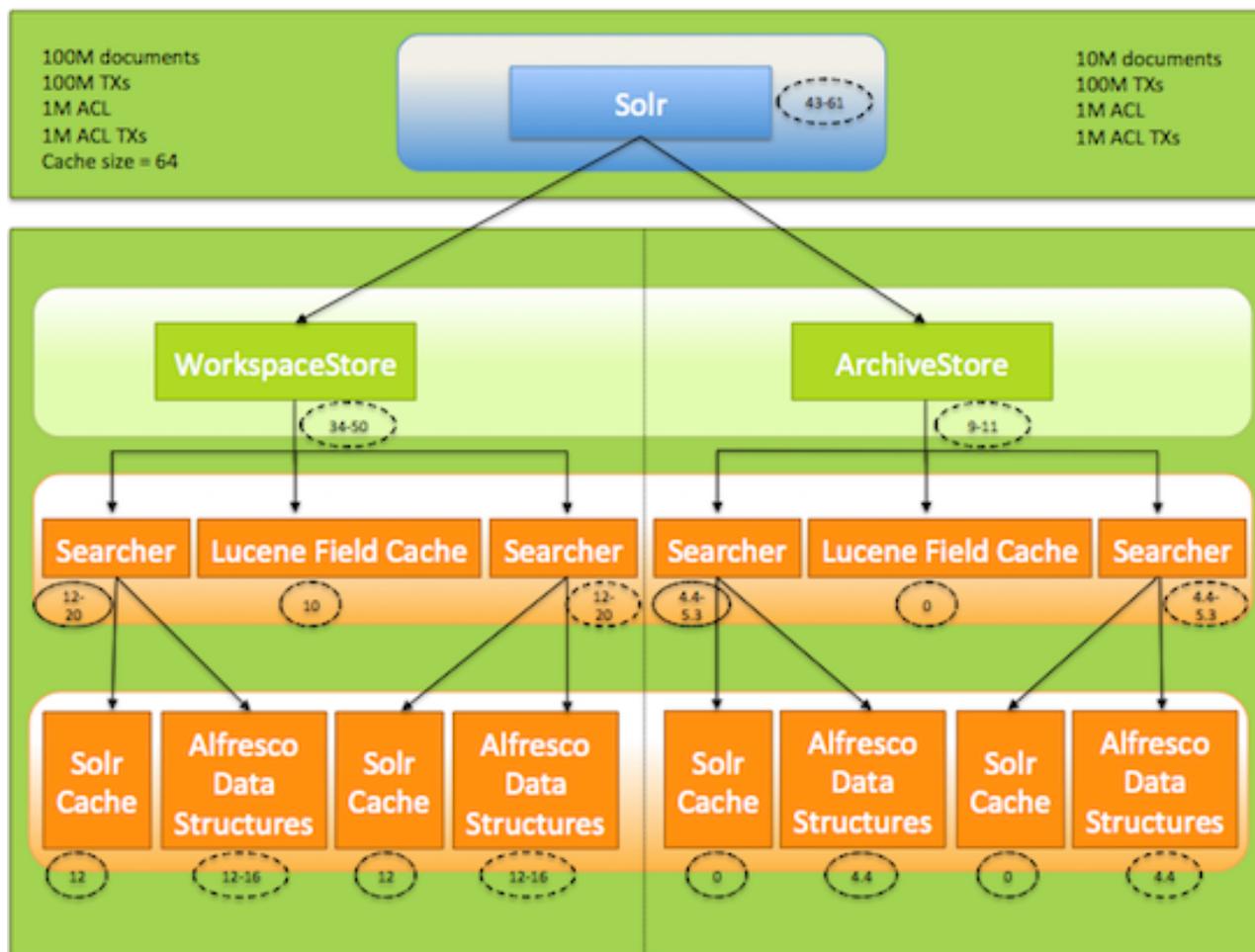
This example is based on the data given.

For WorkspaceStore: Assuming that there are 100 MB docs, 100 MB TXs, 1 MB ACLs and ACL TXs, cache size of 64 entries each for FilterCache, AuthorityCache and QCache, and 128 entries for PathCache, between 12 GB to 20 GB of memory is needed per searcher. Normally, there is one searcher live but around commit time there can be two searchers. So, approximately 34 GB to 50 GB of memory will be needed in total.

For Archivistore: Assuming that there are 100 MB transactions, 10 MB docs and all caches are tuned down, between 4.4 GB to 5.3 GB of memory is needed per searcher. Total memory needed for both the searchers will be between 9 GB to 11 GB.

So, the total memory requirement for both the cores is between 43 GB to 61 GB.

The following diagram shows the overall memory use for a Solr 4 node as explained in the example:



Minimize the memory requirements for Solr 4 nodes

- Reduce the cache sizes and check the cache hit rate.
- Disable ACL checks using `alfresco.doPermissionChecks=false`
- Disable archive indexing, if you are not using it.
- Check the number of empty transactions. If there are many empty transactions, purge the transactions from Alfresco using the `action=FIX` action.
- Find the exact number of nodes in the store (N), exact number of transactions in the repository (T), number of ACLs (A) and related ACL transactions in the repository (X).
- Since everything scales to the number of documents in the index, add the Index control aspect to the documents you do not want in the index.

Transactional metadata query

This section provides an overview on the transactional metadata query. It describes its features and instructions on configuring it. This section also describes the process of configuring the optional patch for upgrade.

Overview of transactional metadata query

Alfresco supports the execution of a subset of the CMIS Query Language (CMIS QL) and Alfresco Full Text Search (AFTS) queries directly against the database. This feature is called transactional metadata query. This section provides an overview on the transactional metadata query.

Prior to Alfresco One 4.2, the Solr search subsystem does not support transactional indexing. The Solr subsystem is eventually consistent. A change can take anytime to be reflected in the index, ranging from a few seconds to several minutes. Solr indexes the metadata and the content of each updated node, in the order in which the nodes were last changed. The rate at which the nodes are indexed is mainly determined by the time it takes to transform the content and the rate at which the nodes are being changed.

Features of transactional metadata query

This topic describes the features of the transactional metadata query.

- Transactional metadata query is supported for both Solr 4 and noindex search subsystems.
- When you enable transactional metadata queries, a query is parsed to check if all of its parts are supported by the database-based query engine. If yes, the database is used automatically.
- Using the database gives transactional consistency as opposed to the eventual consistency provided by Solr 4.
- Normally, a query will be executed against the database, if possible. Database execution of a query depends on the query itself. It also depends on the application of an optional patch to the database, which creates the required supporting database indexes. If the supporting indexes have been created, each index subsystem can be configured to:
 - perform transactional execution of queries;
 - execute queries transactionally, when possible, and fall back to eventual consistency; or
 - always execute eventual consistency.
- The `SearchParameters` and `QueryOptions` objects can be used to override this behaviour per query.

Transactional metadata queries supported by database

This topic provides information on the queries supported by the database.

CMIS QL

The following object types and their sub-types are supported:

- `cmis:document`

For example:

```
select * from cmis:document
```

- `cmis:folder`

For example:

```
select * from cmis:folder
```

- Alfresco aspects

For example:

```
select * from cm:dublincore
```

CMIS property data types

The `WHERE` and `ORDER BY` clauses support the following property data types and comparisons:

- `string`
 - Supports all properties and comparisons, such as `=`, `<>`, `<`, `<=`, `>=`, `>`, `IN`, `NOT IN`, `LIKE`

- Supports ordering for single-valued properties

For example:

```
select * from cmis:document where cmis:name <> 'fred' order by cmis:name
```

- integer
 - Supports all properties and comparisons, such as =, <>, <, <=, >=, >, IN, NOT IN
 - Supports ordering for single-valued properties
- id
 - Supports cmis:objectId, cmis:baseTypeId, cmis:objectTypeId, cmis:parentId, =, <>, IN, NOT IN
 - Ordering using a property, which is a CMIS identifier, is not supported.
- datetime
 - Supports all properties and comparisons =, <>, <, <=, >=, >, IN, NOT IN
 - Supports ordering for single-valued properties

For example:

```
select * from cmis:document where cmis:lastModificationDate =
'2010-04-01T12:15:00.000Z' order by
cmis:creationDate ASC
```



While the CMIS Decimal, Boolean and URI data types are not supported, the multi-valued properties and the multi-valued predicates as defined in the CMIS specification are supported. For example,

```
select * from ext:doc where 'test' = ANY ext:multiValuedStringProperty
```

Supported predicates

A predicate specifies a condition that is true or false about a given row or group. The following predicates are supported:

- Comparison predicates, such as =, <>, <, <=, >=, >, <>
- IN predicate
- LIKE predicate



Prefixed expressions perform better and should be used where possible.

- NULL predicate
- Quantified comparison predicate (= ANY)
- Quantified IN predicate (ANY IN (. . . .))
- IN_FOLDER predicate function

Unsupported predicates

The following predicates are not supported:

- TEXT search predicate, such as CONTAINS() and SCORE()
- IN_TREE() predicate

Supported logical operators

The following logical operators are supported:

- AND

- NOT

Unsupported logical operators

The following logical operator is not supported:

- OR

Other operators

In the following cases, the query will go to the database but the result might not be as expected. In all other unsupported cases, the database query will fail and fall back to be executed against the Solr 4 subsystem.

- `IS NOT NULL`
- `IS NULL`: Currently, this operator will only find properties that are explicitly NULL as opposed to the property not existing.
- `SORT`: The multi-valued and `mltext` properties will sort according to one of the values. Ordering is not localized and relies on the database collation. It uses an `INNER JOIN`, which will also filter NULL values from the result set.
- `d:mltext`: This data type ignores locale. However, if there is more than one locale, the localised values behave as a multi-valued string. Ordering on `mltext` will be undefined as it is effectively multi-valued.
- `UPPER()` and `LOWER()`: Comparison predicates provide additional support for SQL `UPPER()` and `LOWER()` functions (that were dropped from a draft version of Alfresco CMIS specification but are supported for backward compatibility).

Configuring transactional metadata query

This topic describes how to configure the transaction metadata query using the subsystem properties.

The common properties used to configure the transactional metadata query for the search subsystems are:

- `solr.query.cmis.queryConsistency`
- `solr.query.fts.queryConsistency`

These properties should be set in the `alfresco-global.properties` file.

The default value for these properties is `TRANSACTIONAL_IF_POSSIBLE`. However, you can override it with any of the following permitted values:

- `EVENTUAL`
- `TRANSACTIONAL`

The `solr.query.cmis.queryConsistency` and `solr.query.fts.queryConsistency` properties can also be set per query on the `SearchParameters` and `QueryOptions` objects.

Configuring an optional patch for upgrade

This topic describes how to configure an optional patch for upgrade.

To enable the patch that adds the required indexes to the database, set the following property in the `alfresco-global.properties` file :

```
system.metadata-query-indexes.ignored=false
```

If this patch has not been run, the metadata query will not be used, regardless of the configuration. This configuration is checked when the subsystem is reloaded.

For a new install, the default behaviour is to use the `TRANSACTIONAL_IF_POSSIBLE` metadata queries. For an upgraded system, the `TRANSACTIONAL_IF_POSSIBLE` metadata queries will be used only if the upgrade patch has been run.

Adding optional indexes to database

When you are upgrading the database, you can add optional indexes in order to support the metadata query feature. This information lets you know the likely duration of the upgrade and how to do it incrementally.

For large repositories, creating the database indexes to support the transactional metadata query can take some time. To check how long it will take, you can add the first index to the database and note the time taken. The full upgrade is estimated to take less than 10 times this value. However, this can vary depending on the structure of the data, the database, and the size of the repository.

The [SQL patch script](#) can be run in parts, adding one index at a time. The patch is marked complete by the statement that inserts into `alf_applied_patch`. The patch can be marked as unapplied using the SQL delete statement.

Setting up Enterprise to Cloud Sync

Enterprise to Cloud Sync gives Alfresco on-premise users the ability to synchronize their content to Alfresco in the cloud. This feature supports scenarios where users wish to collaborate on documents with external parties that do not have access to systems behind the firewall. In these circumstances, the on-premise Alfresco instance becomes the system of record, and the cloud instance is the system of engagement for external collaboration.

Once content has been setup to synchronize, the cloud and on-premise instances of the documents are automatically synchronized with each other whenever either version is updated.

It is worth noting the following recommendations for using Enterprise to Cloud Sync:

- Use Enterprise to Cloud Sync only on content that you wish to share with other users
- Ensure that you do not set up synchronization on content that is sensitive
- Remember that other users of the network might have access to your synced content

To set up Enterprise to Cloud Sync, you need an Alfresco in the cloud account.

 Enterprise to Cloud Sync is not supported with a multi-tenant on-premise Alfresco instance.

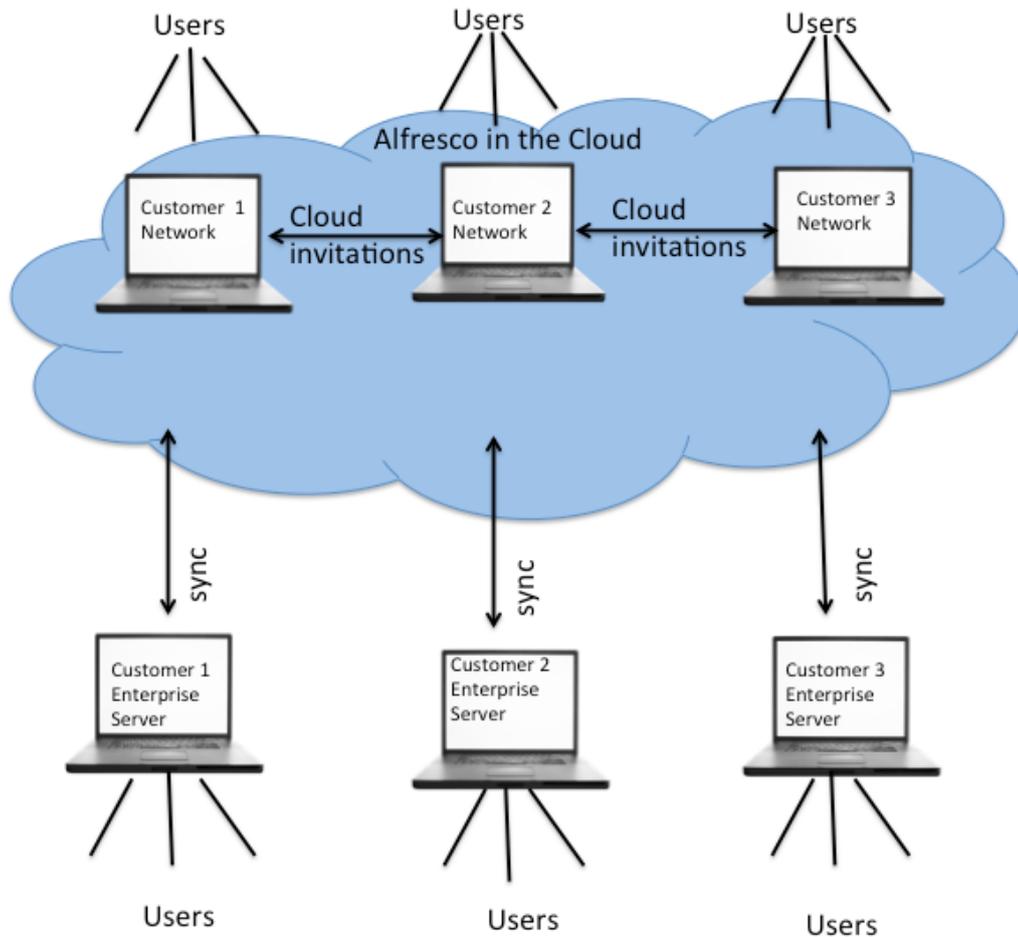
Enterprise to Cloud Sync overview

Enterprise to Cloud Sync allows you to select content that is automatically synchronized between on-premise Alfresco and a network on Alfresco in the cloud.

The Enterprise to Cloud Sync capabilities include:

- Synchronization of individual and multiple files, folders, and folder hierarchies between on-premise and Alfresco in the cloud
- Content and common metadata is included within synchronized payloads
- Automatic synchronization
- Secure exchange of information over HTTPS connection
- All actions initiated on the on-premise Alfresco instance
- Choice over what can be synchronized to ensure sensitive content remains on-premise

The following diagram shows the synchronization flow for Enterprise to Cloud Sync.



Content is automatically synchronized between on-premise Alfresco repositories and the cloud instance. This ensures that the on-premise system is in sync with any changes.

Other Alfresco in the cloud users can access the content within the same network. Alfresco in the cloud users can also send invitations to other cloud users to join their network and share the synchronized content.

Configuring Enterprise to Cloud Sync

This section describes how to enable and disable Enterprise to Cloud Sync.

Enabling Enterprise to Cloud Sync

To enable synchronization from your on-premise Alfresco server, you need an Enterprise Alfresco subscription and an Alfresco license that contains the synchronization feature.

Ensure that you have access to port 443 and that you are able to access https.

Enterprise to Cloud Sync relies on the Audit feature. In an Alfresco installation, auditing is enabled by default but if you set the `audit.enabled=` property to false, synchronization will stop working.

1. Copy the license file to the Alfresco installation directory.
The license file has a file extension of `.lic`.
2. Launch the Alfresco Admin Console.
For more information on the Admin Console, see [Admin Console Quick Guide](#).

3. Click **Apply New License**.

You have now applied the license and this will automatically enable the Enterprise to Cloud Sync feature.

 You may need to restart the Alfresco server for the license to be applied.

4. Verify that you have successfully enabled Enterprise to Cloud Sync.

- a. **Sync to Cloud** action is available for documents and folders in the Alfresco Share Document Library.
- b. Make sure that the log contains the following message:

```
2012-09-04 13:38:50,458 INFO [repo.sync.SyncAdminServiceImpl] [main]
A key is provided for cloud sync
```

To start using Enterprise to Cloud Sync, you need to set up synchronization in your on-premise Alfresco. For more information on setting up Enterprise to Cloud Sync, see [Setting up Enterprise to Cloud Sync](#).

Disabling Enterprise to Cloud Sync

If your Alfresco license contains Enterprise to Cloud Sync access, then synchronization is enabled by default when you apply the license. You have the option to disable this feature if you prefer not to make it available to your users.

1. Open the `<classpathRoot>/alfresco-global.properties` file.
2. Add the `sync.mode=OFF` property, and then save the file.
3. Restart the Alfresco server.

Enterprise to Cloud Sync on replica on-premise instances

Alfresco does not support Enterprise to Cloud Sync running simultaneously from identical Alfresco Enterprise instances to Alfresco Cloud.

If you are backing up or restoring a replica of the Alfresco repository that has Enterprise to Cloud Sync enabled to my.alfresco.com, it will result in sync issues for the original and the replica on-premise environments, and it will potentially cause functional issues and data loss.

To prevent this issue, before start up, disable Enterprise to Cloud Sync in the replica on-premise server by adding the following properties to the `alfresco-global.properties` file:

```
sync.mode=OFF
sync.pushJob.enabled=false
sync.pullJob.enabled=false
```

To prevent the cloned repositories from syncing to the cloud, enable Enterprise to Cloud Sync only on the main production system by setting the following property in the `alfresco-global.properties` file:

```
system.serverMode=PRODUCTION
```

The default value for this property is `UNKNOWN`.

 If the `system.serverMode` is not set to `PRODUCTION` on your main production system, Enterprise to Cloud Sync will not work.

If you currently have more than one Alfresco Enterprise on premise instance using Sync against Cloud, Alfresco recommends that you open an Alfresco Support ticket for assistance.

Troubleshooting Enterprise to Cloud Sync

This section lists the possible error messages and solutions for troubleshooting unexpected behavior.

Enterprise to Cloud Sync error messages

Use the following list of common error messages for troubleshooting the possible solutions.

End user error message	Description	Possible causes	Solutions
Could not create sync	Authorized account is not suitable for synchronization	The network you are trying to authorize is not a standard, enterprise or partner network.	Create and authorize a standard, enterprise or partner network.
Unable to connect to the sync server	Unable to connect to Alfresco Cloud.	The server on which Alfresco Cloud is running is disconnected.	Connect to the synchronization server. If the server is down, contact Alfresco Cloud support.
Could not remove sync	Unable to remove synchronization	Files within the synchronized folder or its subfolders are locked for editing in Alfresco Cloud.	Make sure that all files within the synchronized folder and its subfolders are unlocked for editing in Alfresco Cloud.
Could not request sync	Unable to put in a request to synchronize content from on-premise cloud into Alfresco Cloud.	Your on-premise Alfresco, cannot communicate with the Alfresco Cloud.	Make sure that your on-premise Alfresco is up and running and that you are logged in to it. Make sure that Alfresco Cloud is up and running.
A node already exists in the target folder with the same name	Unable to synchronize content between Alfresco Cloud and Alfresco Share on-premise	The node with the same name exists on Alfresco Cloud.	Rename the on-premise node and try syncing again or delete the node with the same name from Alfresco Cloud and try synchronizing again.
Target folder could not be found	The folder specified as the target folder does not exist in Alfresco Cloud.	The folder specified as the target folder for the synchronization does not exist on Alfresco Cloud.	Specify a different target folder on Alfresco Cloud or create a new folder with a matching name.
Content cannot be created, it is already synchronized from somewhere else	Content with the same name cannot be synchronized twice to the same location in Alfresco Cloud	Different users are trying to synchronize content item at the same time to the same Alfresco Cloud target location.	Synchronize content to a different location.
Content has already been synchronized from somewhere else	A content item can be synchronized only once and to one location in Alfresco Cloud.	The content item that you are trying to synchronize, has already been synchronized	Make sure that the content item does not exist anywhere else on Alfresco Cloud or that the content item has not already been synchronized.

End user error message	Description	Possible causes	Solutions
Content no longer exists on the remote system	Unable to synchronize content as the content item no longer exists on cloud.	Content does not exist on Alfresco Cloud.	Make sure that the content item exists on Alfresco Cloud.
Content can not be updated, access denied	Unable to update content on Alfresco Cloud.	The user does not have permission to update content on Alfresco Cloud	Make sure that the user has the correct permissions to update content.
Content size violation (limit exceeded)	Unable to synchronize content on to Alfresco Cloud	The user has exceed the allocated content size limit for an individual file on a network.	Try to reduce the size of the content item, if that is not possible, contact Alfresco Support to request an increase to content size limit for individual file for your cloud Network.
Quota violation (limit exceeded)	Unable to synchronize content on to Alfresco Cloud	User has exceeded the allocated quota of storage space on cloud	Try to reduce the size of the content item and/or empty your trashcan by using the Account settings. If that is not possible, contact Alfresco Support to request an increase to the overall storage space quota for your cloud Network.
Unable to push changes for this node. The authentication details are no longer valid.	Unable to make any changes to the content on to this node in Alfresco Cloud.	The user has not provided valid authentication details	Make sure that the user has valid authentication details to gain access to the cloud.
Unable to push changes for this node. The owner no longer exists.	Unable to make any changes to the content on to this node in Alfresco Cloud.	The owner of this node no longer exists	Unsynchronize the content.
No network is enabled for sync	No network is enabled for synchronization	The user has not set the correct URL for Alfresco Cloud in the <code>alfresco-global.properties</code> file.	Set a valid URL for Alfresco Cloud in the <code>alfresco-global.properties</code> file and run on-premise Alfresco again.

Enterprise to Cloud Sync frequently asked questions

This section lists the FAQs about the Enterprise to Cloud Sync feature.

Why can't I synchronize my content?

Synchronization problems in Enterprise to Cloud Sync can be caused by any of the following issues:

- You are logged on to the wrong cloud network type
- You do not have network access to cloud
- You do not have a valid on-premise license key
- Your new license key is not sync-enabled

- Your global cloud property has a wrong value

To troubleshoot these issues, use the following steps to check whether the issue is resolved. If the issue is not resolved, continue to the next item in the list.

1. Check that the cloud network type is **Enterprise**: Log into Alfresco in the cloud and check your network type.
2. Check that you have network access to Alfresco in the cloud: There might be a communication problem either on the Alfresco in the cloud side or with your on-premise instance.
3. Check that your on-premise license key has not expired.
4. Check that the `sync-mode` property is set to `ON_PREMISE`: You will not be able to synchronize if this property has a different value. The default value for this property is `ON_PREMISE` and you do not need to change it.
5. Check that the value for the global cloud property `sync.cloud.url` is set to `https://a.alfresco.me/alfresco/a/{network}/`. This is the default value for this property and you do not need to change it. The format of this property is:
`sync.cloud.url=https://a.alfresco.me/alfresco/a/{network}/`

How do I know if my content has only partially synchronized?

The following list shows the error messages that you might see and their possible meanings.

This file exceeds the content limit.

The file is too large to perform the action.

You have exceeded the content quota.

There is not enough free space to perform the action.

Managing transformations

Standard transform options

There are many file types (also known as MIME types) available in Alfresco and it's not always possible to transform one file type to another.

Most images can be transformed to most other image types, but you can never transform audio or video files. The tables give details of registered file types with information about their available transform options.

If you have installed a transform tool, such as Transformation Server, there are additional transform options, which are listed in [Additional transform options](#) on page 284.

You can also view more information about file types and the proxies used to transform them by using the browser command:

```
localhost:8080/alfresco/service/mimetypes?mimetype=*
```

where `localhost:8080` is the host and port number of your active Alfresco instance.

application/acp and application/dita+xml - acp, dita

These formats cannot be transformed into, or generated from, any other format.

application/eps - eps

Format	Transformable to:	Transformable from:
application/illustrator		X
application/vnd.msword		X

Format	Transformable to:	Transformable from:
application/pdf		X
application/rtf		X
application/vnd.apple.keynote		X
application/vnd.apple.numbers		X
application/vnd.apple.pages		X
application/vnd.ms-excel		X
application/vnd.ms-excel.addin.macroenabled.12		X
application/vnd.ms-excel.sheet.binary.macroenabled.12		X
application/vnd.ms-excel.sheet.macroenabled.12		X
application/vnd.ms-excel.template.macroenabled.12		X
application/vnd.ms-outlook		X
application/vnd.ms-powerpoint		X
application/vnd.ms-powerpoint.addin.macroenabled.12		X
application/vnd.ms-powerpoint.presentation.macroenabled.12		X
application/vnd.ms-powerpoint.slide.macroenabled.12		X
application/vnd.ms-powerpoint.slideshow.macroenabled.12		X
application/vnd.ms-powerpoint.template.macroenabled.12		X
application/vnd.ms-word.document.macroenabled.12		X
application/vnd.ms-word.template.macroenabled.12		X
application/vnd.oasis.opendocument.presentation		X
application/vnd.oasis.opendocument.presentation-template		X
application/vnd.oasis.opendocument.spreadsheet		X
application/vnd.oasis.opendocument.spreadsheet-template		X
application/vnd.oasis.opendocument.text		X
application/vnd.oasis.opendocument.text-template		X
application/vnd.openxmlformats-officedocument.presentationml.presentation		X
application/vnd.openxmlformats-officedocument.presentationml.slide		X
application/vnd.openxmlformats-officedocument.presentationml.slideshow		X
application/vnd.openxmlformats-officedocument.presentationml.template		X
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet		X
application/vnd.openxmlformats-officedocument.spreadsheetml.template		X
application/vnd.openxmlformats-officedocument.wordprocessingml.document		X
application/vnd.openxmlformats-officedocument.wordprocessingml.template		X
application/vnd.sun.xml.calc		X
application/vnd.sun.xml.calc.template		X
application/vnd.sun.xml.impress		X
application/vnd.sun.xml.impress.template		X
application/vnd.sun.xml.writer		X
application/vnd.sun.xml.writer.template		X

Format	Transformable to:	Transformable from:
application/vnd.visio		X
application/wordperfect		X
image/bmp	X	X
image/cgm	X	X
image/gif	X	X
image/ief	X	X
image/jp2	X	X
image/jpeg	X	X
image/png	X	X
image/tiff	X	X
image/vnd.adobe.photoshop	X	X
image/vnd.adobe.premiere	X	X
image/x-cmu-raster	X	X
image/x-dwt	X	X
image/x-portable-anymap	X	X
image/x-portable-bitmap	X	X
image/x-portable-graymap	X	X
image/x-portable-pixmap	X	X
image/x-raw-adobe	X	X
image/x-raw-canon	X	X
image/x-raw-fuji	X	X
image/x-raw-hasselblad	X	X
image/x-raw-kodak	X	X
image/x-raw-leica	X	X
image/x-raw-minolta	X	X
image/x-raw-nikon	X	X
image/x-raw-olympus	X	X
image/x-raw-panasonic	X	X
image/x-raw-pentax	X	X
image/x-raw-red	X	X
image/x-raw-sigma	X	X
image/x-raw-sony	X	X
image/x-xbitmap	X	X
image/x-xpixmap	X	X
image/x-xwindowdump	X	X
text/csv		X
text/html		X
text/plain		X
text/xml		X

application/framemaker - fm

This format cannot be transformed into, or generated from, any other format.

application/illustrator - ai

 This format cannot be generated from any other format.

Format	Transformable to:
application/eps	X
application/x-shockwave-flash	X
image/bmp	X
image/cgm	X
image/gif	X
image/ief	X
image/jp2	X
image/jpeg	X
image/png	X
image/tiff	X
image/vnd.adobe.photoshop	X
image/vnd.adobe.premiere	X
image/x-cmu-raster	X
image/x-dwt	X
image/x-portable-anymap	X
image/x-portable-bitmap	X
image/x-portable-graymap	X
image/x-portable-pixmap	X
image/x-raw-adobe	X
image/x-raw-canon	X
image/x-raw-fuji	X
image/x-raw-hasselblad	X
image/x-raw-kodak	X
image/x-raw-leica	X
image/x-raw-minolta	X
image/x-raw-nikon	X
image/x-raw-olympus	X
image/x-raw-panasonic	X
image/x-raw-pentax	X
image/x-raw-red	X
image/x-raw-sigma	X
image/x-raw-sony	X
image/x-xbitmap	X
image/x-xpixmap	X
image/x-xwindowdump	X

application/java, application/json and application/mac-binhex40 - class, json, hqx

These formats cannot be transformed into, or generated from, any other format.

application/vnd.msword - doc

Format	Transformable to:	Transformable from:
application/eps	X	
application/pdf	X	
application/rtf	X	X
application/vnd.ms-outlook		X
application/vnd.ms-word.document.macroenabled.12		X
application/vnd.ms-word.template.macroenabled.12		X

Format	Transformable to:	Transformable from:
application/vnd.oasis.opendocument.text	X	X
application/vnd.oasis.opendocument.text-template	X	X
application/vnd.openxmlformats-officedocument.wordprocessingml.document		X
application/vnd.openxmlformats-officedocument.wordprocessingml.template		X
application/vnd.sun.xml.writer	X	X
application/vnd.sun.xml.writer.template	X	X
application/wordperfect		X
application/x-shockwave-flash	X	
application/xhtml+xml	X	
image/bmp	X	
image/cgm	X	
image/gif	X	
image/ief	X	
image/jp2	X	
image/jpeg	X	
image/png	X	
image/tiff	X	
image/vnd.adobe.photoshop	X	
image/vnd.adobe.premiere	X	
image/x-cmu-raster	X	
image/x-dwt	X	
image/x-portable-anymap	X	
image/x-portable-bitmap	X	
image/x-portable-graymap	X	
image/x-portable-pixmap	X	
image/x-raw-adobe	X	
image/x-raw-canon	X	
image/x-raw-fuji	X	
image/x-raw-hasselblad	X	
image/x-raw-kodak	X	
image/x-raw-leica	X	
image/x-raw-minolta	X	
image/x-raw-nikon	X	
image/x-raw-olympus	X	
image/x-raw-panasonic	X	
image/x-raw-pentax	X	
image/x-raw-red	X	
image/x-raw-sigma	X	
image/x-raw-sony	X	
image/x-xbitmap	X	
image/x-ypixmap	X	
image/x-xwindowdump	X	
text/html	X	X
text/plain	X	X
text/xml	X	

application/octet-stream and application/oda - bin, oda

These formats cannot be transformed into, or generated from, any other format.

application/ogg - ogx

 This format cannot be generated from any other format.

Format	Transformable to:
application/xhtml+xml	X
text/html	X
text/plain	X
text/xml	X

application/pagemaker - pmd

This format cannot be transformed into, or generated from, any other format.

application/pdf - pdf

Format	Transformable to:	Transformable from:
application/eps	X	
application/vnd.msword		X
application/rtf		X
application/vnd.apple.keynote		X
application/vnd.apple.numbers		X
application/vnd.apple.pages		X
application/vnd.ms-excel		X
application/vnd.ms-excel.sheet.binary.macroenabled.12		X
application/vnd.ms-excel.sheet.macroenabled.12		X
application/vnd.ms-excel.template.macroenabled.12		X
application/vnd.ms-outlook		X
application/vnd.ms-powerpoint		X
application/vnd.ms-powerpoint.addin.macroenabled.12		X
application/vnd.ms-powerpoint.presentation.macroenabled.12		X
application/vnd.ms-powerpoint.slide.macroenabled.12		X
application/vnd.ms-powerpoint.template.macroenabled.12		X
application/vnd.ms-word.document.macroenabled.12		X
application/vnd.ms-word.template.macroenabled.12		X
application/vnd.oasis.opendocument.presentation		X
application/vnd.oasis.opendocument.presentation-template		X
application/vnd.oasis.opendocument.spreadsheet		X
application/vnd.oasis.opendocument.spreadsheet-template		X
application/vnd.oasis.opendocument.text		X
application/vnd.oasis.opendocument.text-template		X
application/vnd.openxmlformats-officedocument.presentationml.presentation		X
application/vnd.openxmlformats-officedocument.presentationml.slide		X
application/vnd.openxmlformats-officedocument.presentationml.template		X

Format	Transformable to:	Transformable from:
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet		X
application/vnd.openxmlformats-officedocument.spreadsheetml.template		X
application/vnd.openxmlformats-officedocument.wordprocessingml.document		X
application/vnd.openxmlformats-officedocument.wordprocessingml.template		X
application/vnd.sun.xml.calc		X
application/vnd.sun.xml.calc.template		X
application/vnd.sun.xml.impress		X
application/vnd.sun.xml.impress.template		X
application/vnd.sun.xml.writer		X
application/vnd.sun.xml.writer.template		X
application/vnd.visio		X
application/wordperfect		X
application/x-shockwave-flash	X	
application/xhtml+xml	X	
image/bmp	X	
image/cgm	X	
image/gif	X	
image/ief	X	
image/jp2	X	
image/jpeg	X	
image/png	X	
image/tiff	X	X
image/vnd.adobe.photoshop	X	
image/vnd.adobe.premiere	X	
image/x-cmu-raster	X	
image/x-dwt	X	
image/x-portable-anymap	X	
image/x-portable-bitmap	X	
image/x-portable-graymap	X	
image/x-portable-pixmap	X	
image/x-raw-adobe	X	
image/x-raw-canon	X	
image/x-raw-fuji	X	
image/x-raw-hasselblad	X	
image/x-raw-kodak	X	
image/x-raw-leica	X	
image/x-raw-minolta	X	
image/x-raw-nikon	X	
image/x-raw-olympus	X	
image/x-raw-panasonic	X	
image/x-raw-pentax	X	
image/x-raw-red	X	
image/x-raw-sigma	X	

Format	Transformable to:	Transformable from:
image/x-raw-sony	X	
image/x-xbitmap	X	
image/x-xpixmap	X	
image/x-xwindowdump	X	
text/csv		X
text/html	X	X
text/plain	X	X
text/xml	X	X

application/postscript and application/remote-printing - ps, prn

These formats cannot be transformed into, or generated from, any other format.

application/rss+xml - rss



This format cannot be generated from any other format.

Format	Transformable to:
application/xhtml+xml	X
text/html	X
text/plain	X
text/xml	X

application/rtf - rtf

Format	Transformable to:	Transformable from:
application/eps	X	
application/vnd.msword	X	X
application/pdf	X	
application/vnd.ms-outlook		X
application/vnd.ms-word.document.macroenabled.12		X
application/vnd.ms-word.template.macroenabled.12		X
application/vnd.oasis.opendocument.text	X	X
application/vnd.oasis.opendocument.text-template	X	X
application/vnd.openxmlformats-officedocument.wordprocessingml.document		X
application/vnd.openxmlformats-officedocument.wordprocessingml.template		X
application/vnd.sun.xml.writer	X	X
application/vnd.sun.xml.writer.template	X	X
application/wordperfect		X
application/x-shockwave-flash	X	
application/xhtml+xml	X	
image/bmp	X	
image/cgm	X	
image/gif	X	
image/ief	X	
image/jp2	X	
image/jpeg	X	
image/png	X	

Format	Transformable to:	Transformable from:
image/tiff	X	
image/vnd.adobe.photoshop	X	
image/vnd.adobe.premiere	X	
image/x-cmu-raster	X	
image/x-dwt	X	
image/x-portable-anymap	X	
image/x-portable-bitmap	X	
image/x-portable-graymap	X	
image/x-portable-pixmap	X	
image/x-raw-adobe	X	
image/x-raw-canon	X	
image/x-raw-fuji	X	
image/x-raw-hasselblad	X	
image/x-raw-kodak	X	
image/x-raw-leica	X	
image/x-raw-minolta	X	
image/x-raw-nikon	X	
image/x-raw-olympus	X	
image/x-raw-panasonic	X	
image/x-raw-pentax	X	
image/x-raw-red	X	
image/x-raw-sigma	X	
image/x-raw-sony	X	
image/x-xbitmap	X	
image/x-ypixmap	X	
image/x-xwindowdump	X	
text/html	X	X
text/plain	X	X
text/xml	X	

application/sgml, application/vnd.adobe.aftereffects.project, application/vnd.adobe.aftereffects.template, application/vnd.adobe.air-application-installer-package+zip, application/vnd.adobe.xdp+xml and application/vnd.android.package-archive - sgml, aep, aet, air, xdp, apk

These formats cannot be transformed into, or generated from, any other format.

application/vnd.apple.keynote, application/vnd.apple.numbers and application/vnd.apple.pages - key, numbers, pages



These formats cannot be generated from any other format.

Format	Transformable to:
application/eps	X
application/pdf	X
application/x-shockwave-flash	X
application/xhtml+xml	X
image/bmp	X
image/cgm	X

Format	Transformable to:
image/gif	X
image/ief	X
image/jp2	X
image/jpeg	X
image/png	X
image/tiff	X
image/vnd.adobe.photoshop	X
image/vnd.adobe.premiere	X
image/x-cmu-raster	X
image/x-dwt	X
image/x-portable-anymap	X
image/x-portable-bitmap	X
image/x-portable-graymap	X
image/x-portable-pixmap	X
image/x-raw-adobe	X
image/x-raw-canon	X
image/x-raw-fuji	X
image/x-raw-hasselblad	X
image/x-raw-kodak	X
image/x-raw-leica	X
image/x-raw-minolta	X
image/x-raw-nikon	X
image/x-raw-olympus	X
image/x-raw-panasonic	X
image/x-raw-pentax	X
image/x-raw-red	X
image/x-raw-sigma	X
image/x-raw-sony	X
image/x-xbitmap	X
image/x-xpixmap	X
image/x-xwindowdump	X
text/html	X
text/plain	X
text/xml	X

application/vnd.ms-excel - xls

Format	Transformable to:	Transformable from:
application/eps	X	
application/pdf	X	
application/vnd.ms-excel.sheet.binary.macroenabled.12		X
application/vnd.ms-excel.sheet.macroenabled.12		X
application/vnd.ms-excel.template.macroenabled.12		X
application/vnd.oasis.opendocument.spreadsheet	X	X
application/vnd.oasis.opendocument.spreadsheet-template	X	X
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet		X

Format	Transformable to:	Transformable from:
application/vnd.openxmlformats-officedocument.spreadsheetml.template		X
application/vnd.sun.xml.calc	X	X
application/vnd.sun.xml.calc.template	X	X
application/x-shockwave-flash	X	
application/xhtml+xml	X	
image/bmp	X	
image/cgm	X	
image/gif	X	
image/ief	X	
image/jp2	X	
image/jpeg	X	
image/png	X	
image/tiff	X	
image/vnd.adobe.photoshop	X	
image/vnd.adobe.premiere	X	
image/x-cmu-raster	X	
image/x-dwt	X	
image/x-portable-anymap	X	
image/x-portable-bitmap	X	
image/x-portable-graymap	X	
image/x-portable-pixmap	X	
image/x-raw-adobe	X	
image/x-raw-canon	X	
image/x-raw-fuji	X	
image/x-raw-hasselblad	X	
image/x-raw-kodak	X	
image/x-raw-leica	X	
image/x-raw-minolta	X	
image/x-raw-nikon	X	
image/x-raw-olympus	X	
image/x-raw-panasonic	X	
image/x-raw-pentax	X	
image/x-raw-red	X	
image/x-raw-sigma	X	
image/x-raw-sony	X	
image/x-xbitmap	X	
image/x-ypixmap	X	
image/x-xwindowdump	X	
text/csv	X	
text/html	X	
text/plain	X	
text/xml	X	

application/vnd.ms-excel.addin.macroenabled.12 - xlam



This format cannot be generated from any other format.

Format	Transformable to:
application/eps	X
application/xhtml+xml	X
image/bmp	X
image/cgm	X
image/gif	X
image/ief	X
image/jp2	X
image/jpeg	X
image/png	X
image/tiff	X
image/vnd.adobe.photoshop	X
image/vnd.adobe.premiere	X
image/x-cmu-raster	X
image/x-dwt	X
image/x-portable-anymap	X
image/x-portable-bitmap	X
image/x-portable-graymap	X
image/x-portable-pixmap	X
image/x-raw-adobe	X
image/x-raw-canon	X
image/x-raw-fuji	X
image/x-raw-hasselblad	X
image/x-raw-kodak	X
image/x-raw-leica	X
image/x-raw-minolta	X
image/x-raw-nikon	X
image/x-raw-olympus	X
image/x-raw-panasonic	X
image/x-raw-pentax	X
image/x-raw-red	X
image/x-raw-sigma	X
image/x-raw-sony	X
image/x-xbitmap	X
image/x-xpixmap	X
image/x-xwindowdump	X
text/html	X
text/plain	X
text/xml	X

application/vnd.ms-excel.sheet.binary.macroenabled.12, application/vnd.ms-excel.sheet.macroenabled.12, application/vnd.ms-excel.template.macroenabled.12 and application/vnd.openxmlformats-officedocument.spreadsheetml.template - xlsb, xlsx, xltm, xltx



These formats cannot be generated from any other format.

Format	Transformable to:
application/eps	X

Format	Transformable to:
application/pdf	X
application/vnd.ms-excel	X
application/vnd.oasis.opendocument.spreadsheet	X
application/vnd.oasis.opendocument.spreadsheet-template	X
application/vnd.sun.xml.calc	X
application/vnd.sun.xml.calc.template	X
application/x-shockwave-flash	X
application/xhtml+xml	X
image/bmp	X
image/cgm	X
image/gif	X
image/ief	X
image/jp2	X
image/jpeg	X
image/png	X
image/tiff	X
image/vnd.adobe.photoshop	X
image/vnd.adobe.premiere	X
image/x-cmu-raster	X
image/x-dwt	X
image/x-portable-anymap	X
image/x-portable-bitmap	X
image/x-portable-graymap	X
image/x-portable-pixmap	X
image/x-raw-adobe	X
image/x-raw-canon	X
image/x-raw-fuji	X
image/x-raw-hasselblad	X
image/x-raw-kodak	X
image/x-raw-leica	X
image/x-raw-minolta	X
image/x-raw-nikon	X
image/x-raw-olympus	X
image/x-raw-panasonic	X
image/x-raw-pentax	X
image/x-raw-red	X
image/x-raw-sigma	X
image/x-raw-sony	X
image/x-xbitmap	X
image/x-pixmap	X
image/x-windowdump	X
text/html	X
text/plain	X
text/xml	X

application/vnd.openxmlformats-officedocument.spreadsheetml.sheet - xlsx



This format cannot be generated from any other format.

Format	Transformable to:
application/eps	X
application/pdf	X
application/vnd.ms-excel	X
application/vnd.oasis.opendocument.spreadsheet	X
application/vnd.oasis.opendocument.spreadsheet-template	X
application/vnd.sun.xml.calc	X
application/vnd.sun.xml.calc.template	X
application/x-shockwave-flash	X
application/xhtml+xml	X
image/bmp	X
image/cgm	X
image/gif	X
image/ief	X
image/jp2	X
image/jpeg	X
image/png	X
image/tiff	X
image/vnd.adobe.photoshop	X
image/vnd.adobe.premiere	X
image/x-cmu-raster	X
image/x-dwt	X
image/x-portable-anymap	X
image/x-portable-bitmap	X
image/x-portable-graymap	X
image/x-portable-pixmap	X
image/x-raw-adobe	X
image/x-raw-canon	X
image/x-raw-fuji	X
image/x-raw-hasselblad	X
image/x-raw-kodak	X
image/x-raw-leica	X
image/x-raw-minolta	X
image/x-raw-nikon	X
image/x-raw-olympus	X
image/x-raw-panasonic	X
image/x-raw-pentax	X
image/x-raw-red	X
image/x-raw-sigma	X
image/x-raw-sony	X
image/x-xbitmap	X
image/x-xpixmap	X
image/x-xwindowdump	X
text/csv	X
text/html	X

Format	Transformable to:
text/plain	X
text/xml	X

application/vnd.ms-outlook - msg



This format cannot be generated from any other format.

Format	Transformable to:
application/eps	X
application/vnd.msword	X
application/pdf	X
application/rtf	X
application/vnd.oasis.opendocument.text	X
application/vnd.oasis.opendocument.text-template	X
application/vnd.sun.xml.writer	X
application/vnd.sun.xml.writer.template	X
application/x-shockwave-flash	X
application/xhtml+xml	X
image/bmp	X
image/cgm	X
image/gif	X
image/ief	X
image/jp2	X
image/jpeg	X
image/png	X
image/tiff	X
image/vnd.adobe.photoshop	X
image/vnd.adobe.premiere	X
image/x-cmu-raster	X
image/x-dwt	X
image/x-portable-anymap	X
image/x-portable-bitmap	X
image/x-portable-graymap	X
image/x-portable-pixmap	X
image/x-raw-adobe	X
image/x-raw-canon	X
image/x-raw-fuji	X
image/x-raw-hasselblad	X
image/x-raw-kodak	X
image/x-raw-leica	X
image/x-raw-minolta	X
image/x-raw-nikon	X
image/x-raw-olympus	X
image/x-raw-panasonic	X
image/x-raw-pentax	X
image/x-raw-red	X
image/x-raw-sigma	X
image/x-raw-sony	X

Format	Transformable to:
image/x-xbitmap	X
image/x-xpixmap	X
image/x-xwindowdump	X
text/html	X
text/plain	X
text/xml	X

application/vnd.ms-powerpoint - ppt

Format	Transformable to:	Transformable from:
application/eps	X	
application/pdf	X	
application/vnd.ms-powerpoint.addin.macroenabled.12		X
application/vnd.ms-powerpoint.presentation.macroenabled.12		X
application/vnd.ms-powerpoint.slide.macroenabled.12		X
application/vnd.ms-powerpoint.template.macroenabled.12		X
application/vnd.oasis.opendocument.presentation	X	X
application/vnd.oasis.opendocument.presentation-template	X	X
application/vnd.openxmlformats-officedocument.presentationml.presentation		X
application/vnd.openxmlformats-officedocument.presentationml.slide		X
application/vnd.openxmlformats-officedocument.presentationml.template		X
application/vnd.sun.xml.impress	X	X
application/vnd.sun.xml.impress.template	X	X
application/vnd.visio	X	X
application/x-shockwave-flash	X	
application/xhtml+xml	X	
image/bmp	X	
image/cgm	X	
image/gif	X	
image/ief	X	
image/jp2	X	
image/jpeg	X	
image/png	X	
image/tiff	X	
image/vnd.adobe.photoshop	X	
image/vnd.adobe.premiere	X	
image/x-cmu-raster	X	
image/x-dwt	X	
image/x-portable-anymap	X	
image/x-portable-bitmap	X	
image/x-portable-graymap	X	
image/x-portable-pixmap	X	
image/x-raw-adobe	X	
image/x-raw-canon	X	

Format	Transformable to:	Transformable from:
image/x-raw-fuji	X	
image/x-raw-hasselblad	X	
image/x-raw-kodak	X	
image/x-raw-leica	X	
image/x-raw-minolta	X	
image/x-raw-nikon	X	
image/x-raw-olympus	X	
image/x-raw-panasonic	X	
image/x-raw-pentax	X	
image/x-raw-red	X	
image/x-raw-sigma	X	
image/x-raw-sony	X	
image/x-xbitmap	X	
image/x-xpixmap	X	
image/x-xwindowdump	X	
text/html	X	
text/plain	X	
text/xml	X	

application/vnd.ms-powerpoint.addin.macroenabled.12, application/vnd.ms-powerpoint.presentation.macroenabled.12 and application/vnd.ms-powerpoint.template.macroenabled.12 - ppam, pptm, potm



These formats cannot be generated from any other format.

Format	Transformable to:
application/eps	X
application/pdf	X
application/vnd.ms-powerpoint	X
application/vnd.oasis.opendocument.presentation	X
application/vnd.oasis.opendocument.presentation-template	X
application/vnd.sun.xml.impress	X
application/vnd.sun.xml.impress.template	X
application/vnd.visio	X
application/x-shockwave-flash	X
application/xhtml+xml	X
image/bmp	X
image/cgm	X
image/gif	X
image/ief	X
image/jp2	X
image/jpeg	X
image/png	X
image/tiff	X
image/vnd.adobe.photoshop	X
image/vnd.adobe.premiere	X
image/x-cmu-raster	X

Format	Transformable to:
image/x-dwt	X
image/x-portable-anymap	X
image/x-portable-bitmap	X
image/x-portable-graymap	X
image/x-portable-pixmap	X
image/x-raw-adobe	X
image/x-raw-canon	X
image/x-raw-fuji	X
image/x-raw-hasselblad	X
image/x-raw-kodak	X
image/x-raw-leica	X
image/x-raw-minolta	X
image/x-raw-nikon	X
image/x-raw-olympus	X
image/x-raw-panasonic	X
image/x-raw-pentax	X
image/x-raw-red	X
image/x-raw-sigma	X
image/x-raw-sony	X
image/x-xbitmap	X
image/x-ypixmap	X
image/x-xwindowdump	X
text/html	X
text/plain	X
text/xml	X

application/vnd.ms-powerpoint.slide.macroenabled.12 - sldm



This format cannot be generated from any other format.

Format	Transformable to:
application/eps	X
application/pdf	X
application/vnd.ms-powerpoint	X
application/vnd.oasis.opendocument.presentation	X
application/vnd.oasis.opendocument.presentation-template	X
application/vnd.sun.xml.impress	X
application/vnd.sun.xml.impress.template	X
application/vnd.visio	X
application/x-shockwave-flash	X
application/xhtml+xml	X
image/bmp	X
image/cgm	X
image/gif	X
image/ief	X
image/jp2	X
image/jpeg	X
image/png	X

Format	Transformable to:
image/tiff	X
image/vnd.adobe.photoshop	X
image/vnd.adobe.premiere	X
image/x-cmu-raster	X
image/x-dwt	X
image/x-portable-anymap	X
image/x-portable-bitmap	X
image/x-portable-graymap	X
image/x-portable-pixmap	X
image/x-raw-adobe	X
image/x-raw-canon	X
image/x-raw-fuji	X
image/x-raw-hasselblad	X
image/x-raw-kodak	X
image/x-raw-leica	X
image/x-raw-minolta	X
image/x-raw-nikon	X
image/x-raw-olympus	X
image/x-raw-panasonic	X
image/x-raw-pentax	X
image/x-raw-red	X
image/x-raw-sigma	X
image/x-raw-sony	X
image/x-xbitmap	X
image/x-xpixmap	X
image/x-xwindowdump	X
text/html	X
text/xml	X

application/vnd.ms-powerpoint.slideshow.macroenabled.12 - ppsm



This format cannot be generated from any other format.

Format	Transformable to:
application/eps	X
application/xhtml+xml	X
image/bmp	X
image/cgm	X
image/gif	X
image/ief	X
image/jp2	X
image/jpeg	X
image/png	X
image/tiff	X
image/vnd.adobe.photoshop	X
image/vnd.adobe.premiere	X
image/x-cmu-raster	X
image/x-dwt	X

Format	Transformable to:
image/x-portable-anymap	X
image/x-portable-bitmap	X
image/x-portable-graymap	X
image/x-portable-pixmap	X
image/x-raw-adobe	X
image/x-raw-canon	X
image/x-raw-fuji	X
image/x-raw-hasselblad	X
image/x-raw-kodak	X
image/x-raw-leica	X
image/x-raw-minolta	X
image/x-raw-nikon	X
image/x-raw-olympus	X
image/x-raw-panasonic	X
image/x-raw-pentax	X
image/x-raw-red	X
image/x-raw-sigma	X
image/x-raw-sony	X
image/x-xbitmap	X
image/x-xpixmap	X
image/x-xwindowdump	X
text/html	X
text/plain	X
text/xml	X

application/vnd.ms-project - mpp



This format cannot be generated from any other format.

Format	Transformable to:
application/xhtml+xml	X
text/html	X
text/plain	X
text/xml	X

application/vnd.ms-word.document.macroenabled.12, application/vnd.ms-word.template.macroenabled.12, application/vnd.openxmlformats-officedocument.wordprocessingml.document and application/vnd.openxmlformats-officedocument.wordprocessingml.template - docm, dotm, docx, dotx



These formats cannot be generated from any other format.

Format	Transformable to:
application/eps	X
application/vnd.msword	X
application/pdf	X
application/rtf	X
application/vnd.oasis.opendocument.text	X
application/vnd.oasis.opendocument.text-template	X
application/vnd.sun.xml.writer	X

Format	Transformable to:
application/vnd.sun.xml.writer.template	X
application/x-shockwave-flash	X
application/xhtml+xml	X
image/bmp	X
image/cgm	X
image/gif	X
image/ief	X
image/jp2	X
image/jpeg	X
image/png	X
image/tiff	X
image/vnd.adobe.photoshop	X
image/vnd.adobe.premiere	X
image/x-cmu-raster	X
image/x-dwt	X
image/x-portable-anymap	X
image/x-portable-bitmap	X
image/x-portable-graymap	X
image/x-portable-pixmap	X
image/x-raw-adobe	X
image/x-raw-canon	X
image/x-raw-fuji	X
image/x-raw-hasselblad	X
image/x-raw-kodak	X
image/x-raw-leica	X
image/x-raw-minolta	X
image/x-raw-nikon	X
image/x-raw-olympus	X
image/x-raw-panasonic	X
image/x-raw-pentax	X
image/x-raw-red	X
image/x-raw-sigma	X
image/x-raw-sony	X
image/x-xbitmap	X
image/x-ypixmap	X
image/x-xwindowdump	X
text/html	X
text/plain	X
text/xml	X

application/vnd.oasis.opendocument.chart, application/vnd.oasis.opendocument.image, application/vnd.oasis.opendocument.text-master and application/vnd.oasis.opendocument.text-web - odc, odi, odm, oth



These formats cannot be generated from any other format.

Format	Transformable to:
application/xhtml+xml	X

Format	Transformable to:
text/html	X
text/plain	X
text/xml	X

application/vnd.oasis.opendocument.database, application/vnd.oasis.opendocument.formula, application/vnd.oasis.opendocument.graphics and application/vnd.oasis.opendocument.graphics-template - odb, odf, odg, otg

These formats cannot be transformed into, or generated from, any other format.

application/vnd.oasis.opendocument.presentation - odp

Format	Transformable to:	Transformable from:
application/eps	X	
application/pdf	X	
application/vnd.ms-powerpoint	X	X
application/vnd.ms-powerpoint.addin.macroenabled.12		X
application/vnd.ms-powerpoint.presentation.macroenabled.12		X
application/vnd.ms-powerpoint.slide.macroenabled.12		X
application/vnd.ms-powerpoint.template.macroenabled.12		X
application/vnd.oasis.opendocument.presentation-template	X	X
application/vnd.openxmlformats-officedocument.presentationml.presentation		X
application/vnd.openxmlformats-officedocument.presentationml.slide		X
application/vnd.openxmlformats-officedocument.presentationml.template		X
application/vnd.sun.xml.impress	X	X
application/vnd.sun.xml.impress.template	X	X
application/vnd.visio	X	X
application/x-shockwave-flash	X	
application/xhtml+xml	X	
image/bmp	X	
image/cgm	X	
image/gif	X	
image/ief	X	
image/jp2	X	
image/jpeg	X	
image/png	X	
image/tiff	X	
image/vnd.adobe.photoshop	X	
image/vnd.adobe.premiere	X	
image/x-cmu-raster	X	
image/x-dwt	X	
image/x-portable-anymap	X	
image/x-portable-bitmap	X	
image/x-portable-graymap	X	
image/x-portable-pixmap	X	

Format	Transformable to:	Transformable from:
image/x-raw-adobe	X	
image/x-raw-canon	X	
image/x-raw-fuji	X	
image/x-raw-hasselblad	X	
image/x-raw-kodak	X	
image/x-raw-leica	X	
image/x-raw-minolta	X	
image/x-raw-nikon	X	
image/x-raw-olympus	X	
image/x-raw-panasonic	X	
image/x-raw-pentax	X	
image/x-raw-red	X	
image/x-raw-sigma	X	
image/x-raw-sony	X	
image/x-xbitmap	X	
image/x-pixmap	X	
image/x-xwindowdump	X	
text/html	X	
text/plain	X	
text/xml	X	

application/vnd.oasis.opendocument.presentation-template - otp

Format	Transformable to:	Transformable from:
application/eps	X	
application/pdf	X	
application/vnd.ms-powerpoint	X	X
application/vnd.ms-powerpoint.addin.macroenabled.12		X
application/vnd.ms-powerpoint.presentation.macroenabled.12		X
application/vnd.ms-powerpoint.slide.macroenabled.12		X
application/vnd.ms-powerpoint.template.macroenabled.12		X
application/vnd.oasis.opendocument.presentation	X	X
application/vnd.openxmlformats-officedocument.presentationml.presentation		X
application/vnd.openxmlformats-officedocument.presentationml.slide		X
application/vnd.openxmlformats-officedocument.presentationml.template		X
application/vnd.sun.xml.impress	X	X
application/vnd.sun.xml.impress.template	X	X
application/vnd.visio	X	X
application/x-shockwave-flash	X	
application/xhtml+xml	X	
image/bmp	X	
image/cgm	X	
image/gif	X	

Format	Transformable to:	Transformable from:
image/ief	X	
image/jp2	X	
image/jpeg	X	
image/png	X	
image/tiff	X	
image/vnd.adobe.photoshop	X	
image/vnd.adobe.premiere	X	
image/x-cmu-raster	X	
image/x-dwt	X	
image/x-portable-anymap	X	
image/x-portable-bitmap	X	
image/x-portable-graymap	X	
image/x-portable-pixmap	X	
image/x-raw-adobe	X	
image/x-raw-canon	X	
image/x-raw-fuji	X	
image/x-raw-hasselblad	X	
image/x-raw-kodak	X	
image/x-raw-leica	X	
image/x-raw-minolta	X	
image/x-raw-nikon	X	
image/x-raw-olympus	X	
image/x-raw-panasonic	X	
image/x-raw-pentax	X	
image/x-raw-red	X	
image/x-raw-sigma	X	
image/x-raw-sony	X	
image/x-xbitmap	X	
image/x-ypixmap	X	
image/x-xwindowdump	X	
text/html	X	
text/plain	X	
text/xml	X	

application/vnd.oasis.opendocument.spreadsheet - ods

Format	Transformable to:	Transformable from:
application/eps	X	
application/pdf	X	
application/vnd.ms-excel	X	X
application/vnd.ms-excel.sheet.binary.macroenabled.12		X
application/vnd.ms-excel.sheet.macroenabled.12		X
application/vnd.ms-excel.template.macroenabled.12		X
application/vnd.oasis.opendocument.spreadsheet-template	X	X
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet		X

Format	Transformable to:	Transformable from:
application/vnd.openxmlformats-officedocument.spreadsheetml.template		X
application/vnd.sun.xml.calc	X	X
application/vnd.sun.xml.calc.template	X	X
application/x-shockwave-flash	X	
application/xhtml+xml	X	
image/bmp	X	
image/cgm	X	
image/gif	X	
image/ief	X	
image/jp2	X	
image/jpeg	X	
image/png	X	
image/tiff	X	
image/vnd.adobe.photoshop	X	
image/vnd.adobe.premiere	X	
image/x-cmu-raster	X	
image/x-dwt	X	
image/x-portable-anymap	X	
image/x-portable-bitmap	X	
image/x-portable-graymap	X	
image/x-portable-pixmap	X	
image/x-raw-adobe	X	
image/x-raw-canon	X	
image/x-raw-fuji	X	
image/x-raw-hasselblad	X	
image/x-raw-kodak	X	
image/x-raw-leica	X	
image/x-raw-minolta	X	
image/x-raw-nikon	X	
image/x-raw-olympus	X	
image/x-raw-panasonic	X	
image/x-raw-pentax	X	
image/x-raw-red	X	
image/x-raw-sigma	X	
image/x-raw-sony	X	
image/x-xbitmap	X	
image/x-xpixmap	X	
image/x-xwindowdump	X	
text/html	X	
text/plain	X	
text/xml	X	

application/vnd.oasis.opendocument.spreadsheet-template - ots

Format	Transformable to:	Transformable from:
application/eps	X	
application/pdf	X	

Format	Transformable to:	Transformable from:
application/vnd.ms-excel	X	X
application/vnd.ms-excel.sheet.binary.macroenabled.12		X
application/vnd.ms-excel.sheet.macroenabled.12		X
application/vnd.ms-excel.template.macroenabled.12		X
application/vnd.oasis.opendocument.spreadsheet	X	X
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet		X
application/vnd.openxmlformats-officedocument.spreadsheetml.template		X
application/vnd.sun.xml.calc	X	X
application/vnd.sun.xml.calc.template	X	X
application/x-shockwave-flash	X	
application/xhtml+xml	X	
image/bmp	X	
image/cgm	X	
image/gif	X	
image/ief	X	
image/jp2	X	
image/jpeg	X	
image/png	X	
image/tiff	X	
image/vnd.adobe.photoshop	X	
image/vnd.adobe.premiere	X	
image/x-cmu-raster	X	
image/x-dwt	X	
image/x-portable-anymap	X	
image/x-portable-bitmap	X	
image/x-portable-graymap	X	
image/x-portable-pixmap	X	
image/x-raw-adobe	X	
image/x-raw-canon	X	
image/x-raw-fuji	X	
image/x-raw-hasselblad	X	
image/x-raw-kodak	X	
image/x-raw-leica	X	
image/x-raw-minolta	X	
image/x-raw-nikon	X	
image/x-raw-olympus	X	
image/x-raw-panasonic	X	
image/x-raw-pentax	X	
image/x-raw-red	X	
image/x-raw-sigma	X	
image/x-raw-sony	X	
image/x-xbitmap	X	
image/x-xpixmap	X	
image/x-xwindowdump	X	
text/html	X	

Format	Transformable to:	Transformable from:
text/plain	X	
text/xml	X	

application/vnd.oasis.opendocument.text - odt

Format	Transformable to:	Transformable from:
application/eps	X	
application/vnd.msword	X	X
application/pdf	X	
application/rtf	X	X
application/vnd.ms-outlook		X
application/vnd.ms-word.document.macroenabled.12		X
application/vnd.ms-word.template.macroenabled.12		X
application/vnd.oasis.opendocument.text-template	X	X
application/vnd.openxmlformats-officedocument.wordprocessingml.document		X
application/vnd.openxmlformats-officedocument.wordprocessingml.template		X
application/vnd.sun.xml.writer	X	X
application/vnd.sun.xml.writer.template	X	X
application/wordperfect		X
application/x-shockwave-flash	X	
application/xhtml+xml	X	
image/bmp	X	
image/cgm	X	
image/gif	X	
image/ief	X	
image/jp2	X	
image/jpeg	X	
image/png	X	
image/tiff	X	
image/vnd.adobe.photoshop	X	
image/vnd.adobe.premiere	X	
image/x-cmu-raster	X	
image/x-dwt	X	
image/x-portable-anymap	X	
image/x-portable-bitmap	X	
image/x-portable-graymap	X	
image/x-portable-pixmap	X	
image/x-raw-adobe	X	
image/x-raw-canon	X	
image/x-raw-fuji	X	
image/x-raw-hasselblad	X	
image/x-raw-kodak	X	
image/x-raw-leica	X	
image/x-raw-minolta	X	
image/x-raw-nikon	X	
image/x-raw-olympus	X	

Format	Transformable to:	Transformable from:
image/x-raw-panasonic	X	
image/x-raw-pentax	X	
image/x-raw-red	X	
image/x-raw-sigma	X	
image/x-raw-sony	X	
image/x-xbitmap	X	
image/x-ypixmap	X	
image/x-xwindowdump	X	
text/html	X	X
text/plain	X	X
text/xml	X	

application/vnd.oasis.opendocument.text-template - ott

Format	Transformable to:	Transformable from:
application/eps	X	
application/vnd.msword	X	X
application/pdf	X	
application/rtf	X	X
application/vnd.ms-outlook		X
application/vnd.ms-word.document.macroenabled.12		X
application/vnd.ms-word.template.macroenabled.12		X
application/vnd.oasis.opendocument.text	X	X
application/vnd.openxmlformats-officedocument.wordprocessingml.document		X
application/vnd.openxmlformats-officedocument.wordprocessingml.template		X
application/vnd.sun.xml.writer	X	X
application/vnd.sun.xml.writer.template	X	X
application/wordperfect		X
application/x-shockwave-flash	X	
application/xhtml+xml	X	
image/bmp	X	
image/cgm	X	
image/gif	X	
image/ief	X	
image/jp2	X	
image/jpeg	X	
image/png	X	
image/tiff	X	
image/vnd.adobe.photoshop	X	
image/vnd.adobe.premiere	X	
image/x-cmu-raster	X	
image/x-dwt	X	
image/x-portable-anymap	X	
image/x-portable-bitmap	X	
image/x-portable-graymap	X	
image/x-portable-pixmap	X	

Format	Transformable to:	Transformable from:
image/x-raw-adobe	X	
image/x-raw-canon	X	
image/x-raw-fuji	X	
image/x-raw-hasselblad	X	
image/x-raw-kodak	X	
image/x-raw-leica	X	
image/x-raw-minolta	X	
image/x-raw-nikon	X	
image/x-raw-olympus	X	
image/x-raw-panasonic	X	
image/x-raw-pentax	X	
image/x-raw-red	X	
image/x-raw-sigma	X	
image/x-raw-sony	X	
image/x-xbitmap	X	
image/x-xpixmap	X	
image/x-xwindowdump	X	
text/html	X	X
text/plain	X	X
text/xml	X	

application/vnd.openxmlformats-officedocument.presentationml.presentation - pptx



This format cannot be generated from any other format.

Format	Transformable to:
application/eps	X
application/pdf	X
application/vnd.ms-powerpoint	X
application/vnd.oasis.opendocument.presentation	X
application/vnd.oasis.opendocument.presentation-template	X
application/vnd.sun.xml.impress	X
application/vnd.sun.xml.impress.template	X
application/vnd.visio	X
application/x-shockwave-flash	X
application/xhtml+xml	X
image/bmp	X
image/cgm	X
image/gif	X
image/ief	X
image/jp2	X
image/jpeg	X
image/png	X
image/tiff	X
image/vnd.adobe.photoshop	X
image/vnd.adobe.premiere	X
image/x-cmu-raster	X
image/x-dwt	X

Format	Transformable to:
image/x-portable-anymap	X
image/x-portable-bitmap	X
image/x-portable-graymap	X
image/x-portable-pixmap	X
image/x-raw-adobe	X
image/x-raw-canon	X
image/x-raw-fuji	X
image/x-raw-hasselblad	X
image/x-raw-kodak	X
image/x-raw-leica	X
image/x-raw-minolta	X
image/x-raw-nikon	X
image/x-raw-olympus	X
image/x-raw-panasonic	X
image/x-raw-pentax	X
image/x-raw-red	X
image/x-raw-sigma	X
image/x-raw-sony	X
image/x-xbitmap	X
image/x-xpixmap	X
image/x-xwindowdump	X
text/html	X
text/plain	X
text/xml	X

application/vnd.openxmlformats-officedocument.presentationml.slide - sldx



This format cannot be generated from any other format.

Format	Transformable to:
application/eps	X
application/pdf	X
application/vnd.ms-powerpoint	X
application/vnd.oasis.opendocument.presentation	X
application/vnd.oasis.opendocument.presentation-template	X
application/vnd.sun.xml.impress	X
application/vnd.sun.xml.impress.template	X
application/vnd.visio	X
application/x-shockwave-flash	X
application/xhtml+xml	X
image/bmp	X
image/cgm	X
image/gif	X
image/ief	X
image/jp2	X
image/jpeg	X
image/png	X
image/tiff	X

Format	Transformable to:
image/vnd.adobe.photoshop	X
image/vnd.adobe.premiere	X
image/x-cmu-raster	X
image/x-dwt	X
image/x-portable-anymap	X
image/x-portable-bitmap	X
image/x-portable-graymap	X
image/x-portable-pixmap	X
image/x-raw-adobe	X
image/x-raw-canon	X
image/x-raw-fuji	X
image/x-raw-hasselblad	X
image/x-raw-kodak	X
image/x-raw-leica	X
image/x-raw-minolta	X
image/x-raw-nikon	X
image/x-raw-olympus	X
image/x-raw-panasonic	X
image/x-raw-pentax	X
image/x-raw-red	X
image/x-raw-sigma	X
image/x-raw-sony	X
image/x-xbitmap	X
image/x-xpixmap	X
image/x-xwindowdump	X
text/html	X
text/xml	X

application/vnd.openxmlformats-officedocument.presentationml.slideshow - ppsx



This format cannot be generated from any other format.

Format	Transformable to:
application/eps	X
application/xhtml+xml	X
image/bmp	X
image/cgm	X
image/gif	X
image/ief	X
image/jp2	X
image/jpeg	X
image/png	X
image/tiff	X
image/vnd.adobe.photoshop	X
image/vnd.adobe.premiere	X
image/x-cmu-raster	X
image/x-dwt	X
image/x-portable-anymap	X

Format	Transformable to:
image/x-portable-bitmap	X
image/x-portable-graymap	X
image/x-portable-pixmap	X
image/x-raw-adobe	X
image/x-raw-canon	X
image/x-raw-fuji	X
image/x-raw-hasselblad	X
image/x-raw-kodak	X
image/x-raw-leica	X
image/x-raw-minolta	X
image/x-raw-nikon	X
image/x-raw-olympus	X
image/x-raw-panasonic	X
image/x-raw-pentax	X
image/x-raw-red	X
image/x-raw-sigma	X
image/x-raw-sony	X
image/x-xbitmap	X
image/x-ypixmap	X
image/x-xwindowdump	X
text/html	X
text/plain	X
text/xml	X

application/vnd.openxmlformats-officedocument.presentationml.template - potx



This format cannot be generated from any other format.

Format	Transformable to:
application/eps	X
application/pdf	X
application/vnd.ms-powerpoint	X
application/vnd.oasis.opendocument.presentation	X
application/vnd.oasis.opendocument.presentation-template	X
application/vnd.sun.xml.impress	X
application/vnd.sun.xml.impress.template	X
application/vnd.visio	X
application/x-shockwave-flash	X
application/xhtml+xml	X
image/bmp	X
image/cgm	X
image/gif	X
image/ief	X
image/jp2	X
image/jpeg	X
image/png	X
image/tiff	X
image/vnd.adobe.photoshop	X

Format	Transformable to:
image/vnd.adobe.premiere	X
image/x-cmu-raster	X
image/x-dwt	X
image/x-portable-anymap	X
image/x-portable-bitmap	X
image/x-portable-graymap	X
image/x-portable-pixmap	X
image/x-raw-adobe	X
image/x-raw-canon	X
image/x-raw-fuji	X
image/x-raw-hasselblad	X
image/x-raw-kodak	X
image/x-raw-leica	X
image/x-raw-minolta	X
image/x-raw-nikon	X
image/x-raw-olympus	X
image/x-raw-panasonic	X
image/x-raw-pentax	X
image/x-raw-red	X
image/x-raw-sigma	X
image/x-raw-sony	X
image/x-xbitmap	X
image/x-ypixmap	X
image/x-xwindowdump	X
text/html	X
text/plain	X
text/xml	X

application/vnd.stardivision.calc, application/vnd.stardivision.chart, application/vnd.stardivision.draw, application/vnd.stardivision.impress, application/vnd.stardivision.impress-packed, application/vnd.stardivision.math, application/vnd.stardivision.writer, application/vnd.stardivision.writer-global - sdc, sds, sda, sdd, sdp, smf, sdw, sgl

These formats cannot be transformed into, or generated from, any other format.

application/vnd.sun.xml.calc - sxc

Format	Transformable to:	Transformable from:
application/eps	X	
application/pdf	X	
application/vnd.ms-excel	X	X
application/vnd.ms-excel.sheet.binary.macroenabled.12		X
application/vnd.ms-excel.sheet.macroenabled.12		X
application/vnd.ms-excel.template.macroenabled.12		X
application/vnd.oasis.opendocument.spreadsheet	X	X
application/vnd.oasis.opendocument.spreadsheet-template	X	X
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet		X

Format	Transformable to:	Transformable from:
application/vnd.openxmlformats-officedocument.spreadsheetml.template		X
application/vnd.sun.xml.calc.template	X	X
application/x-shockwave-flash	X	
application/xhtml+xml	X	
image/bmp	X	
image/cgm	X	
image/gif	X	
image/ief	X	
image/jp2	X	
image/jpeg	X	
image/png	X	
image/tiff	X	
image/vnd.adobe.photoshop	X	
image/vnd.adobe.premiere	X	
image/x-cmu-raster	X	
image/x-dwt	X	
image/x-portable-anymap	X	
image/x-portable-bitmap	X	
image/x-portable-graymap	X	
image/x-portable-pixmap	X	
image/x-raw-adobe	X	
image/x-raw-canon	X	
image/x-raw-fuji	X	
image/x-raw-hasselblad	X	
image/x-raw-kodak	X	
image/x-raw-leica	X	
image/x-raw-minolta	X	
image/x-raw-nikon	X	
image/x-raw-olympus	X	
image/x-raw-panasonic	X	
image/x-raw-pentax	X	
image/x-raw-red	X	
image/x-raw-sigma	X	
image/x-raw-sony	X	
image/x-xbitmap	X	
image/x-xpixmap	X	
image/x-xwindowdump	X	
text/html	X	
text/plain	X	
text/xml	X	

application/vnd.sun.xml.calc.template - stc

Format	Transformable to:	Transformable from:
application/eps	X	
application/pdf	X	
application/vnd.ms-excel	X	X

Format	Transformable to:	Transformable from:
application/vnd.ms-excel.sheet.binary.macroenabled.12		X
application/vnd.ms-excel.sheet.macroenabled.12		X
application/vnd.ms-excel.template.macroenabled.12		X
application/vnd.oasis.opendocument.spreadsheet	X	X
application/vnd.oasis.opendocument.spreadsheet-template	X	X
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet		X
application/vnd.openxmlformats-officedocument.spreadsheetml.template		X
application/vnd.sun.xml.calc	X	X
application/x-shockwave-flash	X	
application/xhtml+xml	X	
image/bmp	X	
image/cgm	X	
image/gif	X	
image/ief	X	
image/jp2	X	
image/jpeg	X	
image/png	X	
image/tiff	X	
image/vnd.adobe.photoshop	X	
image/vnd.adobe.premiere	X	
image/x-cmu-raster	X	
image/x-dwt	X	
image/x-portable-anymap	X	
image/x-portable-bitmap	X	
image/x-portable-graymap	X	
image/x-portable-pixmap	X	
image/x-raw-adobe	X	
image/x-raw-canon	X	
image/x-raw-fuji	X	
image/x-raw-hasselblad	X	
image/x-raw-kodak	X	
image/x-raw-leica	X	
image/x-raw-minolta	X	
image/x-raw-nikon	X	
image/x-raw-olympus	X	
image/x-raw-panasonic	X	
image/x-raw-pentax	X	
image/x-raw-red	X	
image/x-raw-sigma	X	
image/x-raw-sony	X	
image/x-xbitmap	X	
image/x-xpixmap	X	
image/x-xwindowdump	X	
text/html	X	
text/plain	X	

Format	Transformable to:	Transformable from:
text/xml	X	

application/vnd.sun.xml.draw - sxd

This format cannot be transformed into, or generated from, any other format.

application/vnd.sun.xml.impress - sxi

Format	Transformable to:	Transformable from:
application/eps	X	
application/pdf	X	
application/vnd.ms-powerpoint	X	X
application/vnd.ms-powerpoint.addin.macroenabled.12		X
application/vnd.ms-powerpoint.presentation.macroenabled.12		X
application/vnd.ms-powerpoint.slide.macroenabled.12		X
application/vnd.ms-powerpoint.template.macroenabled.12		X
application/vnd.oasis.opendocument.presentation	X	X
application/vnd.oasis.opendocument.presentation-template	X	X
application/vnd.openxmlformats-officedocument.presentationml.presentation		X
application/vnd.openxmlformats-officedocument.presentationml.slide		X
application/vnd.openxmlformats-officedocument.presentationml.template		X
application/vnd.sun.xml.impress.template	X	X
application/vnd.visio	X	X
application/x-shockwave-flash	X	
application/xhtml+xml	X	
image/bmp	X	
image/cgm	X	
image/gif	X	
image/ief	X	
image/jp2	X	
image/jpeg	X	
image/png	X	
image/tiff	X	
image/vnd.adobe.photoshop	X	
image/vnd.adobe.premiere	X	
image/x-cmu-raster	X	
image/x-dwt	X	
image/x-portable-anymap	X	
image/x-portable-bitmap	X	
image/x-portable-graymap	X	
image/x-portable-pixmap	X	
image/x-raw-adobe	X	
image/x-raw-canon	X	
image/x-raw-fuji	X	
image/x-raw-hasselblad	X	

Format	Transformable to:	Transformable from:
image/x-raw-kodak	X	
image/x-raw-leica	X	
image/x-raw-minolta	X	
image/x-raw-nikon	X	
image/x-raw-olympus	X	
image/x-raw-panasonic	X	
image/x-raw-pentax	X	
image/x-raw-red	X	
image/x-raw-sigma	X	
image/x-raw-sony	X	
image/x-xbitmap	X	
image/x-xpixmap	X	
image/x-xwindowdump	X	
text/html	X	
text/plain	X	
text/xml	X	

application/vnd.sun.xml.impress.template - sti

Format	Transformable to:	Transformable from:
application/eps	X	
application/pdf	X	
application/vnd.ms-powerpoint	X	X
application/vnd.ms-powerpoint.addin.macroenabled.12		X
application/vnd.ms-powerpoint.presentation.macroenabled.12		X
application/vnd.ms-powerpoint.slide.macroenabled.12		X
application/vnd.ms-powerpoint.template.macroenabled.12		X
application/vnd.oasis.opendocument.presentation	X	X
application/vnd.oasis.opendocument.presentation-template	X	X
application/vnd.openxmlformats-officedocument.presentationml.presentation		X
application/vnd.openxmlformats-officedocument.presentationml.slide		X
application/vnd.openxmlformats-officedocument.presentationml.template		X
application/vnd.sun.xml.impress	X	X
application/vnd.visio	X	X
application/x-shockwave-flash	X	
application/xhtml+xml	X	
image/bmp	X	
image/cgm	X	
image/gif	X	
image/ief	X	
image/jp2	X	
image/jpeg	X	
image/png	X	

Format	Transformable to:	Transformable from:
image/tiff	X	
image/vnd.adobe.photoshop	X	
image/vnd.adobe.premiere	X	
image/x-cmu-raster	X	
image/x-dwt	X	
image/x-portable-anymap	X	
image/x-portable-bitmap	X	
image/x-portable-graymap	X	
image/x-portable-pixmap	X	
image/x-raw-adobe	X	
image/x-raw-canon	X	
image/x-raw-fuji	X	
image/x-raw-hasselblad	X	
image/x-raw-kodak	X	
image/x-raw-leica	X	
image/x-raw-minolta	X	
image/x-raw-nikon	X	
image/x-raw-olympus	X	
image/x-raw-panasonic	X	
image/x-raw-pentax	X	
image/x-raw-red	X	
image/x-raw-sigma	X	
image/x-raw-sony	X	
image/x-xbitmap	X	
image/x-ypixmap	X	
image/x-xwindowdump	X	
text/html	X	
text/plain	X	
text/xml	X	

application/vnd.sun.xml.writer - sxw

Format	Transformable to:	Transformable from:
application/eps	X	
application/vnd.msword	X	X
application/pdf	X	
application/rtf	X	X
application/vnd.ms-outlook		X
application/vnd.ms-word.document.macroenabled.12		X
application/vnd.ms-word.template.macroenabled.12		X
application/vnd.oasis.opendocument.text	X	X
application/vnd.oasis.opendocument.text-template	X	X
application/vnd.openxmlformats-officedocument.wordprocessingml.document		X
application/vnd.openxmlformats-officedocument.wordprocessingml.template		X
application/vnd.sun.xml.writer.template	X	X

Format	Transformable to:	Transformable from:
application/wordperfect		X
application/x-shockwave-flash	X	
application/xhtml+xml	X	
image/bmp	X	
image/cgm	X	
image/gif	X	
image/ief	X	
image/jp2	X	
image/jpeg	X	
image/png	X	
image/tiff	X	
image/vnd.adobe.photoshop	X	
image/vnd.adobe.premiere	X	
image/x-cmu-raster	X	
image/x-dwt	X	
image/x-portable-anymap	X	
image/x-portable-bitmap	X	
image/x-portable-graymap	X	
image/x-portable-pixmap	X	
image/x-raw-adobe	X	
image/x-raw-canon	X	
image/x-raw-fuji	X	
image/x-raw-hasselblad	X	
image/x-raw-kodak	X	
image/x-raw-leica	X	
image/x-raw-minolta	X	
image/x-raw-nikon	X	
image/x-raw-olympus	X	
image/x-raw-panasonic	X	
image/x-raw-pentax	X	
image/x-raw-red	X	
image/x-raw-sigma	X	
image/x-raw-sony	X	
image/x-xbitmap	X	
image/x-xpixmap	X	
image/x-xwindowdump	X	
text/html	X	X
text/plain	X	X
text/xml	X	

application/vnd.sun.xml.writer.template - stw

Format	Transformable to:	Transformable from:
application/eps	X	
application/vnd.msword	X	X
application/pdf	X	
application/rtf	X	X
application/vnd.ms-outlook		X

Format	Transformable to:	Transformable from:
application/vnd.ms-word.document.macroenabled.12		X
application/vnd.ms-word.template.macroenabled.12		X
application/vnd.oasis.opendocument.text	X	X
application/vnd.oasis.opendocument.text-template	X	X
application/vnd.openxmlformats-officedocument.wordprocessingml.document		X
application/vnd.openxmlformats-officedocument.wordprocessingml.template		X
application/vnd.sun.xml.writer	X	X
application/wordperfect		X
application/x-shockwave-flash	X	
application/xhtml+xml	X	
image/bmp	X	
image/cgm	X	
image/gif	X	
image/ief	X	
image/jp2	X	
image/jpeg	X	
image/png	X	
image/tiff	X	
image/vnd.adobe.photoshop	X	
image/vnd.adobe.premiere	X	
image/x-cmu-raster	X	
image/x-dwt	X	
image/x-portable-anymap	X	
image/x-portable-bitmap	X	
image/x-portable-graymap	X	
image/x-portable-pixmap	X	
image/x-raw-adobe	X	
image/x-raw-canon	X	
image/x-raw-fuji	X	
image/x-raw-hasselblad	X	
image/x-raw-kodak	X	
image/x-raw-leica	X	
image/x-raw-minolta	X	
image/x-raw-nikon	X	
image/x-raw-olympus	X	
image/x-raw-panasonic	X	
image/x-raw-pentax	X	
image/x-raw-red	X	
image/x-raw-sigma	X	
image/x-raw-sony	X	
image/x-xbitmap	X	
image/x-xpixmap	X	
image/x-xwindowdump	X	
text/html	X	X
text/plain	X	X

Format	Transformable to:	Transformable from:
text/xml	X	

application/vnd.visio - vsd

Format	Transformable to:	Transformable from:
application/eps	X	
application/pdf	X	
application/vnd.ms-powerpoint	X	X
application/vnd.ms-powerpoint.addin.macroenabled.12		X
application/vnd.ms-powerpoint.presentation.macroenabled.12		X
application/vnd.ms-powerpoint.slide.macroenabled.12		X
application/vnd.ms-powerpoint.template.macroenabled.12		X
application/vnd.oasis.opendocument.presentation	X	X
application/vnd.oasis.opendocument.presentation-template	X	X
application/vnd.openxmlformats-officedocument.presentationml.presentation		X
application/vnd.openxmlformats-officedocument.presentationml.slide		X
application/vnd.openxmlformats-officedocument.presentationml.template		X
application/vnd.sun.xml.impress	X	X
application/vnd.sun.xml.impress.template	X	X
application/x-shockwave-flash	X	
application/xhtml+xml	X	
image/bmp	X	
image/cgm	X	
image/gif	X	
image/ief	X	
image/jp2	X	
image/jpeg	X	
image/png	X	
image/tiff	X	
image/vnd.adobe.photoshop	X	
image/vnd.adobe.premiere	X	
image/x-cmu-raster	X	
image/x-dwt	X	
image/x-portable-anymap	X	
image/x-portable-bitmap	X	
image/x-portable-graymap	X	
image/x-portable-pixmap	X	
image/x-raw-adobe	X	
image/x-raw-canon	X	
image/x-raw-fuji	X	
image/x-raw-hasselblad	X	
image/x-raw-kodak	X	
image/x-raw-leica	X	
image/x-raw-minolta	X	

Format	Transformable to:	Transformable from:
image/x-raw-nikon	X	
image/x-raw-olympus	X	
image/x-raw-panasonic	X	
image/x-raw-pentax	X	
image/x-raw-red	X	
image/x-raw-sigma	X	
image/x-raw-sony	X	
image/x-xbitmap	X	
image/x-pximap	X	
image/x-xwindowdump	X	
text/html	X	
text/plain	X	
text/xml	X	

application/wordperfect - wpd



This format cannot be generated from any other format.

Format	Transformable to:
application/eps	X
application/vnd.msword	X
application/pdf	X
application/rtf	X
application/vnd.oasis.opendocument.text	X
application/vnd.oasis.opendocument.text-template	X
application/vnd.sun.xml.writer	X
application/vnd.sun.xml.writer.template	X
application/x-shockwave-flash	X
application/xhtml+xml	X
image/bmp	X
image/cgm	X
image/gif	X
image/ief	X
image/jp2	X
image/jpeg	X
image/png	X
image/tiff	X
image/vnd.adobe.photoshop	X
image/vnd.adobe.premiere	X
image/x-cmu-raster	X
image/x-dwt	X
image/x-portable-anymap	X
image/x-portable-bitmap	X
image/x-portable-graymap	X
image/x-portable-pixmap	X
image/x-raw-adobe	X
image/x-raw-canon	X

Format	Transformable to:
image/x-raw-fuji	X
image/x-raw-hasselblad	X
image/x-raw-kodak	X
image/x-raw-leica	X
image/x-raw-minolta	X
image/x-raw-nikon	X
image/x-raw-olympus	X
image/x-raw-panasonic	X
image/x-raw-pentax	X
image/x-raw-red	X
image/x-raw-sigma	X
image/x-raw-sony	X
image/x-xbitmap	X
image/x-pximap	X
image/x-xwindowdump	X
text/html	X
text/plain	X
text/xml	X

application/x-bcpio, application/x-compress, application/x-csh, application/x-dvi, application/x-fla and application/x-gtar, application/x-indesign, application/x-latex, application/x-mif, application/x-sh, application/x-shar, application/x-sv4cpio, application/x-sv4crc, application/x-tcl, application/x-tex, application/x-texinfo, application/x-troff, application/x-troff-man, application/x-troff-me, application/x-troff-mes, application/x-ustar, application/x-wais-source, application/x-x509-ca-cert and application/x-zip - bcpio, z, csh, dvi, fla, gtar, indd, latex, mif, sh, shar, sv4cpio, sv4crc, tcl, tex, texinfo, tr, man, me, ms, ustar, src, cer, fxp

These formats cannot be transformed into, or generated from, any other format.

application/x-cpio, application/x-tar and application/zip - cpio, tar, zip

 These formats cannot be generated from any other format.

Format	Transformable to:
application/x-shockwave-flash	X
application/xhtml+xml	X
text/html	X
text/plain	X
text/xml	X

application/x-gzip, application/x-hdf, application/x-netcdf - gzip, hdf, cdf

 These formats cannot be generated from any other format.

Format	Transformable to:
application/xhtml+xml	X
text/html	X
text/plain	X
text/xml	X

application/x-javascript - js

This format cannot be generated from any other format.

Format	Transformable to:
text/plain	X

application/x-shockwave-flash - swf

This format cannot be transformed into any other format.

Format	Transformable from:
application/illustrator	X
application/vnd.msword	X
application/pdf	X
application/rtf	X
application/vnd.apple.keynote	X
application/vnd.apple.numbers	X
application/vnd.apple.pages	X
application/vnd.ms-excel	X
application/vnd.ms-excel.sheet.binary.macroenabled.12	X
application/vnd.ms-excel.sheet.macroenabled.12	X
application/vnd.ms-excel.template.macroenabled.12	X
application/vnd.ms-outlook	X
application/vnd.ms-powerpoint	X
application/vnd.ms-powerpoint.addin.macroenabled.12	X
application/vnd.ms-powerpoint.presentation.macroenabled.12	X
application/vnd.ms-powerpoint.slide.macroenabled.12	X
application/vnd.ms-powerpoint.template.macroenabled.12	X
application/vnd.ms-word.document.macroenabled.12	X
application/vnd.ms-word.template.macroenabled.12	X
application/vnd.oasis.opendocument.presentation	X
application/vnd.oasis.opendocument.presentation-template	X
application/vnd.oasis.opendocument.spreadsheet	X
application/vnd.oasis.opendocument.spreadsheet-template	X
application/vnd.oasis.opendocument.text	X
application/vnd.oasis.opendocument.text-template	X
application/vnd.openxmlformats-officedocument.presentationml.presentation	X
application/vnd.openxmlformats-officedocument.presentationml.slide	X
application/vnd.openxmlformats-officedocument.presentationml.template	X
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	X
application/vnd.openxmlformats-officedocument.spreadsheetml.template	X
application/vnd.openxmlformats-officedocument.wordprocessingml.document	X
application/vnd.openxmlformats-officedocument.wordprocessingml.template	X
application/vnd.sun.xml.calc	X
application/vnd.sun.xml.calc.template	X
application/vnd.sun.xml.impress	X
application/vnd.sun.xml.impress.template	X
application/vnd.sun.xml.writer	X

Format	Transformable from:
application/vnd.sun.xml.writer.template	X
application/vnd.visio	X
application/wordperfect	X
application/x-cpio	X
application/x-tar	X
application/zip	X
image/tiff	X
message/rfc822	X
text/csv	X
text/html	X
text/plain	X
text/xml	X

application/xhtml+xml - xhtml

Format	Transformable to:	Transformable from:
application/vnd.msword		X
application/ogg		X
application/pdf		X
application/rss+xml		X
application/rtf		X
application/vnd.apple.keynote		X
application/vnd.apple.numbers		X
application/vnd.apple.pages		X
application/vnd.ms-excel		X
application/vnd.ms-excel.addin.macroenabled.12		X
application/vnd.ms-excel.sheet.binary.macroenabled.12		X
application/vnd.ms-excel.sheet.macroenabled.12		X
application/vnd.ms-excel.template.macroenabled.12		X
application/vnd.ms-outlook		X
application/vnd.ms-powerpoint		X
application/vnd.ms-powerpoint.addin.macroenabled.12		X
application/vnd.ms-powerpoint.presentation.macroenabled.12		X
application/vnd.ms-powerpoint.slide.macroenabled.12		X
application/vnd.ms-powerpoint.slideshow.macroenabled.12		X
application/vnd.ms-powerpoint.template.macroenabled.12		X
application/vnd.ms-project		X
application/vnd.ms-word.document.macroenabled.12		X
application/vnd.ms-word.template.macroenabled.12		X
application/vnd.oasis.opendocument.chart		X
application/vnd.oasis.opendocument.image		X
application/vnd.oasis.opendocument.presentation		X
application/vnd.oasis.opendocument.presentation-template		X
application/vnd.oasis.opendocument.spreadsheet		X
application/vnd.oasis.opendocument.spreadsheet-template		X
application/vnd.oasis.opendocument.text		X
application/vnd.oasis.opendocument.text-master		X

Format	Transformable to:	Transformable from:
application/vnd.oasis.opendocument.text-template		X
application/vnd.oasis.opendocument.text-web		X
application/vnd.openxmlformats-officedocument.presentationml.presentation		X
application/vnd.openxmlformats-officedocument.presentationml.slide		X
application/vnd.openxmlformats-officedocument.presentationml.slideshow		X
application/vnd.openxmlformats-officedocument.presentationml.template		X
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet		X
application/vnd.openxmlformats-officedocument.spreadsheetml.template		X
application/vnd.openxmlformats-officedocument.wordprocessingml.document		X
application/vnd.openxmlformats-officedocument.wordprocessingml.template		X
application/vnd.sun.xml.calc		X
application/vnd.sun.xml.calc.template		X
application/vnd.sun.xml.impress		X
application/vnd.sun.xml.impress.template		X
application/vnd.sun.xml.writer		X
application/vnd.sun.xml.writer.template		X
application/vnd.visio		X
application/wordperfect		X
application/x-cpio		X
application/x-gzip		X
application/x-hdf		X
application/x-netcdf		X
application/x-tar		X
application/zip		X
text/html	X	X
text/plain	X	X
text/xml	X	X

audio/basic, audio/mp4, audio/mpeg, audio/ogg, audio/vnd.adobe.soundbooth, audio/vorbis, audio/x-aiff, audio/x-flac, audio/x-ms-wma, audio/x-wav - au, m4a, mp3, oga, asnd, ogg, aiff, flac, wma, wav

These formats cannot be transformed into, or generated from, any other format.

image/tiff - tiff

Format	Transformable to:	Transformable from:
application/eps	X	X
application/illustrator		X
application/vnd.msword		X
application/pdf	X	X

Format	Transformable to:	Transformable from:
application/rtf		X
application/vnd.apple.keynote		X
application/vnd.apple.numbers		X
application/vnd.apple.pages		X
application/vnd.ms-excel		X
application/vnd.ms-excel.addin.macroenabled.12		X
application/vnd.ms-excel.sheet.binary.macroenabled.12		X
application/vnd.ms-excel.sheet.macroenabled.12		X
application/vnd.ms-excel.template.macroenabled.12		X
application/vnd.ms-outlook		X
application/vnd.ms-powerpoint		X
application/vnd.ms-powerpoint.addin.macroenabled.12		X
application/vnd.ms-powerpoint.presentation.macroenabled.12		X
application/vnd.ms-powerpoint.slide.macroenabled.12		X
application/vnd.ms-powerpoint.slideshow.macroenabled.12		X
application/vnd.ms-powerpoint.template.macroenabled.12		X
application/vnd.ms-word.document.macroenabled.12		X
application/vnd.ms-word.template.macroenabled.12		X
application/vnd.oasis.opendocument.presentation		X
application/vnd.oasis.opendocument.presentation-template		X
application/vnd.oasis.opendocument.spreadsheet		X
application/vnd.oasis.opendocument.spreadsheet-template		X
application/vnd.oasis.opendocument.text		X
application/vnd.oasis.opendocument.text-template		X
application/vnd.openxmlformats-officedocument.presentationml.presentation		X
application/vnd.openxmlformats-officedocument.presentationml.slide		X
application/vnd.openxmlformats-officedocument.presentationml.slideshow		X
application/vnd.openxmlformats-officedocument.presentationml.template		X
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet		X
application/vnd.openxmlformats-officedocument.spreadsheetml.template		X
application/vnd.openxmlformats-officedocument.wordprocessingml.document		X
application/vnd.openxmlformats-officedocument.wordprocessingml.template		X
application/vnd.sun.xml.calc		X
application/vnd.sun.xml.calc.template		X
application/vnd.sun.xml.impress		X
application/vnd.sun.xml.impress.template		X
application/vnd.sun.xml.writer		X
application/vnd.sun.xml.writer.template		X
application/vnd.visio		X

Format	Transformable to:	Transformable from:
application/wordperfect		X
application/x-shockwave-flash	X	
image/bmp	X	
image/cgm	X	X
image/gif	X	X
image/ief	X	X
image/jp2	X	X
image/jpeg	X	X
image/png	X	X
image/tiff	X	X
image/vnd.adobe.photoshop	X	X
image/vnd.adobe.premiere	X	X
image/x-cmu-raster	X	X
image/x-dwt	X	X
image/x-portable-anymap	X	X
image/x-portable-bitmap	X	X
image/x-portable-graymap	X	X
image/x-portable-pixmap	X	X
image/x-raw-adobe	X	X
image/x-raw-canon	X	X
image/x-raw-fuji	X	X
image/x-raw-hasselblad	X	X
image/x-raw-kodak	X	X
image/x-raw-leica	X	X
image/x-raw-minolta	X	X
image/x-raw-nikon	X	X
image/x-raw-olympus	X	X
image/x-raw-panasonic	X	X
image/x-raw-pentax	X	X
image/x-raw-red	X	X
image/x-raw-sigma	X	X
image/x-raw-sony	X	X
image/x-xbitmap	X	X
image/x-ypixmap	X	X
image/x-xwindowdump	X	X
text/csv		X
text/html		X
text/plain		X
text/xml		X

image/bmp, image/cgm, image/gif, image/ief, image/jp2, image/jpeg, image/png, image/tiff, image/vnd.adobe.photoshop, image/vnd.adobe.premiere, image/x-cmu-raster, image/x-dwt, image/x-portable-anymap, image/x-portable-bitmap, image/x-portable-

graymap and image/x-portable-pixmap - bmp, cgm, gif, ief, jp2, jpg, png, tiff, psd, ppj, ras, dwt, pnm, pbm, pgm, ppm

All image types are transformable into and from the following formats, excepting themselves (i.e. image/bmp is not transformable into image/bmp, or from image/bmp).

Format	Transformable to:	Transformable from:
application/eps	X	X
application/illustrator		X
application/vnd.msword		X
application/pdf		X
application/rtf		X
application/vnd.apple.keynote		X
application/vnd.apple.numbers		X
application/vnd.apple.pages		X
application/vnd.ms-excel		X
application/vnd.ms-excel.addin.macroenabled.12		X
application/vnd.ms-excel.sheet.binary.macroenabled.12		X
application/vnd.ms-excel.sheet.macroenabled.12		X
application/vnd.ms-excel.template.macroenabled.12		X
application/vnd.ms-outlook		X
application/vnd.ms-powerpoint		X
application/vnd.ms-powerpoint.addin.macroenabled.12		X
application/vnd.ms-powerpoint.presentation.macroenabled.12		X
application/vnd.ms-powerpoint.slide.macroenabled.12		X
application/vnd.ms-powerpoint.slideshow.macroenabled.12		X
application/vnd.ms-powerpoint.template.macroenabled.12		X
application/vnd.ms-word.document.macroenabled.12		X
application/vnd.ms-word.template.macroenabled.12		X
application/vnd.oasis.opendocument.presentation		X
application/vnd.oasis.opendocument.presentation-template		X
application/vnd.oasis.opendocument.spreadsheet		X
application/vnd.oasis.opendocument.spreadsheet-template		X
application/vnd.oasis.opendocument.text		X
application/vnd.oasis.opendocument.text-template		X
application/vnd.openxmlformats-officedocument.presentationml.presentation		X
application/vnd.openxmlformats-officedocument.presentationml.slide		X
application/vnd.openxmlformats-officedocument.presentationml.slideshow		X
application/vnd.openxmlformats-officedocument.presentationml.template		X
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet		X
application/vnd.openxmlformats-officedocument.spreadsheetml.template		X
application/vnd.openxmlformats-officedocument.wordprocessingml.document		X

Format	Transformable to:	Transformable from:
application/vnd.openxmlformats-officedocument.wordprocessingml.template		X
application/vnd.sun.xml.calc		X
application/vnd.sun.xml.calc.template		X
application/vnd.sun.xml.impress		X
application/vnd.sun.xml.impress.template		X
application/vnd.sun.xml.writer		X
application/vnd.sun.xml.writer.template		X
application/vnd.visio		X
application/wordperfect		X
image/bmp	X	X
image/cgm	X	X
image/gif	X	X
image/ief	X	X
image/jp2	X	X
image/jpeg	X	X
image/png	X	X
image/tiff	X	X
image/vnd.adobe.photoshop	X	X
image/vnd.adobe.premiere	X	X
image/x-cmu-raster	X	X
image/x-dwt	X	X
image/x-portable-anymap	X	X
image/x-portable-bitmap	X	X
image/x-portable-graymap	X	X
image/x-portable-pixmap	X	X
image/x-raw-adobe	X	X
image/x-raw-canon	X	X
image/x-raw-fuji	X	X
image/x-raw-hasselblad	X	X
image/x-raw-kodak	X	X
image/x-raw-leica	X	X
image/x-raw-minolta	X	X
image/x-raw-nikon	X	X
image/x-raw-olympus	X	X
image/x-raw-panasonic	X	X
image/x-raw-pentax	X	X
image/x-raw-red	X	X
image/x-raw-sigma	X	X
image/x-raw-sony	X	X
image/x-xbitmap	X	X
image/x-ypixmap	X	X
image/x-xwindowdump	X	X
text/csv		X
text/html		X
text/plain		X

Format	Transformable to:	Transformable from:
text/xml		X

image/svg+xml, image/vnd.dwg, image/x-rgb - svg, dwg, rgb

These formats cannot be transformed into, or generated from, any other format.

image/x-raw-adobe, image/x-raw-canon, image/x-raw-fuji, image/x-raw-hasselblad, image/x-raw-kodak, image/x-raw-leica, image/x-raw-minolta, image/x-raw-nikon, image/x-raw-olympus, image/x-raw-panasonic, image/x-raw-pentax, image/x-raw-red, image/x-raw-sigma, image/x-raw-sony, image/x-xbitmap, image/x-xpixmap and image/x-xwindowdump - dng, cr2, raf, 3fr, k25, rwl, mrw, nef, orf, rw2, pef, r3d, x3f, arw, xbm, xpm, xwd

All image types are transformable into and from the following formats, excepting themselves (i.e. image/x-raw-adobe is not transformable into image/x-raw-adobe).

Format	Transformable to:	Transformable from:
application/eps	X	X
application/illustrator		X
application/vnd.msword		X
application/pdf		X
application/rtf		X
application/vnd.apple.keynote		X
application/vnd.apple.numbers		X
application/vnd.apple.pages		X
application/vnd.ms-excel		X
application/vnd.ms-excel.addin.macroenabled.12		X
application/vnd.ms-excel.sheet.binary.macroenabled.12		X
application/vnd.ms-excel.sheet.macroenabled.12		X
application/vnd.ms-excel.template.macroenabled.12		X
application/vnd.ms-outlook		X
application/vnd.ms-powerpoint		X
application/vnd.ms-powerpoint.addin.macroenabled.12		X
application/vnd.ms-powerpoint.presentation.macroenabled.12		X
application/vnd.ms-powerpoint.slide.macroenabled.12		X
application/vnd.ms-powerpoint.slideshow.macroenabled.12		X
application/vnd.ms-powerpoint.template.macroenabled.12		X
application/vnd.ms-word.document.macroenabled.12		X
application/vnd.ms-word.template.macroenabled.12		X
application/vnd.oasis.opendocument.presentation		X
application/vnd.oasis.opendocument.presentation-template		X
application/vnd.oasis.opendocument.spreadsheet		X
application/vnd.oasis.opendocument.spreadsheet-template		X
application/vnd.oasis.opendocument.text		X
application/vnd.oasis.opendocument.text-template		X
application/vnd.openxmlformats-officedocument.presentationml.presentation		X
application/vnd.openxmlformats-officedocument.presentationml.slide		X

Format	Transformable to:	Transformable from:
application/vnd.openxmlformats-officedocument.presentationml.slideshow		X
application/vnd.openxmlformats-officedocument.presentationml.template		X
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet		X
application/vnd.openxmlformats-officedocument.spreadsheetml.template		X
application/vnd.openxmlformats-officedocument.wordprocessingml.document		X
application/vnd.openxmlformats-officedocument.wordprocessingml.template		X
application/vnd.sun.xml.calc		X
application/vnd.sun.xml.calc.template		X
application/vnd.sun.xml.impress		X
application/vnd.sun.xml.impress.template		X
application/vnd.sun.xml.writer		X
application/vnd.sun.xml.writer.template		X
application/vnd.visio		X
application/wordperfect		X
image/bmp	X	X
image/cgm	X	X
image/gif	X	X
image/ief	X	X
image/jp2	X	X
image/jpeg	X	X
image/png	X	X
image/tiff	X	X
image/vnd.adobe.photoshop	X	X
image/vnd.adobe.premiere	X	X
image/x-cmu-raster	X	X
image/x-dwt	X	X
image/x-portable-anymap	X	X
image/x-portable-bitmap	X	X
image/x-portable-graymap	X	X
image/x-portable-pixmap	X	X
image/x-raw-adobe	X	X
image/x-raw-canon	X	X
image/x-raw-fuji	X	X
image/x-raw-hasselblad	X	X
image/x-raw-kodak	X	X
image/x-raw-leica	X	X
image/x-raw-minolta	X	X
image/x-raw-nikon	X	X
image/x-raw-olympus	X	X
image/x-raw-panasonic	X	X
image/x-raw-pentax	X	X

Format	Transformable to:	Transformable from:
image/x-raw-red	X	X
image/x-raw-sigma	X	X
image/x-raw-sony	X	X
image/x-xbitmap	X	X
image/x-pximap	X	X
image/x-xwindowdump	X	X
text/csv		X
text/html		X
text/plain		X
text/xml		X

message/rfc822 - eml

 This format cannot be generated from any other format.

Format	Transformable to:
application/x-shockwave-flash	X
text/plain	X

text/calendar, text/css, text/richtext, text/sgml, text/tab-separated-values, text/x-markdown and text/x-setext - ics, css, rtx, sgml, tsv, md, etc

 These formats cannot be generated from any other format.

Format	Transformable to:
text/plain	X

text/csv - csv

Format	Transformable to:	Transformable from:
application/eps	X	
application/pdf	X	
application/vnd.ms-excel		X
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet		X
application/x-shockwave-flash	X	
image/bmp	X	
image/cgm	X	
image/gif	X	
image/ief	X	
image/jp2	X	
image/jpeg	X	
image/png	X	
image/tiff	X	
image/vnd.adobe.photoshop	X	
image/vnd.adobe.premiere	X	
image/x-cmu-raster	X	
image/x-dwt	X	
image/x-portable-anymap	X	
image/x-portable-bitmap	X	

Format	Transformable to:	Transformable from:
image/x-portable-graymap	X	
image/x-portable-pixmap	X	
image/x-raw-adobe	X	
image/x-raw-canon	X	
image/x-raw-fuji	X	
image/x-raw-hasselblad	X	
image/x-raw-kodak	X	
image/x-raw-leica	X	
image/x-raw-minolta	X	
image/x-raw-nikon	X	
image/x-raw-olympus	X	
image/x-raw-panasonic	X	
image/x-raw-pentax	X	
image/x-raw-red	X	
image/x-raw-sigma	X	
image/x-raw-sony	X	
image/x-xbitmap	X	
image/x-xpixmap	X	
image/x-xwindowdump	X	
text/plain	X	

text/html - html

Format	Transformable to:	Transformable from:
application/eps	X	
application/vnd.msword	X	X
application/ogg		X
application/pdf	X	X
application/rss+xml		X
application/rtf	X	X
application/vnd.apple.keynote		X
application/vnd.apple.numbers		X
application/vnd.apple.pages		X
application/vnd.ms-excel		X
application/vnd.ms-excel.addin.macroenabled.12		X
application/vnd.ms-excel.sheet.binary.macroenabled.12		X
application/vnd.ms-excel.sheet.macroenabled.12		X
application/vnd.ms-excel.template.macroenabled.12		X
application/vnd.ms-outlook		X
application/vnd.ms-powerpoint		X
application/vnd.ms-powerpoint.addin.macroenabled.12		X
application/vnd.ms-powerpoint.presentation.macroenabled.12		X
application/vnd.ms-powerpoint.slide.macroenabled.12		X
application/vnd.ms-powerpoint.slideshow.macroenabled.12		X
application/vnd.ms-powerpoint.template.macroenabled.12		X
application/vnd.ms-project		X
application/vnd.ms-word.document.macroenabled.12		X

Format	Transformable to:	Transformable from:
application/vnd.ms-word.template.macroenabled.12		X
application/vnd.oasis.opendocument.chart		X
application/vnd.oasis.opendocument.image		X
application/vnd.oasis.opendocument.presentation		X
application/vnd.oasis.opendocument.presentation-template		X
application/vnd.oasis.opendocument.spreadsheet		X
application/vnd.oasis.opendocument.spreadsheet-template		X
application/vnd.oasis.opendocument.text	X	X
application/vnd.oasis.opendocument.text-master		X
application/vnd.oasis.opendocument.text-template	X	X
application/vnd.oasis.opendocument.text-web		X
application/vnd.openxmlformats-officedocument.presentationml.presentation		X
application/vnd.openxmlformats-officedocument.presentationml.slide		X
application/vnd.openxmlformats-officedocument.presentationml.slideshow		X
application/vnd.openxmlformats-officedocument.presentationml.template		X
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet		X
application/vnd.openxmlformats-officedocument.spreadsheetml.template		X
application/vnd.openxmlformats-officedocument.wordprocessingml.document		X
application/vnd.openxmlformats-officedocument.wordprocessingml.template		X
application/vnd.sun.xml.calc		X
application/vnd.sun.xml.calc.template		X
application/vnd.sun.xml.impress		X
application/vnd.sun.xml.impress.template		X
application/vnd.sun.xml.writer	X	X
application/vnd.sun.xml.writer.template	X	X
application/vnd.visio		X
application/wordperfect		X
application/x-cpio		X
application/x-gzip		X
application/x-hdf		X
application/x-netcdf		X
application/x-shockwave-flash	X	
application/x-tar		X
application/xhtml+xml	X	X
application/zip		X
image/bmp	X	
image/cgm	X	
image/gif	X	
image/ief	X	

Format	Transformable to:	Transformable from:
image/jp2	X	
image/jpeg	X	
image/png	X	
image/tiff	X	
image/vnd.adobe.photoshop	X	
image/vnd.adobe.premiere	X	
image/x-cmu-raster	X	
image/x-dwt	X	
image/x-portable-anymap	X	
image/x-portable-bitmap	X	
image/x-portable-graymap	X	
image/x-portable-pixmap	X	
image/x-raw-adobe	X	
image/x-raw-canon	X	
image/x-raw-fuji	X	
image/x-raw-hasselblad	X	
image/x-raw-kodak	X	
image/x-raw-leica	X	
image/x-raw-minolta	X	
image/x-raw-nikon	X	
image/x-raw-olympus	X	
image/x-raw-panasonic	X	
image/x-raw-pentax	X	
image/x-raw-red	X	
image/x-raw-sigma	X	
image/x-raw-sony	X	
image/x-xbitmap	X	
image/x-xpixmap	X	
image/x-xwindowdump	X	
text/mediawiki		X
text/plain	X	X
text/xml	X	X

text/mediawiki - mw



This format cannot be generated from any other format.

Format	Transformable to:
text/html	X
text/plain	X

text/plain - txt

Format	Transformable to:	Transformable from:
application/eps	X	
application/vnd.msword	X	X
application/ogg		X

Format	Transformable to:	Transformable from:
application/pdf	X	X
application/rss+xml		X
application/rtf	X	X
application/vnd.apple.keynote		X
application/vnd.apple.numbers		X
application/vnd.apple.pages		X
application/vnd.ms-excel		X
application/vnd.ms-excel.addin.macroenabled.12		X
application/vnd.ms-excel.sheet.binary.macroenabled.12		X
application/vnd.ms-excel.sheet.macroenabled.12		X
application/vnd.ms-excel.template.macroenabled.12		X
application/vnd.ms-outlook		X
application/vnd.ms-powerpoint		X
application/vnd.ms-powerpoint.addin.macroenabled.12		X
application/vnd.ms-powerpoint.presentation.macroenabled.12		X
application/vnd.ms-powerpoint.slideshow.macroenabled.12		X
application/vnd.ms-powerpoint.template.macroenabled.12		X
application/vnd.ms-project		X
application/vnd.ms-word.document.macroenabled.12		X
application/vnd.ms-word.template.macroenabled.12		X
application/vnd.oasis.opendocument.chart		X
application/vnd.oasis.opendocument.image		X
application/vnd.oasis.opendocument.presentation		X
application/vnd.oasis.opendocument.presentation-template		X
application/vnd.oasis.opendocument.spreadsheet		X
application/vnd.oasis.opendocument.spreadsheet-template		X
application/vnd.oasis.opendocument.text	X	X
application/vnd.oasis.opendocument.text-master		X
application/vnd.oasis.opendocument.text-template	X	X
application/vnd.oasis.opendocument.text-web		X
application/vnd.openxmlformats-officedocument.presentationml.presentation		X
application/vnd.openxmlformats-officedocument.presentationml.slideshow		X
application/vnd.openxmlformats-officedocument.presentationml.template		X
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet		X
application/vnd.openxmlformats-officedocument.spreadsheetml.template		X
application/vnd.openxmlformats-officedocument.wordprocessingml.document		X
application/vnd.openxmlformats-officedocument.wordprocessingml.template		X
application/vnd.sun.xml.calc		X
application/vnd.sun.xml.calc.template		X
application/vnd.sun.xml.impress		X

Format	Transformable to:	Transformable from:
application/vnd.sun.xml.impress.template		X
application/vnd.sun.xml.writer	X	X
application/vnd.sun.xml.writer.template	X	X
application/vnd.visio		X
application/wordperfect		X
application/x-cpio		X
application/x-gzip		X
application/x-hdf		X
application/x-javascript		X
application/x-netcdf		X
application/x-shockwave-flash	X	
application/x-tar		X
application/xhtml+xml	X	X
application/zip		X
image/bmp	X	
image/cgm	X	
image/gif	X	
image/ief	X	
image/jp2	X	
image/jpeg	X	
image/png	X	
image/tiff	X	
image/vnd.adobe.photoshop	X	
image/vnd.adobe.premiere	X	
image/x-cmu-raster	X	
image/x-dwt	X	
image/x-portable-anymap	X	
image/x-portable-bitmap	X	
image/x-portable-graymap	X	
image/x-portable-pixmap	X	
image/x-raw-adobe	X	
image/x-raw-canon	X	
image/x-raw-fuji	X	
image/x-raw-hasselblad	X	
image/x-raw-kodak	X	
image/x-raw-leica	X	
image/x-raw-minolta	X	
image/x-raw-nikon	X	
image/x-raw-olympus	X	
image/x-raw-panasonic	X	
image/x-raw-pentax	X	
image/x-raw-red	X	
image/x-raw-sigma	X	
image/x-raw-sony	X	
image/x-xbitmap	X	
image/x-xpixmap	X	

Format	Transformable to:	Transformable from:
image/x-xwindowdump	X	
message/rfc822		X
text/calendar		X
text/css		X
text/csv		X
text/html	X	X
text/mediawiki		X
text/richtext		X
text/sgml		X
text/tab-separated-values		X
text/x-markdown		X
text/x-setext		X
text/xml	X	X

text/xml - xml

Format	Transformable to:	Transformable from:
application/eps	X	
application/vnd.msword		X
application/ogg		X
application/pdf	X	X
application/rss+xml		X
application/rtf		X
application/vnd.apple.keynote		X
application/vnd.apple.numbers		X
application/vnd.apple.pages		X
application/vnd.ms-excel		X
application/vnd.ms-excel.addin.macroenabled.12		X
application/vnd.ms-excel.sheet.binary.macroenabled.12		X
application/vnd.ms-excel.sheet.macroenabled.12		X
application/vnd.ms-excel.template.macroenabled.12		X
application/vnd.ms-outlook		X
application/vnd.ms-powerpoint		X
application/vnd.ms-powerpoint.addin.macroenabled.12		X
application/vnd.ms-powerpoint.presentation.macroenabled.12		X
application/vnd.ms-powerpoint.slide.macroenabled.12		X
application/vnd.ms-powerpoint.slideshow.macroenabled.12		X
application/vnd.ms-powerpoint.template.macroenabled.12		X
application/vnd.ms-project		X
application/vnd.ms-word.document.macroenabled.12		X
application/vnd.ms-word.template.macroenabled.12		X
application/vnd.oasis.opendocument.chart		X
application/vnd.oasis.opendocument.image		X
application/vnd.oasis.opendocument.presentation		X
application/vnd.oasis.opendocument.presentation-template		X
application/vnd.oasis.opendocument.spreadsheet		X

Format	Transformable to:	Transformable from:
application/vnd.oasis.opendocument.spreadsheet-template		X
application/vnd.oasis.opendocument.text		X
application/vnd.oasis.opendocument.text-master		X
application/vnd.oasis.opendocument.text-template		X
application/vnd.oasis.opendocument.text-web		X
application/vnd.openxmlformats-officedocument.presentationml.presentation		X
application/vnd.openxmlformats-officedocument.presentationml.slide		X
application/vnd.openxmlformats-officedocument.presentationml.slideshow		X
application/vnd.openxmlformats-officedocument.presentationml.template		X
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet		X
application/vnd.openxmlformats-officedocument.spreadsheetml.template		X
application/vnd.openxmlformats-officedocument.wordprocessingml.document		X
application/vnd.openxmlformats-officedocument.wordprocessingml.template		X
application/vnd.sun.xml.calc		X
application/vnd.sun.xml.calc.template		X
application/vnd.sun.xml.impress		X
application/vnd.sun.xml.impress.template		X
application/vnd.sun.xml.writer		X
application/vnd.sun.xml.writer.template		X
application/vnd.visio		X
application/wordperfect		X
application/x-cpio		X
application/x-gzip		X
application/x-hdf		X
application/x-netcdf		X
application/x-shockwave-flash	X	
application/x-tar		X
application/xhtml+xml	X	X
application/zip		X
image/bmp	X	
image/cgm	X	
image/gif	X	
image/ief	X	
image/jp2	X	
image/jpeg	X	
image/png	X	
image/tiff	X	
image/vnd.adobe.photoshop	X	
image/vnd.adobe.premiere	X	

Format	Transformable to:	Transformable from:
image/x-cmu-raster	X	
image/x-dwt	X	
image/x-portable-anymap	X	
image/x-portable-bitmap	X	
image/x-portable-graymap	X	
image/x-portable-pixmap	X	
image/x-raw-adobe	X	
image/x-raw-canon	X	
image/x-raw-fuji	X	
image/x-raw-hasselblad	X	
image/x-raw-kodak	X	
image/x-raw-leica	X	
image/x-raw-minolta	X	
image/x-raw-nikon	X	
image/x-raw-olympus	X	
image/x-raw-panasonic	X	
image/x-raw-pentax	X	
image/x-raw-red	X	
image/x-raw-sigma	X	
image/x-raw-sony	X	
image/x-xbitmap	X	
image/x-xpixmap	X	
image/x-xwindowdump	X	
text/html	X	X
text/plain	X	X

video/3gpp, video/3gpp2, video/mp2t, video/mp4, video/mpeg, video/mpeg2, video/ogg, video/quicktime, video/webm, video/x-flv, video/x-m4v, video/x-ms-asf, video/x-ms-wmv, video/x-msvideo, video/x-rad-screenplay, video/x-sgi-movie, x-world/x-vrml - 3gp, 3g2, ts, mp4, mpg, mpeg2, ogv, mov, webm, flv, m4v, asf, wmv, avi, avx, movie, wrl

These formats cannot be transformed into, or generated from, any other format.

Additional transform options

If you have installed a transformation tool, such as Transformation Server, there are additional transform options available to you.

The tables give details of registered file types with information about their available transform options. See [Standard transform options](#) on page 224 for all standard transform options in Alfresco.

You can also view more information about file types and the proxies used to transform them by using the browser command:

```
localhost:8080/alfresco/service/mimetypes?mimetype=*
```

where `localhost:8080` is the host and port number of your active Alfresco instance.

Alfresco Outlook Integration

Alfresco Outlook provides a transformer (`com.westernacher.wps.alfresco.transformers.mail.aspose.EML_MSG2PdfTransformer`) to

manipulate PDF and Outlook email messages. The formats listed are in addition to the standard formats as specified in [Standard transform options](#) on page 224.

application/pdf - pdf

Format	Transformable from:
application/vnd.ms-outlook	X
message/rfc822	X

application/vnd.ms-outlook - msg

Format	Transformable to:
application/pdf	X

message/rfc822 - eml

Format	Transformable to:
application/pdf	X

Transformation Server

Transformation Server gives an alternative method of remote transformation for a range of applications including pdf, swf, Word, Excel, Powerpoint and openxmlformats. It also supports a range of image transformations. The formats listed are in addition to the standard formats as specified in [Standard transform options](#) on page 224.

application/pdf - pdf

Format	Transformable from:
application/vnd.ms-powerpoint.slideshow.macroenabled.12	X
application/vnd.openxmlformats-officedocument.presentationml.slideshow	X

application/vnd.ms-powerpoint.slideshow.macroenabled.12 - ppsm, application/vnd.openxmlformats-officedocument.presentationml.slideshow - ppsx

Format	Transformable to:
application/pdf	X
application/x-shockwave-flash	X

application/x-shockwave-flash - swf

Format	Transformable from:
application/vnd.ms-powerpoint.slideshow.macroenabled.12	X
application/vnd.openxmlformats-officedocument.presentationml.slideshow	X

Media Management module

The Media Management module provides enhanced video, audio and metadata capabilities. Video files are transformable using image/jpeg, and audio and video files are transformable using ffmpeg (org.alfresco.repo.content.transform.RuntimeExecutableContentTransformerWorker). The formats listed are in addition to the standard formats as specified in [Standard transform options](#) on page 224.

application/eps - eps, image/bmp - bmp, image/cgm - cgm, image/gif - gif, image/ief - ief, image/jp2 - jp2, image/jpeg - jpg, image/png - png, image/tiff - tiff, image/vnd.adobe.photoshop - psd, image/vnd.adobe.premiere - ppj, image/x-dwg - dwg,

image/x-dwt - dwt, image/x-portable-anymap - pnm, image/x-portable-bitmap - pbm, image/x-portable-graymap - pgm, image/x-portable-pixmap - ppm, image/x-raw-adobe - dng, image/x-raw-canon - cr2, image/x-raw-fuji - raf, image/x-raw-hasselblad - 3fr, image/x-raw-kodak - k25, image/x-raw-leica - rwl, image/x-raw-minolta - mrw, image/x-raw-nikon - nef, image/x-raw-olympus - orf, image/x-raw-panasonic - rw2, image/x-raw-pentax - pef, image/x-raw-red - r3d, image/x-raw-sigma - x3f, image/x-raw-sony - arw, image/x-rgb - rgb, image/x-xpixmap - xpm and image/x-xwindowdump - xwd

image/jpeg uses ffmpeg through

org.alfresco.repo.content.transform.RuntimeExecutableContentTransformerWorker.

Format	Transformable from:
video/3gpp	X
video/3gpp2	X
video/mp4	X
video/mpeg	X
video/mpeg2	X
video/ogg	X
video/quicktime	X
video/x-flv	X
video/x-ms-asf	X
video/x-ms-wmv	X
video/x-msvideo	X
video/x-rad-screenplay	X
video/x-sgi-movie	X

audio/basic - au, audio/mp4 - m4a, audio/ogg - oga, audio/x-aiff - aiff, audio/x-wav - wav



These formats cannot be generated from any other format.

Format	Transformable to:
audio/mpeg	X

audio/mpeg - mp3



This format cannot be transformed into any other format.

Format	Transformable from:
audio/basic	X

audio/vnd.adobe.soundbooth - asnd, audio/vorbis - ogg, audio/x-flac - flac, audio/x-ms-wma, video/webm - webm, video/x-m4v - m4v, video/mp2t - ts and x-world/x-vrml - wrl



These formats cannot be transformed into, or generated from, any other format.

video/mp4 - mp4

Format	Transformable to:	Transformable from:
application/eps	X	
image/bmp	X	
image/cgm	X	
image/gif	X	
image/ief	X	
image/jp2	X	

Format	Transformable to:	Transformable from:
image/jpeg	X	
image/png	X	
image/tiff	X	
image/vnd.adobe.photoshop	X	
image/vnd.adobe.premiere	X	
image/x-cmu-raster	X	
image/x-dwt	X	
image/x-portable-anymap	X	
image/x-portable-bitmap	X	
image/x-portable-graymap	X	
image/x-portable-pixmap	X	
image/x-raw-adobe	X	
image/x-raw-canon	X	
image/x-raw-fuji	X	
image/x-raw-hasselblad	X	
image/x-raw-kodak	X	
image/x-raw-leica	X	
image/x-raw-minolta	X	
image/x-raw-nikon	X	
image/x-raw-olympus	X	
image/x-raw-panasonic	X	
image/x-raw-pentax	X	
image/x-raw-red	X	
image/x-raw-sigma	X	
image/x-raw-sony	X	
image/x-xbitmap	X	
image/x-xpixmap	X	
image/x-xwindowdump	X	
video/x-flv	X	
video/3gpp		X
video/3gpp2		X
video/mpeg		X
video/mpeg2		X
video/quicktime		X
video/x-ms-asf		X
video/x-ms-wmv		X
video/x-rad-screenplay		X
video/x-sgi-movie		X

video/ogg - ogv and video/x-msvideo - avi



These formats cannot be generated from any other format.

Format	Transformable to:
application/eps	X
image/bmp	X
image/cgm	X
image/gif	X

Format	Transformable to:
image/ief	X
image/jp2	X
image/jpeg	X
image/png	X
image/tiff	X
image/vnd.adobe.photoshop	X
image/vnd.adobe.premiere	X
image/x-cmu-raster	X
image/x-dwt	X
image/x-portable-anymap	X
image/x-portable-bitmap	X
image/x-portable-graymap	X
image/x-portable-pixmap	X
image/x-raw-adobe	X
image/x-raw-canon	X
image/x-raw-fuji	X
image/x-raw-hasselblad	X
image/x-raw-kodak	X
image/x-raw-leica	X
image/x-raw-minolta	X
image/x-raw-nikon	X
image/x-raw-olympus	X
image/x-raw-panasonic	X
image/x-raw-pentax	X
image/x-raw-red	X
image/x-raw-sigma	X
image/x-raw-sony	X
image/x-xbitmap	X
image/x-ypixmap	X
image/x-xwindowdump	X
video/x-flv	X

video/mpeg - mpg, video/mpeg2 - mpeg2, video/3gpp - 3gp and video/3gpp2 - 3g2, video/quicktime - mov, video/mpeg2 - mpeg2, video/x-m4v - m4v, video/x-ms-asf - asf video/x-ms-wmv - wmv, video/x-rad-screenplay - avx and video/x-sgi-movie - movie



These formats cannot be generated from any other format.

Format	Transformable to:
application/eps	X
image/bmp	X
image/cgm	X
image/gif	X
image/ief	X
image/jp2	X
image/jpeg	X
image/png	X
image/tiff	X

Format	Transformable to:
image/vnd.adobe.photoshop	X
image/vnd.adobe.premiere	X
image/x-cmu-raster	X
image/x-dwt	X
image/x-portable-anymap	X
image/x-portable-bitmap	X
image/x-portable-graymap	X
image/x-portable-pixmap	X
image/x-raw-adobe	X
image/x-raw-canon	X
image/x-raw-fuji	X
image/x-raw-hasselblad	X
image/x-raw-kodak	X
image/x-raw-leica	X
image/x-raw-minolta	X
image/x-raw-nikon	X
image/x-raw-olympus	X
image/x-raw-panasonic	X
image/x-raw-pentax	X
image/x-raw-red	X
image/x-raw-sigma	X
image/x-raw-sony	X
image/x-xbitmap	X
image/x-xpixmap	X
image/x-xwindowdump	X
video/mp4	X
video/x-flv	X

video/x-flv - flv

Format	Transformable to:	Transformable from:
application/eps	X	
image/bmp	X	
image/cgm	X	
image/gif	X	
image/ief	X	
image/jp2	X	
image/jpeg	X	
image/png	X	
image/tiff	X	
image/vnd.adobe.photoshop	X	
image/vnd.adobe.premiere	X	
image/x-cmu-raster	X	
image/x-dwt	X	
image/x-portable-anymap	X	
image/x-portable-bitmap	X	
image/x-portable-graymap	X	
image/x-portable-pixmap	X	

Format	Transformable to:	Transformable from:
image/x-raw-adobe	X	
image/x-raw-canon	X	
image/x-raw-fuji	X	
image/x-raw-hasselblad	X	
image/x-raw-kodak	X	
image/x-raw-leica	X	
image/x-raw-minolta	X	
image/x-raw-nikon	X	
image/x-raw-olympus	X	
image/x-raw-panasonic	X	
image/x-raw-pentax	X	
image/x-raw-red	X	
image/x-raw-sigma	X	
image/x-raw-sony	X	
image/x-xbitmap	X	
image/x-pixmap	X	
image/x-xwindowdump	X	
video/3gpp		X
video/3gpp2		X
video/mp4		X
video/mpeg		X
video/mpeg2		X
video/quicktime		X
video/x-ms-asf		X
video/x-ms-wmv		X
video/x-rad-screenplay		X
video/x-sgi-movie		X

File types that support preview and thumbnail generation

Some file type extensions (MIME types) allow thumbnail or preview generation in Alfresco, instead of standard icons. If the file type can be transformed to application/x-shockwave-flash (swf) format, then it supports preview generation. If the file type can be transformed to image/jpeg (jpg) format, then it supports thumbnail generation.

The following table shows the file types that support these capabilities. See [Standard transform options](#) on page 224 for a full listing of formats.

Formats that support preview and thumbnail generation

Format	File type extension	Supports preview (swf)	Supports thumbnail (jpg)
application/eps	eps		X
application/illustrator	ai	X	X
application/msword	doc	X	X
application/pdf	pdf	X	X
application/rtf	rtf	X	X
application/vnd.apple.keynote	key	X	X
application/vnd.apple.numbers	numbers	X	X
application/vnd.apple.pages	pages	X	X

Format	File type extension	Supports preview (swf)	Supports thumbnail (jpg)
application/vnd.ms-excel	xls	X	X
application/vnd.ms-excel.addin.macroenabled.12	xlam		X
application/vnd.ms-excel.sheet.binary.macroenabled.12	xlsb	X	X
application/vnd.ms-excel.sheet.macroenabled.12	xlsm	X	X
application/vnd.ms-excel.template.macroenabled.12	xltm	X	X
application/vnd.ms-outlook	msg	X	X
application/vnd.ms-powerpoint	ppt	X	X
application/vnd.ms-powerpoint.addin.macroenabled.12	ppam	X	X
application/vnd.ms-powerpoint.presentation.macroenabled.12	pptm	X	X
application/vnd.ms-powerpoint.slide.macroenabled.12	sldm	X	X
application/vnd.ms-powerpoint.slideshow.macroenabled.12	ppsm		X
application/vnd.ms-powerpoint.template.macroenabled.12	potm	X	X
application/vnd.ms-word.document.macroenabled.12	docm	X	X
application/vnd.ms-word.template.macroenabled.12	dotm	X	X
application/vnd.oasis.opendocument.presentation	odp	X	X
application/vnd.oasis.opendocument.presentation-template	otp	X	X
application/vnd.oasis.opendocument.spreadsheet	ods	X	X
application/vnd.oasis.opendocument.spreadsheet-template	ots	X	X
application/vnd.oasis.opendocument.text	odt	X	X
application/vnd.oasis.opendocument.text-template	ott	X	X
application/vnd.openxmlformats-officedocument.presentationml.presentation	pptx	X	X
application/vnd.openxmlformats-officedocument.presentationml.slide	sldx	X	X
application/vnd.openxmlformats-officedocument.presentationml.slideshow	ppsx		X
application/vnd.openxmlformats-officedocument.presentationml.template	potx	X	X
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	xlsx	X	X
application/vnd.openxmlformats-officedocument.spreadsheetml.template	xltx	X	X
application/vnd.openxmlformats-officedocument.wordprocessingml.document	docx	X	X
application/vnd.openxmlformats-officedocument.wordprocessingml.template	dotx	X	X
application/vnd.sun.xml.calc	sxc	X	X
application/vnd.sun.xml.calc.template	stc	X	X
application/vnd.sun.xml.impress	sxi	X	X
application/vnd.sun.xml.impress.template	sti	X	X
application/vnd.sun.xml.writer	sxw	X	X
application/vnd.sun.xml.writer.template	stw	X	X
application/vnd.visio	vsd	X	X

Format	File type extension	Supports preview (swf)	Supports thumbnail (jpg)
application/wordperfect	wpd	X	X
application/x-cpio	cpio	X	
application/x-tar	tar	X	
application/zip	zip	X	
image/bmp	bmp		X
image/cgm	cgm		X
image/gif	gif		X
image/ief	ief		X
image/jp2	jp2		X
image/jpeg	jpg		X
image/png	png		X
image/tiff	tiff	X	X
image/vnd.adobe.photoshop	psd		X
image/vnd.adobe.premiere	ppj		X
image/x-cmu-raster	ras		X
image/x-dwt	dwt		X
image/x-portable-anymap	pnm		X
image/x-portable-bitmap	pbm		X
image/x-portable-graymap	pgm		X
image/x-portable-pixmap	ppm		X
image/x-raw-adobe	dng		X
image/x-raw-canon	cr2		X
image/x-raw-fuji	raf		X
image/x-raw-hasselblad	3fr		X
image/x-raw-kodak	k25		X
image/x-raw-leica	rwl		X
image/x-raw-minolta	mrw		X
image/x-raw-nikon	nef		X
image/x-raw-olympus	orf		X
image/x-raw-panasonic	rw2		X
image/x-raw-pentax	pef		X
image/x-raw-red	r3d		X
image/x-raw-sigma	x3f		X
image/x-raw-sony	arw		X
image/x-xbitmap	xbm		X
image/x-xpixmap	xpm		X
image/x-xwindowdump	xwd		X
text/csv	csv	X	X
text/html	html	X	X
text/plain	txt	X	X
text/xml	xml	X	X

Creating and managing workflows

What is a workflow?

For example, you might have a document that you needed reviewing and approving by a number of people. The sequence of connected tasks would be:

- Send an email to each reviewer asking them to review the document within a certain time
- Each reviewer reviews the document
- Each reviewer approves or rejects the document
- If enough reviewers approve, the task is completed successfully

Alfresco workflows automate the process for you. Users can choose from five workflow definitions provided in Alfresco. You can also create your own workflow definitions for more complex workflows. The five supplied workflow definitions are:

Adhoc

Enables you to assign a task to a single user

Group Review & Approve

Enables you to set up review and approval of content, assigning the workflow task to a single group

Parallel Review & Approve

Enables you to set up review and approval of content, assigning the workflow task to multiple users.

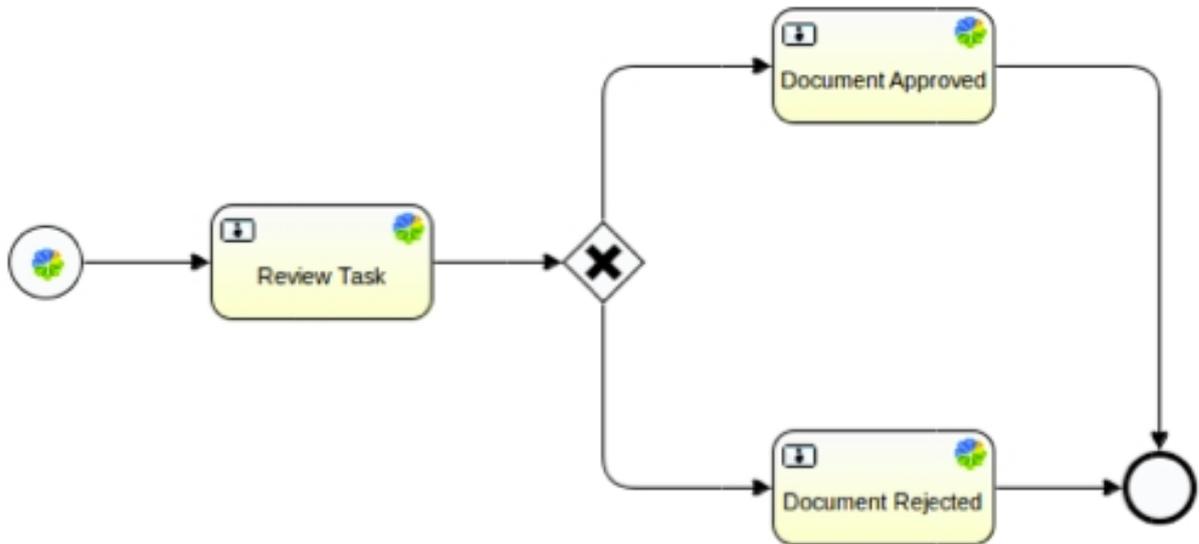
Pooled Review & Approve

Enables you to set up review and approval of content, assigning the workflow task to multiple users. One user can take ownership of the task at a time, completing it or returning it to the pool to be claimed by another user associated with the task.

Review & Approve

Enables you to set up review and approval of content, assigning the workflow task to a single user

A graphical workflow modeler is often used to create a workflow. The following diagram shows a sample workflow taken from the workflow modeler running in Eclipse. The workflow consists of three tasks, a gate, and two events; start and end.



The Alfresco Activiti workflow engine executes BPMN 2.0 process definitions. BPMN 2.0 (Business Process Model and Notation) is an open standard developed by the Object Management Group (OMG) to provide a notation that is easily understandable by all business users: business analysts designing processes, developers implementing technology to perform those processes, and, business people managing and monitoring those processes. BPMN creates a standardized bridge for the gap between the business process design and process.

Standard BPMN 2.0 process definition models can be exchanged between graphical editors, and executed on any BPMN 2.0 compliant engine. Be aware that if you use technology specific features in your definition, you will not be able to use that workflow on a different technology. For example, if you define an Activiti workflow to work with Alfresco, you will not be able to run it on a TIBCO server.

The following image shows part of a BPMN 2.0 process definition:

```

<process id="activitiInvitationModerated" name="Moderated activiti invitation process">
  <startEvent id="start" activiti:formKey="imwf:moderatedInvitationSubmitTask" />
  <sequenceFlow id="flow1" sourceRef="start" targetRef="reviewTask" />
  <userTask id="reviewTask" name="Review Task"
    activiti:formKey="imwf:activitiModeratedInvitationReviewTask">
    <extensionElements>
      <activiti:taskListener event="create"
        class="org.alfresco.repo.workflow.activiti.tasklistener.ScriptTaskListener">
        <activiti:field name="script">
          <activiti:string>
            if (typeof bpm_workflowDueDate != 'undefined')
              task.setVariable('bpm_dueDate', bpm_workflowDueDate);
            if (typeof bpm_workflowPriority != 'undefined')
              task.priority = bpm_workflowPriority;
          </activiti:string>
        </activiti:field>
      </activiti:taskListener>
      <activiti:taskListener event="complete" class="org.alfresco.repo.workflow.activiti.tasklistener.
        <activiti:field name="script">
          <activiti:string>
            execution.setVariable('imwf_reviewOutcome', task.getVariable('imwf_reviewOutcome'));
            execution.setVariable('imwf_reviewer', person.properties.userName);
          </activiti:string>
        </activiti:field>
      </activiti:taskListener>
    </extensionElements>
  </userTask>
  ....

```

Setting up Hybrid workflow

When a workflow is initiated on-premise for a document (for example, for a task for a review), this automatically synchronizes to the cloud and triggers a cloud workflow associated with the document. When the cloud process is complete, the workflow then returns to the on-premise workflow.

Hybrid workflow allows you to collaborate with associates and partners without giving access to your Alfresco on-premise behind the firewall. Tasks can be more easily managed and directed by being able to call for explicit actions, rather than simply sharing a document.

To set up and using Hybrid workflow requires:

- An Alfresco license that includes access to Hybrid workflow
- Access to an appropriate Cloud network

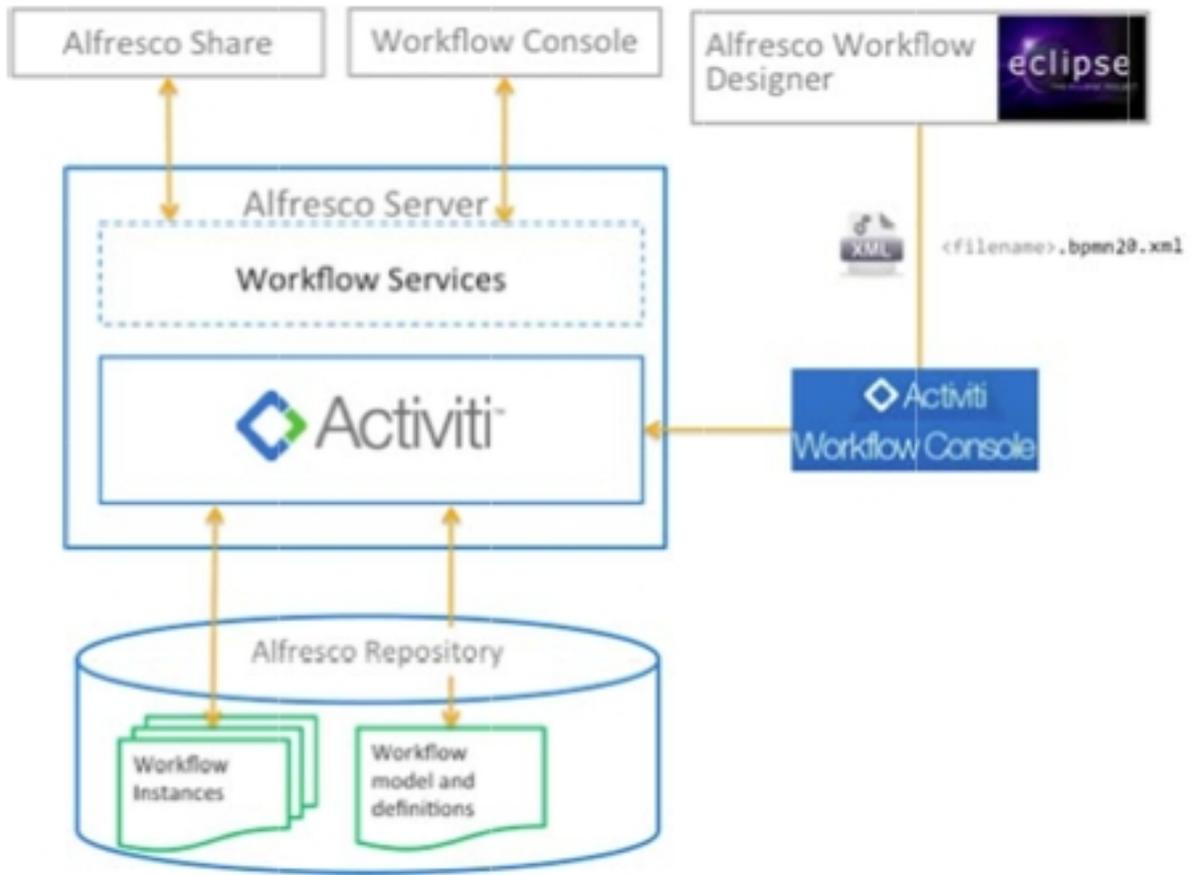
Enabling Hybrid workflow

1. Apply a suitable Enterprise license that includes Hybrid workflow to your Alfresco installation.
2. Ensure that you enable Enterprise to Cloud Sync.
3. Open the `alfresco.global.properties` file.
4. Add the following property:


```
hybridworkflow.enabled=true
```
5. Save the file.
6. Restart the Alfresco server.

Workflow architecture

The following figure shows the high-level architecture for Alfresco workflow.

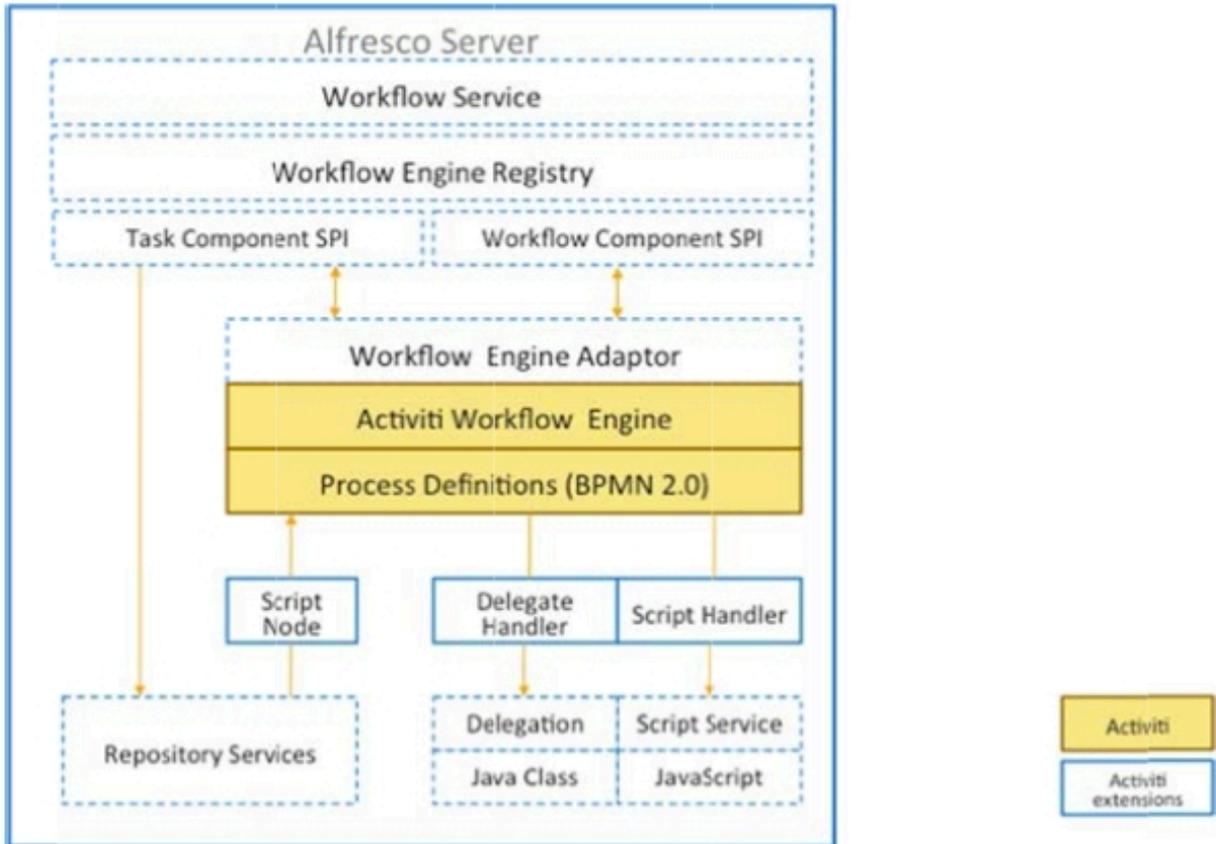


You can design workflow definitions using a graphical workflow designer that supports BPMN 2.0 or write the XML BPMN 2.0 process definition directly using an XML editor. Many workflow editors support BPMN 2.0 but might not understand some of the features of Alfresco workflow. We recommend the use of the Activiti eclipse designer plug-in for Eclipse that is Alfresco-aware.

You can deploy a workflow to Alfresco using the Activiti Workflow Console, or by using a Spring Bean.

Alfresco Activiti process definitions can include Alfresco JavaScript, and this in turn can access Alfresco content models in the repository so that you can provide your own specialized tasks for a workflow and access their properties. Process definitions have script node access which allows you to access objects in the Alfresco repository, such as documents and folders. Your workflow can access and modify document objects, for example marking documents as approved, or signed off.

Alfresco allows you to access your own Java Classes through the delegate handler, so you can integrate with other external systems. The following diagram show these features :-



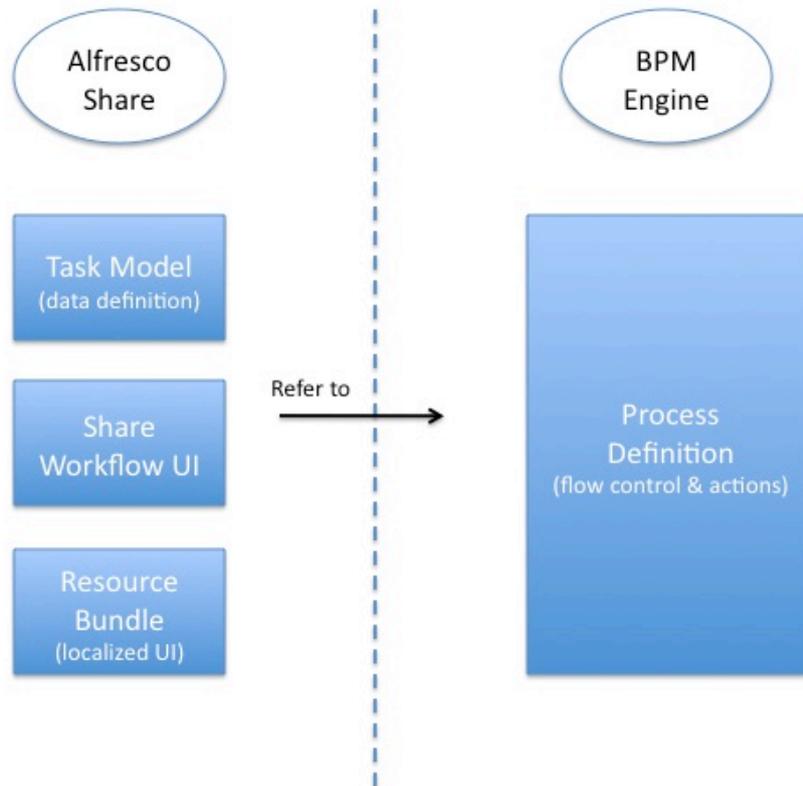
Workflow instances

Once a workflow instance has been started, it can not be changed. If you change the underlying process definition, it will be versioned. Any new workflow instance will reflect any changes to the workflow definition. Any old instances currently running will reference the old definition.

Workflow instances survive Alfresco server restarts, so all user tasks will still be running if you stop and restart the server. Process and task execution variables also survive Alfresco server restarts.

Workflow artifacts

The diagram shows the artifacts and the relationship between them:-



Process Definition

Activiti process definitions describe the events, activities (tasks) and gateways (choices) of a workflow. Tasks can be user tasks or script (system) tasks. User tasks are assigned to human performers (users). System tasks perform some kind of operation against the Alfresco repository. Both are described and implemented in the Process Definition.

Task Model

The Task Model provides a description for each of the user tasks in the workflow. Each task description consists of:

- Name and Title.
- Properties and Associations. For example, the information attached to the task.

The description is used to drive the user interface dialog for viewing and managing the Task. Alfresco provides a Data Dictionary for describing types of object to store, view and edit. This mechanism is also used to describe Workflow Tasks.

Share Workflow UI

You can customize the presentation of Tasks to the user in Alfresco Share. Customizing allows:

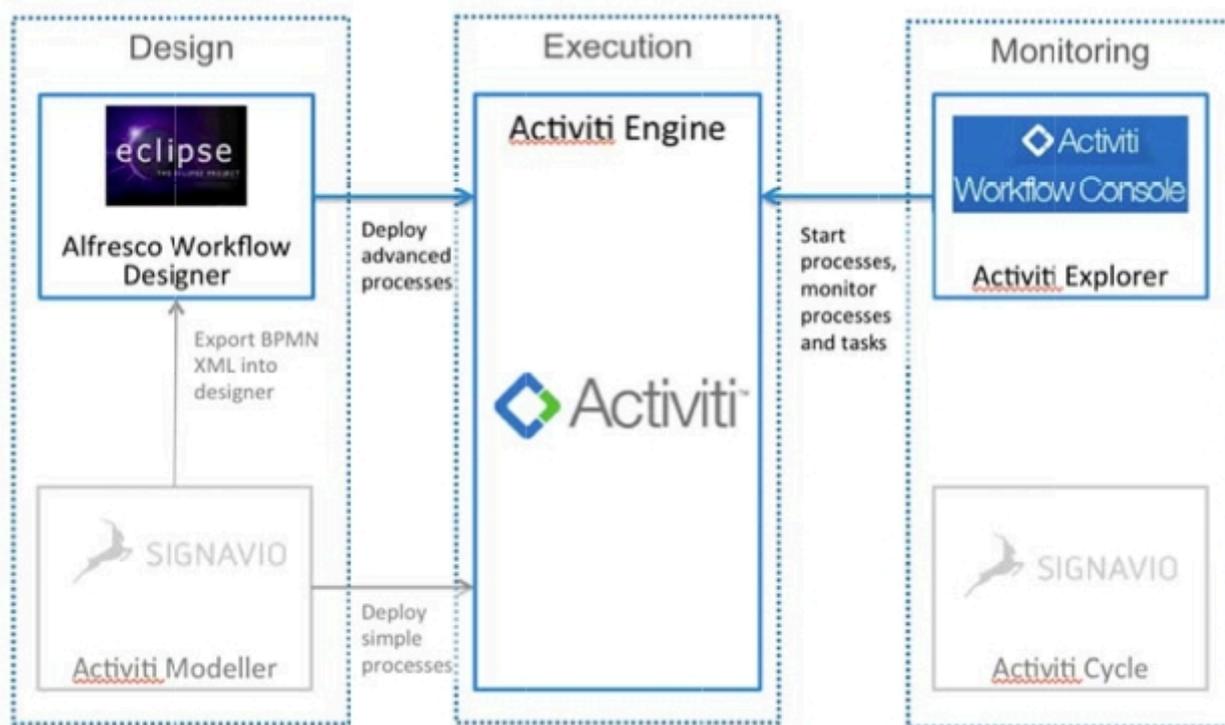
- Control over which Task properties are displayed
- Control over which Task properties are read-only and required
- Control over how each Task property is rendered in the forms

Resource Bundle (optional)

A workflow resource bundle provides all the human-readable messages displayed in the user interface for managing the workflow. Messages include Task titles, task property names, task choices etc. Alfresco supports full localization of Alfresco Share, including workflow. Therefore, the same Alfresco Share resource bundle configuration extends to workflow too.

Workflow tools

The following diagram shows the tools used in designing, executing, and monitoring an Alfresco workflow:



Activiti modeler

allows business and information analysts to model a BPMN 2.0 compliant business process in a web browser. This allows business processes to be shared, and no client software is needed before you can start modeling.

Activiti designer

is an Eclipse plugin, which enables a developer to enhance the model of the business process into a BPMN 2.0 process that can be executed on the Activiti process engine. You can also run unit tests, add Java logic, and create deployment artifacts with the Activiti Designer.

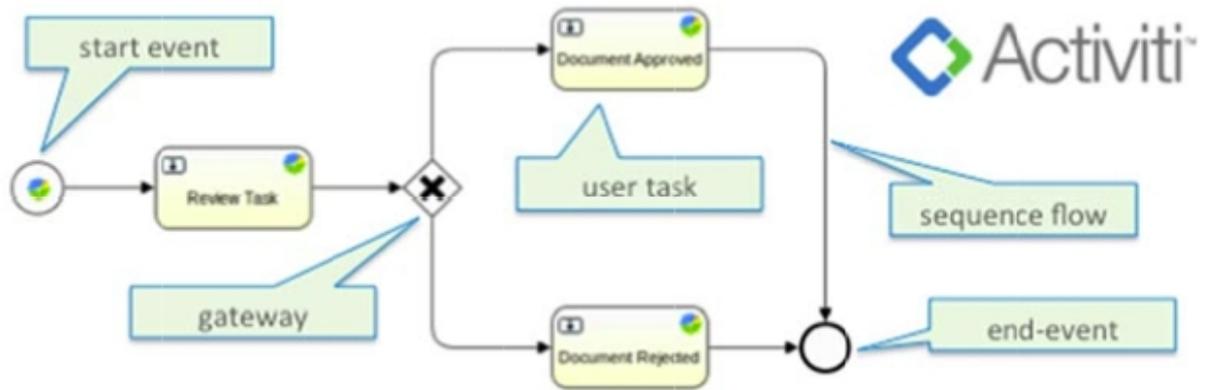
The Activiti workflow console

With the **Activiti Workflow Console** you can:

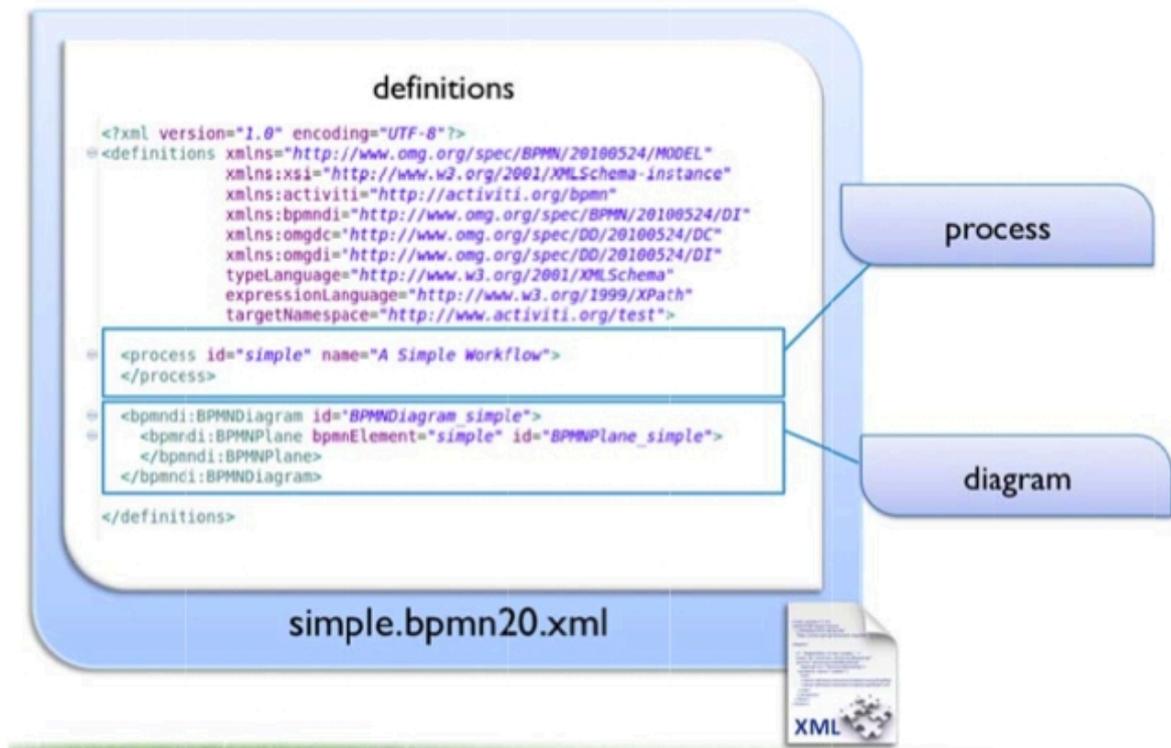
- View process definitions
- Manage deployments; deploy, view versions, and delete versions
- Manage process instances
- View task variables
- Examine the process database

Process definitions

The following diagram shows a simple process definition and highlights the terminology used in BPMN 2.0.



The underlying definition is an xml file. The root element of the BPMN 2.0 schema is the `definitions` element, which can contain multiple process definitions. The following image show an empty process definition:



A `definitions` element contains at least `xmlns` and `targetNamespace` declarations. The `targetNamespace` is an arbitrary string specified by you, and is useful for categorizing process definitions. The `process` element has two attributes:

id

is required and maps to the key property of an Activiti `ProcessDefinition` object. The `id` is used to uniquely identify this process definition, for example when configuring the user interface, or in the Activiti workflow console.

name

is optional and maps to the name property of a ProcessDefinition. The Activiti workflow engine itself does not use this property, but it is used in Alfresco Share for displaying the name in a user interface, so you should specify a name.

The `BPMNDiagram` element specifies the diagram interchange information for this process. The graphical design tool you use generates this information. This element will not appear when you are creating BPMN 2.0 process definition manually. The interchange information is used to re# create the diagram both in another graphical designer and in the run#time environment. Only one diagram is allowed per file, even though there might be more than one process definition.

Events

There are several types of events defined by BPMN 2.0, of which two always exist in a definition:

startEvent

indicates where a process starts. A start event is triggered by the arrival of a message or similar trigger such as the expiration of a timer.

endEvent

models the end process or subprocess. When process execution arrives in an end event, a result is thrown.

Events are described in detail in the Activiti user guide.

Sequence flows

After an element is visited during process execution, all outgoing sequence flows will be followed. So by default two outgoing sequence flows will create two separate, parallel paths of execution. This behavior can be modified. Sequence flows are described in detail in the Activiti user guide.

Tasks**userTask**

describes work to be done by a human actor. When process execution arrives at a user task, a new task is created in the task list of the user or group assigned to that task.

scriptTask

describes an automatic activity. When a process execution arrives at the script task, the corresponding script is executed.

mailTask

is similar to a script task, but is specifically set up to send an email.

Tasks are described in detail in the Activiti user guide.

Gateways

A gateway is capable of consuming or generating tokens. It is graphically visualized as a diamond shape, with an icon inside. The icon describes the type of gateway. Gateways are described in detail in the Activiti user guide.

Parallel gateways**fork**

all outgoing sequence flows are followed in parallel, creating one concurrent execution for each sequence flow.

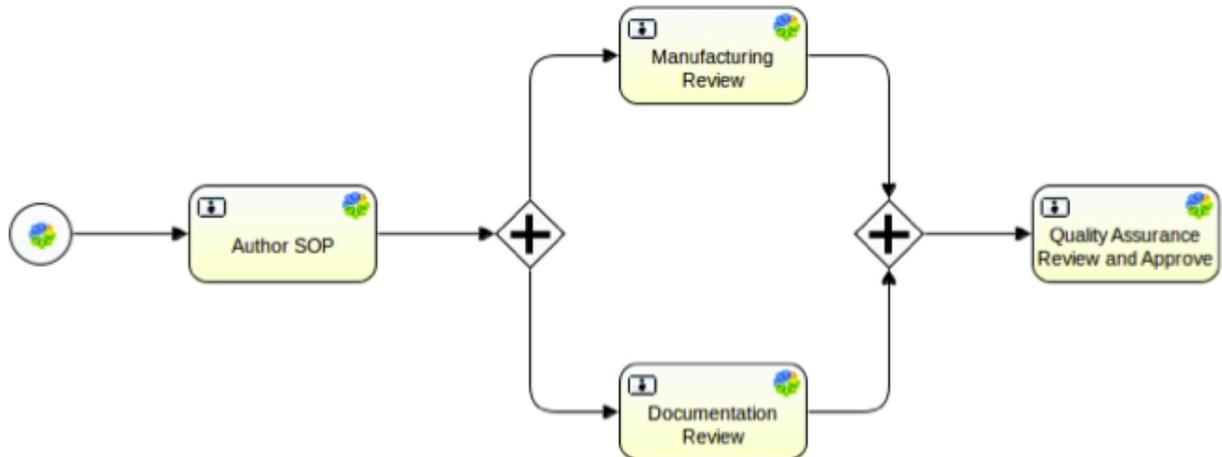
join

all concurrent executions arriving at the parallel gateway wait at the gateway until execution has completed for each of the incoming sequence flows. The process then continues.

A parallel gateway can have both fork and join behavior, if there are multiple incoming and outgoing sequence flows for the same parallel gateway. In this case, the gateway will first join all the incoming sequence flows, before splitting into multiple concurrent paths of execution.

A parallel gateway does not evaluate conditions. If conditions are defined on the sequence flow connected with the parallel gateway, they are ignored.

The following diagram shows a definition with two parallel gateways.



The first gateway forks the flow of execution, generating two tokens for two review tasks. When these two tasks are completed, the second parallel gateway joins the two execution. Since there is only one outgoing sequence flow, no concurrent paths of execution will be created, and only the quality assurance task will be active.

Note that a parallel gateway does not need to be 'balanced'. You do not need to specify a matching number of incoming/outgoing sequence flows for corresponding parallel gateways.

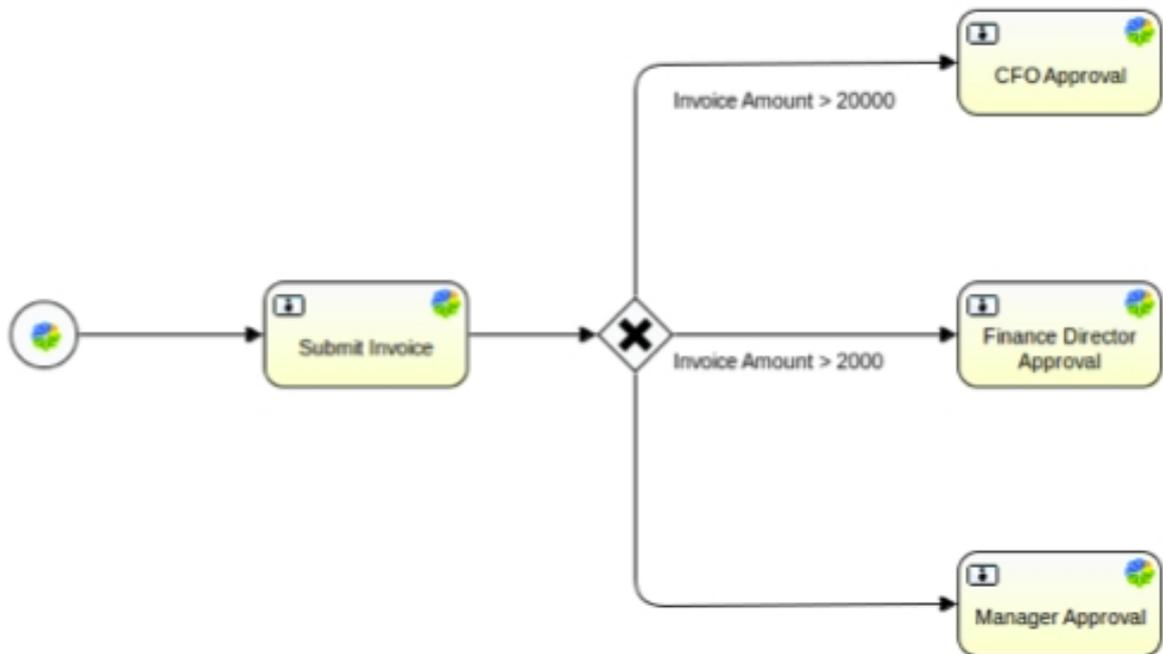
Exclusive gateways

When the execution of a workflow arrives at this gateway, all outgoing sequence flows are evaluated in the order in which they are defined. The sequence flow whose condition evaluates to true, is selected for propagating the token flow.

Note that the semantics of an outgoing sequence flow:

- In general in BPMN 2.0, all sequence flows whose conditions evaluate to true are selected to continue in a parallel way. When using an exclusive gateway, only one sequence flow is selected.
- When multiple sequence flows have conditions which evaluate to true, only the first one defined is selected to continue the process.
- If no sequence flow can be selected, an exception will be thrown. To ensure a sequence flow will always be selected, have no condition on one of your flows. No condition will always evaluate to true.

The following diagram shows an exclusive gateway that will choose one sequence flow based on the value of a property, in this example, the invoice amount. Only two flows have conditions on them going to CFO Approval and Finance Director Approval. The last sequence flow has no condition, and will be selected by default if the other conditional flows evaluate to false.



Variables

For example, the Alfresco supplied BPM task model defines the property **bpm:assignee**. To reference this property in your process definition you would specify the string **bpm_assignee**. Note that the colon character is replaced by an underscore.

Variables in workflows exist at two levels; the process execution level and the task level. If you set the value of a variable in a task, the new value is not available at the process level. If you want to use a variable across tasks, or between a task and conditional flow, you need to copy the variable to the process execution level. Process level variables are available to tasks and sequence flows.

Node objects

The following variables are set by the start task in your process definition, and are accessible after the start task completes:

bpm_workflowDescription

Description for this in-flight workflow.

bpm_workflowDueDate

Due date for the workflow.

bpm_workflowPriority

Priority for the workflow.

bpm_package

A Repository Node with aspect **bpm:workflowPackage** representing the Workflow package containing content being routed through the workflow.

bpm_context

A Repository Node of type **cm:folder** representing the Alfresco folder in which the workflow was started.

There are some special node objects available in the process definition, that are not part of the task model:

initiator

A Repository Node of type **cm:person** representing the person who initiated the workflow.

initiatorhome

A Repository Node of type **cm:space** representing the home folder of the person who initiated the workflow.

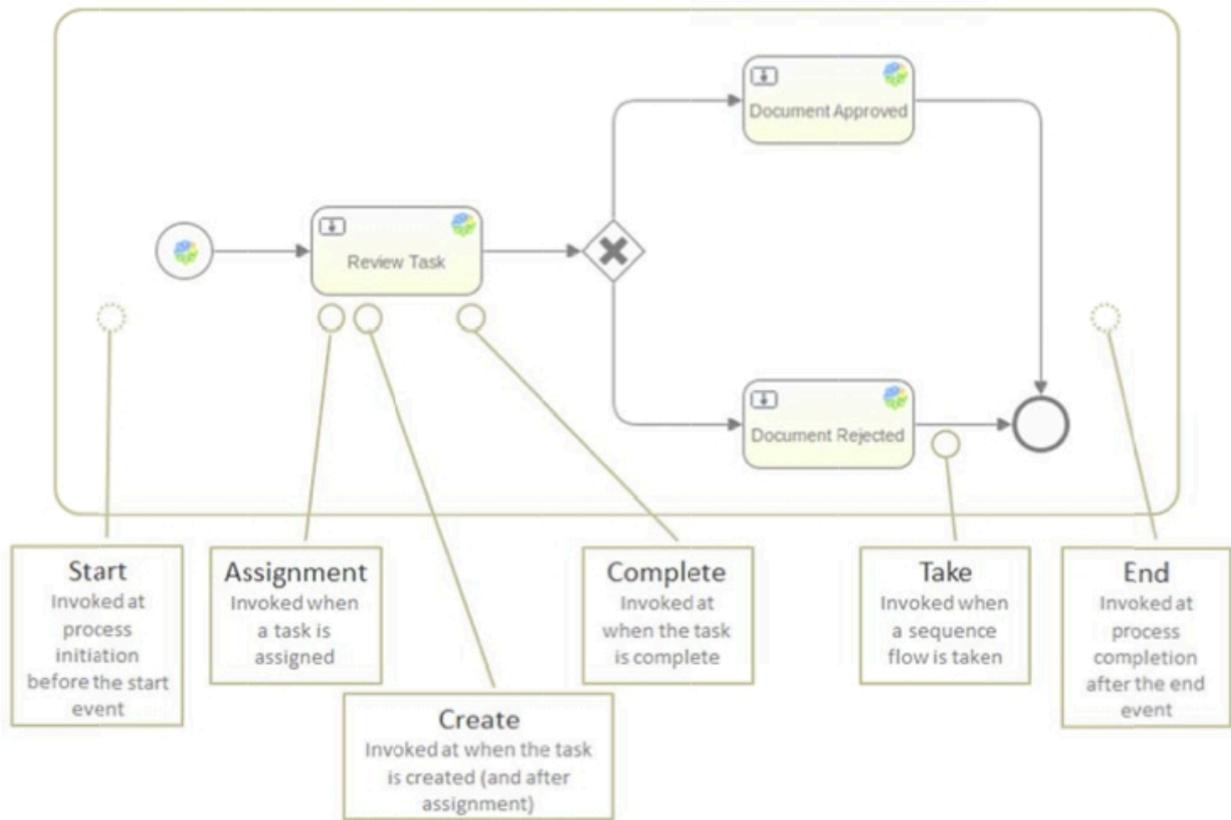
companyhome

A Repository Node of type **cm:space** representing the company home root folder.

Listeners

Execution listeners can be configured on the process itself, as well as activities and transitions. Task listeners can only be configured on user tasks.

Listeners enable you to run your own code in the workflow. This can be Alfresco Javascript or a call to a Java class. The following diagram shows the events in a process definition where you can configure a listener.



Listeners are described in detail in the Activiti user guide.

Task listeners

The following diagram shows an XML fragment from a process definition that contains Alfresco-specific task listener.

```

<userTask id="marketingReview"
  name="Marketing Review"
  activiti:assignee="{bpm_assignee.properties.userName}"
  activiti:formKey="wf:activitiReviewTask">
  <extensionElements>
    <activiti:taskListener event="complete"
      class="org.alfresco.repo.workflow.activiti.tasklistener.ScriptTaskListener">
      <activiti:field name="script">
        <activiti:string>
          reviewOutcome = task.getVariable('wf_reviewOutcome');
          execution.setVariable('wf_reviewOutcome', reviewOutcome);
          logger.log('wf_reviewOutcome: ' + reviewOutcome);</activiti:string>
        </activiti:field>
      </activiti:taskListener>
    </extensionElements>
  </userTask>

```

Listeners are described in detail in the Activiti user guide.

Execution listeners

There are three events available:

start

invoked at the beginning of process execution, before the start event.

end

invoked at the end of the process execution, after the end event.

take

invoked when a sequence flow is invoked.

The code shows an example of an execution listener to be invoked at the beginning of the process execution.

```

<process id="StandardGroupReview" name="Parallel Group Review And Approve Activiti Process">
  <extensionElements>
    <activiti:executionListener
      event="start"
      class="org.alfresco.repo.workflow.activiti.listener.ScriptExecutionListener">
      <activiti:field name="script">
        <activiti:string>
          execution.setVariable('wf_approveCount', 0);
          execution.setVariable('wf_actualPercent', 0);
          execution.setVariable('wf_requiredPercent', wf_requiredApprovePercent);</activiti:string>
        </activiti:field>
      </activiti:executionListener>
    </extensionElements>

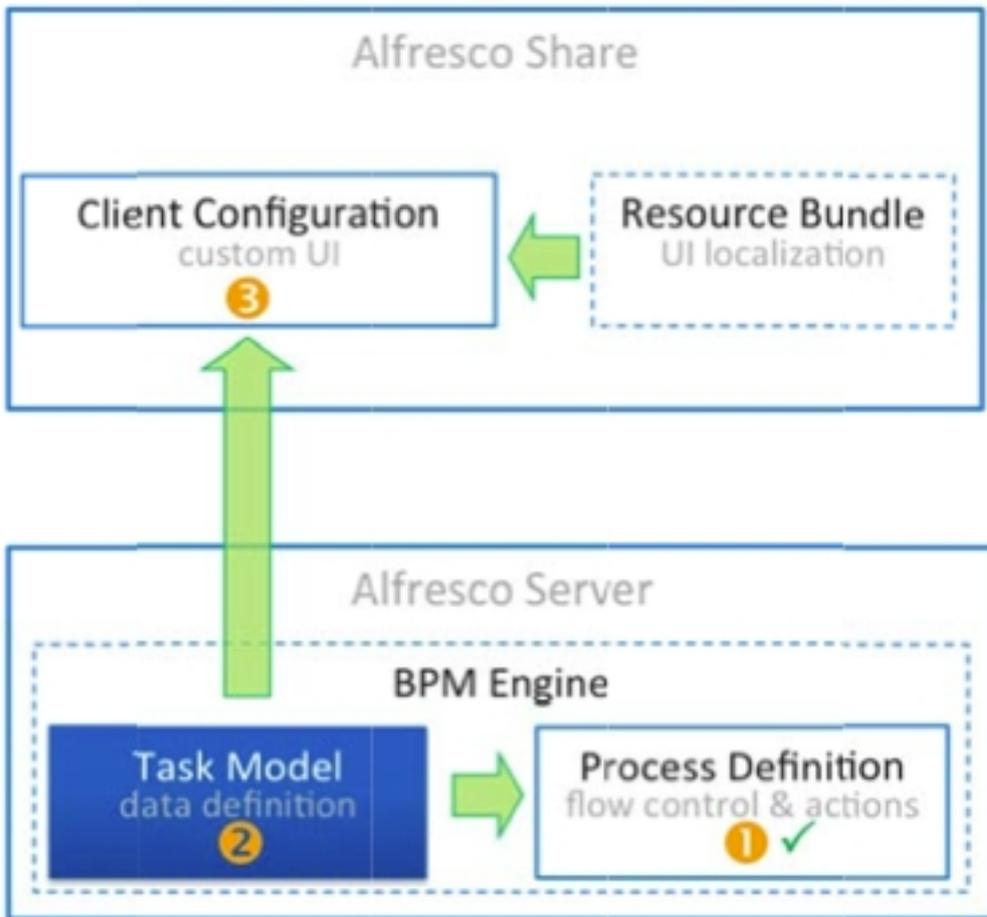
    <!-- Rest of process definition -->
  </process>

```

Listeners are described in detail in the Activiti user guide.

Task model

The client configuration allows for customization of the UI component that is used for presenting workflow-related information to the user and taking inputs from the user. Alfresco uses resource bundles to select the text that displays. Resource bundles allow language-specific strings to be used to display information about a workflow or task. The following diagram shows the relationship between the process definition and the task model on the server, and the client configurations and resource bundle in the client.



When creating workflows you will need to create the process definition using the graphical designer, create a task model to define your specific metadata items required on a task, and optionally look at customizing the user interface to support the custom task model that you have defined. Using a resource bundle is optional.

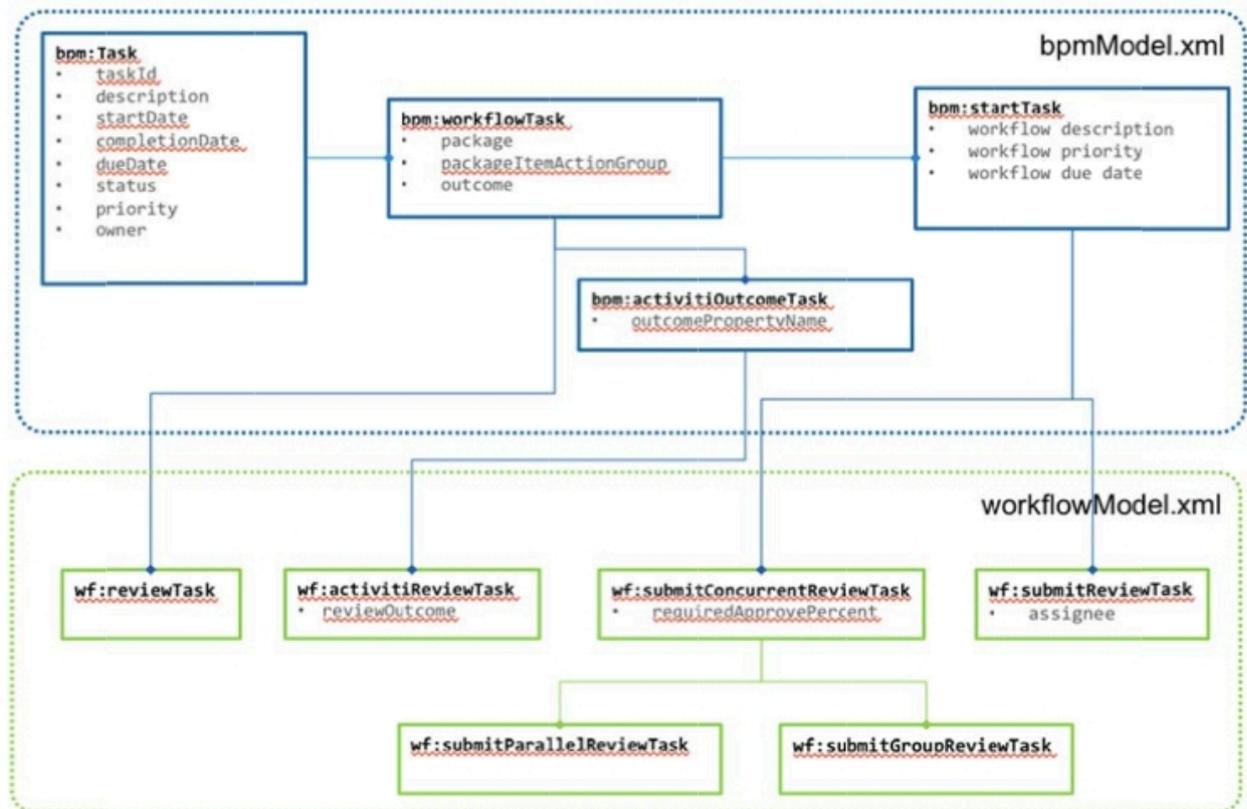
Alfresco ships with two default workflow models that support the default set of process definitions.

bpmModel.xml

is the basic workflow content model

workflowModel.xml

contains more detailed tasks and specializes the basic tasks from the BPM model



The task model is important when considering user interfaces, as the properties from task types are the only properties which can be shown to the user. The following diagram shows how a review task, which is of type `wf:activitiReviewTask` maps to the user interface. The property list in the background is taken from the Activiti workflow explorer.

Specifying the task type

You specify the task type using the **formKey** attribute on a userTask element. If you are developing your BPMN from scratch you can specify this in your XML. If you are using the Activiti designer you can specify it under the main configuration for a task.

```
<userTask id="userTask2" name="Second Task"
  activiti:assignee="{initiator.properties.userName}"
  activiti:formKey="bpm:workflowTask">
</userTask>
```

Setting up Activiti Designer

Installing Eclipse

1. Download the latest version of Eclipse for your platform from <http://www.eclipse.org/downloads>.
2. Follow the installation instructions on linked to on the download page.
3. To run Eclipse, follow the advice in the release notes, `readme_eclipse.html`.

You now have a running eclipse instance in which you can install the Activiti designer plugin.

Installing Activiti designer

Follow these steps to install the plugin.

1. In the eclipse menu bar, click **Help > Install New Software**

2. Click **Add**
The **Add Repository** dialog is displayed
3. Start Eclipse.
4. Fill in the name field with `Activiti BPMN 2.0 designer`, and fill in the location field with `http://activiti.org/designer/update/`.
5. Click **OK**
6. Click **Finish**
Eclipse will install the latest version of the Activiti designer eclipse plugin.

Deploying the task model

In the following example configuration we are deploying a process definition (`adHocModel.bpmn2.0.xml`) and a workflow content model (`adHocModel.bpmn2.0.xml`). In both properties, the “location” is the classpath location of the XML file.

```
<bean id="myworkflows.workflowBootstrap" parent="workflowDeployer">
  <property name="models">
    <list>
      <!-- Task Model associated with above process definition -->
      <value>alfresco/workflow/adhocModel.xml</value>
    </list>
  </property>
  <property name="workflowDefinitions">
    <props>
      <prop key="engineId">activiti</prop>
      <prop key="location">alfresco/extension/adHocModel.bpmn2.0.xml</prop>

      <prop key="mimetype">text/xml</prop>
      <prop key="redeploy">>false</prop>
    </props>
  </property>
</bean>
```

Deploying a process definition

If you use manual deployment, the Alfresco server must be shut down. Process definitions will be deployed when Alfresco starts.

Backing up and restoring

This section describes the process for backing up the Alfresco content repository. It assumes that the various binaries (operating system, database, JDK, application server, and so on) and configuration files (operating system, database, JDK, application server, Alfresco, and so on) are being backed up independently.

Your backup strategy must be tested end-to-end, including restoration of backups that were taken previously. Ensure that you have adequately tested your backup scripts prior to deploying Alfresco to production.

Backing up and restoring the repository

 If Solr is being used, only the following directories must be backed up from the `dir.root` directory:

- `contentstore` directory
- `solr/workspace/` directory

- `solr/archive/ directory`
- `contentstore.deleted directory` (optional)

To restore the backup successfully, the directory and database must be backed up as a single unit. When you restore an Alfresco backup, you must restore both the `dir.root` directory (only the specified directories) and the Alfresco database from the same backup set. Otherwise, the repository will be corrupted.

The `dir.root` directory is defined in the `alfresco-global.properties` file. By default, this directory is named `alf_data` and is located within the directory where Alfresco is installed.

Performing a cold backup

1. Stop Alfresco.
2. Back up the database Alfresco is configured to use, using your database vendor's backup tools.
3. In parallel, back up the `dir.root` directory in its entirety.
4. Store both the database and `dir.root` backups together as a single unit.
For example, store the backups in the same directory or compressed file.
5. Start Alfresco.

Migrating

Migrating servers

The `dir.root` property is usually defined in the `alfresco-global.properties` file.

The `dir.root` is often a directory named `alf_data` within the directory where Alfresco is installed, and will hold both content and full text indexes by default. The `dir.root` location is also reported in the Alfresco logs when the server is started.

Backing up Alfresco Server 1

1. Stop the Alfresco server to ensure that no changes can be made while backing up or restoring.
2. Export the database to `dir.root` (same location as content and indexes).
3. Copy the configuration directory to `dir.root`.

For example:

```
cp -r tomcat/shared/classes/alfresco/extension alf_data
```

4. Back up `dir.root`.

Restoring to Alfresco Server 2

1. Install a compatible Alfresco server. This is typically an identical version to server 1.
 Do not start the new Alfresco server.
2. Restore `dir.root`. If the path is different on server 2, change the `dir.root` configuration.
3. Rename the new server's configuration directory.

For example:

```
mv tomcat/shared/classes/alfresco/extension new_ext
```

4. Move the configuration directory from `dir.root` to the appropriate location
For example:

```
mv alf_data/extension tomcat/shared/classes/alfresco
```

5. If any configuration references server 1 explicitly, change these references to server 2.
6. Import the database from `dir.root`.
7. Start the Alfresco server.

You should now have a new instance of Alfresco on a second server with identical data.

Using the Bulk Import tool

The Bulk Import tool provides a mechanism for bulk importing existing content into a repository from the Alfresco server's file system.

It (optionally) replaces existing content items if they already exist in the repository, but does not perform deletes (it is not designed to fully synchronize the repository with the local file system). The basic on-disk file/folder structure is preserved verbatim in the repository. It is possible to load metadata for the files and spaces being ingested, as well as a version history for files (each version consists of content, metadata, or both).

You can use streaming import to stream the files into the repository content store by copying them in during the import.

There are two types of bulk import:

- Streaming import: This import streams the files into the repository content store by copying them in during the import.
- In-place import: Available in Enterprise Only, these files are assumed to already exist within the repository content store, so no copying is required. This can result in a significant improvement in performance.

There are a number of restrictions:

- Only one bulk import can be running at a time. This is enforced by the `JobLockService`.
- Access to the Bulk Import tool is restricted to Alfresco administrators.

In-place bulk import

The in-place import is available in Enterprise only, and imports files that already exist within the repository content store. As no copying is required, this can result in a significant performance improvement.

Three assumptions are made when importing content "in-place":

- The content is already at its initial repository location prior to import, as it will be **not** be moved during the import.
- The in-place content must be within the tree structure of a registered content store, as defined by either:
 - the default `fileContentStore`
 - a filesystem-based store defined by the content store selector
- Steps have already been taken prior to import to ensure the content structure is well distributed.
 - The default `fileContentStore` distributes content, based on the import date (year/month/day/hour/minute). This avoids having thousands of file under the same root, which is inefficient both for the file system and for computing parent associations in Alfresco (among other things).

- It is recommended you keep immediate children to a few thousands at a maximum.
- In order to choose an efficient distribution scheme, you should know that, when m files are randomly distributed into n leaf folders, when $m \gg n \log n$ the statistical maximum load of a leaf is $m/n + O(\sqrt{(m \log n)/n})$.

In addition, the in-place bulk import provides support for the [Managing the content store](#). This allows you to select under which store the content to import is to be found.

Streaming Bulk Import

The streaming Bulk Import tool copies the source content into the repository content store.

In all other respects, in-place and streaming bulk import are the same.

Preparing the file system

There are a number of tasks you must do to prepare the file system before you do the bulk import.

Metadata files

The Bulk Import tool has the ability to load metadata (types, aspects, and their properties) into the repository. This is done using "shadow" Java property files in XML format as it has good support for Unicode characters. These shadow properties files must have exactly the same name and extension as the file for which it describes the metadata, but with the suffix `.metadata.properties.xml`. For example, if there is a file called `IMG_1967.jpg`, the "shadow" metadata file is called `IMG_1967.jpg.metadata.properties.xml`.

These shadow files can also be used for directories. For example, if you have a directory called "MyDocuments", the shadow metadata file is called `MyDocuments.metadata.properties.xml`.

The metadata file itself follows the usual syntax for Java XML properties files:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <entry key="key1">value1</entry>
  <entry key="key2">value2</entry>
  ...
</properties>
```

There are two special keys:

- `type` contains the qualified name of the content type to use for the file or folder
- `aspects` contains a comma-delimited list of the qualified names of the aspect(s) to attach to the file or folder

The remaining entries in the file are treated as metadata properties, with the key being the qualified name of the property and the value being the value of that property. Multi-valued properties are comma-delimited. However, these values are not trimmed so it's recommended you do not place a space character either before or after the comma, unless you want that in the value of the property.

Here is an example using `IMG_1967.jpg.metadata.properties.xml`:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <entry key="type">cm:content</entry>
  <entry key="aspects">cm:versionable,cm:dublincore</entry>
```

```

<entry key="cm:title">A photo of a flower.</entry>
<entry key="cm:description">A photo I took of a flower while walking around
Bantry Bay.</entry>
<entry key="cm:created">1901-01-01T12:34:56.789+10:00</entry>
<!-- cm:dublincore properties -->
<entry key="cm:author">Peter Monks</entry>
<entry key="cm:publisher">Peter Monks</entry>
<entry key="cm:contributor">Peter Monks</entry>
<entry key="cm:type">Photograph</entry>
<entry key="cm:identifier">IMG_1967.jpg</entry>
<entry key="cm:dcsource">Canon Powershot G2</entry>
<entry key="cm:coverage">Worldwide</entry>
<entry key="cm:rights">Copyright (c) Peter Monks 2002, All Rights
Reserved</entry>
<entry key="cm:subject">A photo of a flower.</entry>
</properties>

```

Additional notes on metadata loading:

- You cannot create a new node based on metadata only, you must have a content file (even if zero bytes) for the metadata to be loaded. Even so, you can "replace" an existing node in the repository with nothing but metadata. Despite the confusing name, this won't replace the content; instead the new metadata is added.
- The metadata must conform to the type and aspect definitions configured in Alfresco (including mandatory fields, constraints, and data types). Any violations will terminate the bulk import process.
- Associations between content items loaded by the tool are not yet nicely supported. Associations to objects that are already in the repository can be created using the NodeRef of the target object as the value of the property.
- Non-string data types (including numeric and date types) have not been exhaustively tested. Date values have been tested and do work when specified using ISO8601 format.
- Updating the aspects or metadata on existing content will not remove any existing aspects not listed in the new metadata file; this tool is not intended to provide a full file system synchronization mechanism.
- The metadata loading facility can be used to supplement content that's already in the Alfresco repository, without having to upload that content again. To use this, create a "naked" metadata file in the same path as the target content file. The tool will match it up with the file in the repository and add the new aspect(s) and/or metadata to that file.

Version History files

The import tool also supports loading a version history for each file. To do this, create a file with the same name as the main file, but append it with a "v#" extension. For example:

```

IMG_1967.jpg.v1    <- version 1 content
IMG_1967.jpg.v2    <- version 2 content
IMG_1967.jpg       <- "head" (latest) revision of the content

```

This also applies to metadata files if you want to capture metadata history as well. For example:

```

IMG_1967.jpg.metadata.properties.xml.v1    <- version 1 metadata
IMG_1967.jpg.metadata.properties.xml.v2    <- version 2 metadata
IMG_1967.jpg.metadata.properties.xml       <- "head" (latest) revision of the
metadata

```

Additional notes on version history loading:

- You cannot create a new node based on a version history only. You must have a head revision of the file.
- Version numbers do not have to be contiguous. You can number your version files however you want, provided you use whole numbers (integers).

- The version numbers in your version files will not be used in Alfresco. The version numbers in Alfresco will be contiguous, starting at 1.0 and increasing by 1.0 for every version (so 1.0, 2.0, 3.0, and so on). Alfresco doesn't allow version labels to be set to arbitrary values, and the bulk import doesn't provide any way to specify whether a given version should have a major or minor increment.
- Each version can contain a content update, a metadata update or both. You are not limited to updating everything for every version. If not included in a version, the prior version's content or metadata will remain in place for the next version.

The following example shows all possible combinations of content, metadata, and version files:

```

IMG_1967.jpg.v1          <- version 1 content
IMG_1967.jpg.metadata.properties.xml.v1  <- version 1 metadata
IMG_1967.jpg.v2        <- version 2 content
IMG_1967.jpg.metadata.properties.xml.v2  <- version 2 metadata
IMG_1967.jpg.v3        <- version 3 content (content only
version)
IMG_1967.jpg.metadata.properties.xml.v4  <- version 4 metadata (metadata
only version)
IMG_1967.jpg.metadata.properties.xml     <- "head" (latest) revision of the
metadata
IMG_1967.jpg              <- "head" (latest) revision of the
content

```

Importing with the Bulk Import tool

You can bulk import by using the user interface, or with a program.

Alfresco web scripts are used for bulk importing. If you choose to code the bulk import, code examples are provided to help you. In both cases, you can use the reference table to determine the fields and data that are required for a successful import.

If you need to troubleshoot or diagnose any issues with a bulk import, you can enable logging. To enable debugging for the Bulk Import tool, add the following command to the `log4j.properties` file before deployment:

```
log4j.logger.org.alfresco.repo.bulkimport=DEBUG
```

Set the debug statements to at least INFO level:

```
log4j.logger.org.alfresco.repo.batch.BatchProcessor=info
```

You can also enable logging for the transaction handler to identify any transactional issues during the import:

```
log4j.logger.org.alfresco.repo.transaction.RetryingTransactionHelper=info
```

For more information about log4j, see [log4j.properties file](#).

Bulk importing using the user interface

Streaming

Streaming bulk import is exposed in two web scripts:

1. A simple UI web script that can be used to manually initiate an import. This is an HTTP GET web script with a path of: `http://localhost:8080/alfresco/service/bulkfsimport`
2. An initiate web script that kicks off an import using parameters that are passed to it (for the source directory, target space, and so on). If you want to script or invoke the tool programmatically, this is the web script that you call. This is an HTTP GET web script with a path of: `http://localhost:8080/alfresco/service/bulkfsimport/initiate`

The UI web script presents the following simplified HTML form:

Bulk Filesystem Import Tool

Alfresco Enterprise v4.2.2 (r-70123-b46)

Import directory:

Target space :
Path:

or NodeRef:

Disable rules: (unchecked means rules are enabled during the import)

Replace existing files: (unchecked means skip files that already exist in the repository)

Batch Size:

Number of Threads:

Initiate Bulk Import

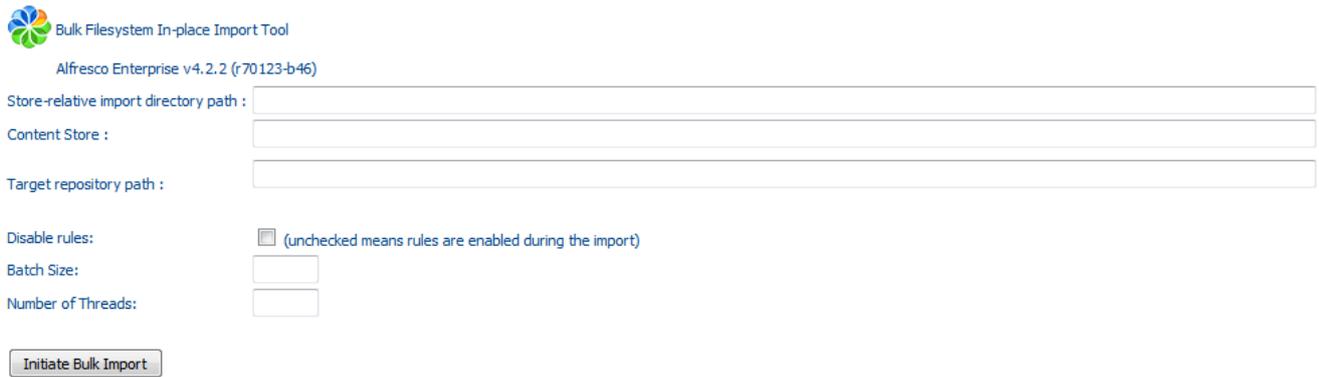
- The **Import directory** field is required and indicates the absolute file system directory to load the content and spaces from, in an OS-specific format. Note that this directory must be locally accessible to the server on which the Alfresco instance is running. It must either be a local file system or a locally mounted remote file system (mounted using NFS, GFS, CIFS, or similar).
- The **Target space (Path)** field is also required and indicates the target space to load the content into, as a path starting with `/Company Home`. The separator character is Unix-style `/`, regardless of the platform Alfresco is running on. This field includes an AJAX auto-suggest feature, so you can type any part of the target space name, and an AJAX search is performed to find and display matching items.
- The **Target space (NodeRef)** field is an alternative to **Target space (Path)** and indicates the target NodeRef to load the content into.
- The **Disable rules** check box allows you to turn off rule processing during the bulk import.
- The **Replace existing files** option indicates whether to replace nodes that already exist in the repository (checked) or skip them (unchecked). Note that if versioning is enabled for a node, the node's existing content and metadata is preserved as the prior version and the new content and/or metadata will be written into the head revision.
- The **Batch Size** text field allows you to override the default batch size (the number of directories and files to import at a time, per transaction; defined by the property `bulkImport.batch.batchSize`) to use in the bulk import.
- The **Number of Threads** text field allows you to override the default number of threads (defined by the property `bulkImport.batch.numThreads`) to use in the bulk import.

In-place

In-place bulk import is exposed in a series of two web scripts:

1. A simple UI web script that can be used to manually initiate an import. This is an HTTP GET web script with a path of: `http://localhost:8080/alfresco/service/bulkfsimport/inplace`
2. An initiate web script that kicks off an import, using parameters that are passed to it (for the source directory, target space, and so on). If you want to script or programmatically invoke the tool, this is the web script that you call. This is an HTTP GET web script with a path of: `http://localhost:8080/alfresco/service/bulkfsimport/inplace/initiate`

The in-place UI web script presents the following simplified HTML form:



Bulk Filesystem In-place Import Tool

Alfresco Enterprise v4.2.2 (r70123-b46)

Store-relative import directory path :

Content Store :

Target repository path :

Disable rules: (unchecked means rules are enabled during the import)

Batch Size:

Number of Threads:

- The **Store-relative import directory path** field is required and indicates the file system path for loading content and spaces, relative to the content store, in an OS-specific format. Note that this directory must be locally accessible to the server the Alfresco instance is running on - it must either be a local file system or a locally mounted remote file system (mounted using NFS, GFS, CIFS or similar). This directory must already be inside an existing content store.
- The **Content Store** field is the name of the store that holds the content, as defined within the storage configuration (content store selector or direct fileContentStore). The default store is by default named "default". An autocomplete menu will assist in selecting the name as the first characters are entered. The **Up** and **Down** keyboard keys can be used to navigate the list, in addition to the mouse.
- The **Target repository path** field is also required and indicates the target space to load the content into, as a path starting with `/Company Home`. The separator character is Unix-style `/`, regardless of the platform Alfresco is running on. This field includes an AJAX auto-suggest feature, so you can type any part of the target space name, and an AJAX search is performed to find and display matching items.
- The **Disable rules** option allows you to turn off rule processing during the bulk import.
- The **Batch Size** text field allows you to override the default batch size (the number of directories and files to import at a time, per transaction; defined by the property `bulkImport.batch.batchSize`) to use in the bulk import.
- The **Number of Threads** text field allows you to override the default number of threads (defined by the property `bulkImport.batch.numThreads`) to use in the bulk import.

The status web page is the same for both streaming and in-place import.

Bulk Filesystem Import Tool Status

The bulk import status web script returns status information on the current import (if one is in progress), or the status of the last import that was initiated. This web script has both HTML and XML views, allowing external programs to programmatically monitor the status of imports. This is an HTTP GET web script with a path of: `http://localhost:8080/alfresco/service/bulkfsimport/status`

The status web page is the same for both streaming and in-place import. The status is updated every five seconds when a bulk import has been initiated. When the bulk import has completed, it displays as follows:

General Statistics				
Current status:	Idle			
Successful:	No			
Batch Size:	20			
Number of threads:	4			
Source Directory:	/data/upgrade/alf_data/contentstore/temp1			
Target Space:	/Company Home/bulk2			
Start Date:	2014-05-15 02:17:09.706PM			
End Date:	2014-05-15 02:17:09.709PM			
Duration:	0d 0h 0m 0s 2.443ms			
Number of Completed Batches:	0			
Source (read) Statistics				
Scanned:	Folders	Files	Unreadable	
	1	0	0	
Read:	Content	Metadata	Content Versions	Metadata Versions
	0 (0B)	0 (0B)	0 (0B)	0 (0B)
Throughput:				
Target (write) Statistics				
Space Nodes:	# Created	# Replaced	# Skipped	# Properties
	0	0	0	0
Content Nodes:	# Created	# Replaced	# Skipped	Data Written
	0	0	0	0B
Content Versions:	# Created	Data Written	# Properties	
	0	0B	0	
Throughput (write):				
Error Information From Last Run				
File that failed:	n/a			
Exception:	<pre> java.lang.IllegalArgumentException: Source '/data/upgrade/alf_data/contentstore/temp1' doesn't exist. at org.alfresco.repo.bulkimport.impl.AbstractBulkFilesystemImporter.validateSourceIsReadableDirectory(AbstractBulkFilesystemImporter.java:387) at org.alfresco.repo.bulkimport.impl.AbstractBulkFilesystemImporter\$3.execute(AbstractBulkFilesystemImporter.java:371) at org.alfresco.repo.bulkimport.impl.AbstractBulkFilesystemImporter\$3.execute(AbstractBulkFilesystemImporter.java:371) at org.alfresco.repo.transaction.RetryingTransactionHelper.doInTransaction(RetryingTransactionHelper.java:450) at org.alfresco.repo.bulkimport.impl.AbstractBulkFilesystemImporter.bulkImport(AbstractBulkFilesystemImporter.java:370) at org.alfresco.repo.bulkimport.impl.AbstractBulkFilesystemImporter\$2\$1.doWork(AbstractBulkFilesystemImporter.java:351) at org.alfresco.repo.security.authentication.AuthenticationUtil.runAs(AuthenticationUtil.java:548) at org.alfresco.repo.bulkimport.impl.AbstractBulkFilesystemImporter\$2.run(AbstractBulkFilesystemImporter.java:347) at java.lang.Thread.run(Thread.java:724) </pre>			

Initiate another import

Initiate another in-place import

For more information about the fields and their meanings, see [Bulk Import tool fields and values](#) on page 318.

Bulk importing using a program

Code examples show you how to complete a streaming bulk import and an in-place bulk import programmatically.

Streaming

```

UserTransaction txn = transactionService.getUserTransaction();
txn.begin();

AuthenticationUtil.setRunAsUser("admin");

StreamingNodeImporterFactory streamingNodeImporterFactory =
(StreamingNodeImporterFactory)ctx.getBean("streamingNodeImporterFactory");
NodeImporter nodeImporter = streamingNodeImporterFactory.getNodeImporter(new
File("importdirectory"));
BulkImportParameters bulkImportParameters = new BulkImportParameters();
bulkImportParameters.setTarget(folderNode);
bulkImportParameters.setReplaceExisting(true);
bulkImportParameters.setBatchSize(40);
bulkImportParameters.setNumThreads(4);
bulkImporter.bulkImport(bulkImportParameters, nodeImporter);

txn.commit();
  
```

In-place

```

txn = transactionService.getUserTransaction();
txn.begin();
  
```

```

AuthenticationUtil.setRunAsUser("admin");

InPlaceNodeImporterFactory inPlaceNodeImporterFactory =
(InPlaceNodeImporterFactory)ctx.getBean("inPlaceNodeImporterFactory");
NodeImporter nodeImporter =
inPlaceNodeImporterFactory.getNodeImporter("default", "2011");
BulkImportParameters bulkImportParameters = new BulkImportParameters();
bulkImportParameters.setTarget(folderNode);
bulkImportParameters.setReplaceExisting(true);
bulkImportParameters.setBatchSize(150);
bulkImportParameters.setNumThreads(4);
bulkImporter.bulkImport(bulkImportParameters, nodeImporter);

txn.commit();

```

For more information about the Alfresco web scripts that you invoke to script a bulk import, see [Bulk importing using the user interface](#) on page 314.

For more information about the bulk import fields and their meanings, see [Bulk Import tool fields and values](#) on page 318.

Bulk Import tool fields and values

The Bulk Import tool has a number of entry and display fields that are displayed in the user interface, but also referenced in the status.xml file that is used if you are programming a bulk import. The labels, fields, possible values and a summary of each entry is explained in this information.

Field label (from Bulk Import status web page)	Field entry (from status.xml file)	Possible values	Summary
Current status	<CurrentStatus>Idle</CurrentStatus>	Idle In Progress	Status of the bulk import
Successful	<ResultOfLastExecution>Yes</ResultOfLastExecution>	Yes No n/a	Result of the bulk import
Batch Size	<batchSize>20</batchSize>	Numeric	The batch size (number of directories and files to import at a time) specified for the bulk import
Number of threads	<numThreads>4</numThreads>	Numeric	The number of threads specified for the bulk import
Source Directory	<SourceDirectory>importdir</SourceDirectory>	Alphanumeric	The absolute path of the filesystem directory being imported
Target Space	<TargetSpace>/Company Home</TargetSpace>	Alphanumeric	The path of the Alfresco space where the content is being loaded, starting with /Company Home
Start Date	<StartDate>2014-05-15 01:30:11.912PM</StartDate>	Date and timestamp	Start of the bulk import. Format is YYYY-MM-DD HH:MM:SS.sss AM PM
End Date	<EndDate>2014-05-15 01:30:12.009PM</EndDate>	Date and timestamp	End of the bulk import. Format is YYYY-MM-DD HH:MM:SS.sss AM PM
Duration	<DurationInNS>0d 0h 0m 0s 96.941ms</DurationInNS>	Alphanumeric	Time taken for the bulk import to complete. Format is xd xh xm xxs xx.xxxms where x is a number

Field label (from Bulk Import status web page)	Field entry (from status.xml file)	Possible values	Summary
Number of Completed Batches	<CompletedBatches>0</CompletedBatches>	Numeric	Number of batches completed in the bulk import
Source (read) Statistics	<SourceStatistics>		
Scanned: Folders	<FoldersScanned>0</FoldersScanned>	Numeric	Number of source folders scanned
Scanned: Files	<FilesScanned>0</FilesScanned>	Numeric	Number of source files scanned
Scanned: Unreadable	<UnreadableEntries>0</UnreadableEntries>	Numeric	Number of unreadable source files
Read: Content	<ContentFilesRead>0</ContentFilesRead>	Numeric	Amount of source content read. Format is numeric with size of content in parentheses
Read: Metadata	<MetadataFilesRead>0</MetadataFilesRead>	Numeric	Amount of source metadata read. Format is numeric with size of metadata in parentheses
Read: Content Versions	<ContentVersionFilesRead>0</ContentVersionFilesRead>	Numeric	Source content versions read. Format is numeric with size of content versions in parentheses
Read: Metadata Versions	<MetadataVersionFilesRead>0</MetadataVersionFilesRead>	Numeric	Source metadata versions read. Format is numeric with size of metadata versions in parentheses
Throughput	N/A	Numeric	Number of entries scanned per second, number of files read per second, and size of data read per second
Target (write) Statistics	<TargetStatistics>		
Space Nodes: # Created	<SpaceNodesCreated>0</SpaceNodesCreated>	Numeric	Number of target space nodes created
Space Nodes: # Replaced	<SpaceNodesReplaced>0</SpaceNodesReplaced>	Numeric	Number of target space nodes replaced
Space Nodes: # Skipped	<SpaceNodesSkipped>0</SpaceNodesSkipped>	Numeric	Number of target space nodes skipped
Space Nodes: # Properties	<SpacePropertiesWritten>0</SpacePropertiesWritten>	Numeric	Number of properties written for target space nodes
Content Nodes: # Created	<ContentNodesCreated>0</ContentNodesCreated>	Numeric	Number of target content nodes created
Content Nodes: # Replaced	<ContentNodesReplaced>0</ContentNodesReplaced>	Numeric	Number of target content nodes replaced
Content Nodes: # Skipped	<ContentNodesSkipped>0</ContentNodesSkipped>	Numeric	Number of target content nodes skipped
Content Nodes: # Data Written	<ContentBytesWritten>0</ContentBytesWritten>	Numeric	Amount of target content node data written
Content Nodes: # Properties	<ContentPropertiesWritten>0</ContentPropertiesWritten>	Numeric	Number of properties written for target content nodes
Content Versions: # Created	<ContentVersionsCreated>0</ContentVersionsCreated>	Numeric	Number of target content versions created

Field label (from Bulk Import status web page)	Field entry (from status.xml file)	Possible values	Summary
Content Versions: # Data Written	<ContentVersionsBytesWritten>ContentVersionsBytesWritten</ContentVersionsBytesWritten>	Numeric	Amount of target content version data written
Content Versions: # Properties	<ContentVersionsPropertiesWritten>0</ContentVersionsPropertiesWritten>	Numeric	Number of properties written for target content versions
Throughput (write)	N/A	Numeric	Number of nodes scanned per second and size of data written per second
Error Information From Last Run	<ErrorInformation>		
File that failed	<FileThatFailed>n/a</FileThatFailed>	Alphanumeric	The name of the file that failed during the bulk import
Exception	<Exception>exceptionLog</Exception>	Alphanumeric	The stack trace of the exception that occurred during the bulk import

Using content stores

A content store provides low-level access to stored binaries ensuring that, for every write, a new binary storage location is made available. This section gives an overview on the content stores, their types, and configuration details with examples.

Content stores overview

This section gives an overview of content store and content binary life cycle.

A content store (`ContentStore`) or combinations of content stores can be used to control how and where the binary files are physically stored. Binary streams can be stored across a range of locations and can be encrypted/decrypted, as necessary. Also, fast versus slow storage options can be wired up together for efficient storage and access.

Alfresco supports seven different content stores. These are the File content store (default Alfresco content store), Content store selector, S3 content store, Caching content store, Aggregating content store, Encrypted content store, and Centera content store. For more information on each content store, see [Content store types](#).

Common behaviour of different content stores

- Content stores always write to a new location, so binary files are never overwritten. The content is never modified.
- Each content store can support its own URL standard.

Content binaries life cycle

Stage 1 - Content writes: When you create a file in Alfresco, it becomes a content (in form of a `.bin` file) and is stored in the default file content store, for example `<ALFRESCO_HOME>\alf_data\contentstore` directory. The metadata of the content is stored in the database. The database contains a reference to that `.bin` file.

Stage 2 - Content reads: When a request is made to the `ContentStore` for a `ContentReader`, the client reads the content using methods on the `ContentReader`.

Stage 3 - Copying, moving and versioning files: The content binaries are **never modified** by any high-level process. Moving, copying and versioning a file merely affects the content metadata. It is possible to end up with several references to the same raw binary content. Also,

writes to the file system do not become visible until the metadata has been committed to the database.

Stage 4 - Cleaning up binary files: When a content URL is no longer attached to any metadata in the system, it is referred to as orphaned. In order to allow adequate time for backup, the content binaries are not deleted immediately. Instead, they are deleted on a schedule. The job runs against the following CRON expression:

```
system.content.orphanCleanup.cronExpression=0 0 4 * * ?
```

As an additional safety measure, the binaries are first copied to a local backup at:

```
dir.contentstore.deleted=${dir.root}/contentstore.deleted
```

This location can be cleared out by administrators, as necessary. The time to protect orphaned binaries is controlled by:

```
system.content.orphanProtectDays=14
```

In most cases, there is no need to change this and the value should be large enough to encompass a sufficient number of full content backups.

Content store types

By default, Alfresco is configured to save files or content items in the File content store and orphaned files in the Deleted content store. Alfresco also provides other content stores, which may be used in place of or in addition to the default stores. This section gives an overview on the File content store and additional content stores that you can use with Alfresco.

File content store

This topic describes the File content store, which is Alfresco's default content store.

The **File content store** saves the files or content items on a file system under the root directory. Within the root directory, the files are stored in numeric directories based upon the creation time of the document. The reason for storing the files in a directory structure is to assist incremental backup. The metadata of your file is stored in the database.

Alfresco does not modify any file that is stored in the content store. The `fileContentStore` is pointed to by the `dir.contentstore` property.

Alfresco S3 content store

The Alfresco S3 Connector is an add-on module that provides an alternative content store. It uses Amazon's Simple Storage Service (S3) as the storage mechanism for Alfresco, allowing for virtually unlimited and inexpensive storage.

For more information on Alfresco S3 connector, see [Installing and configuring Alfresco S3 connector](#).

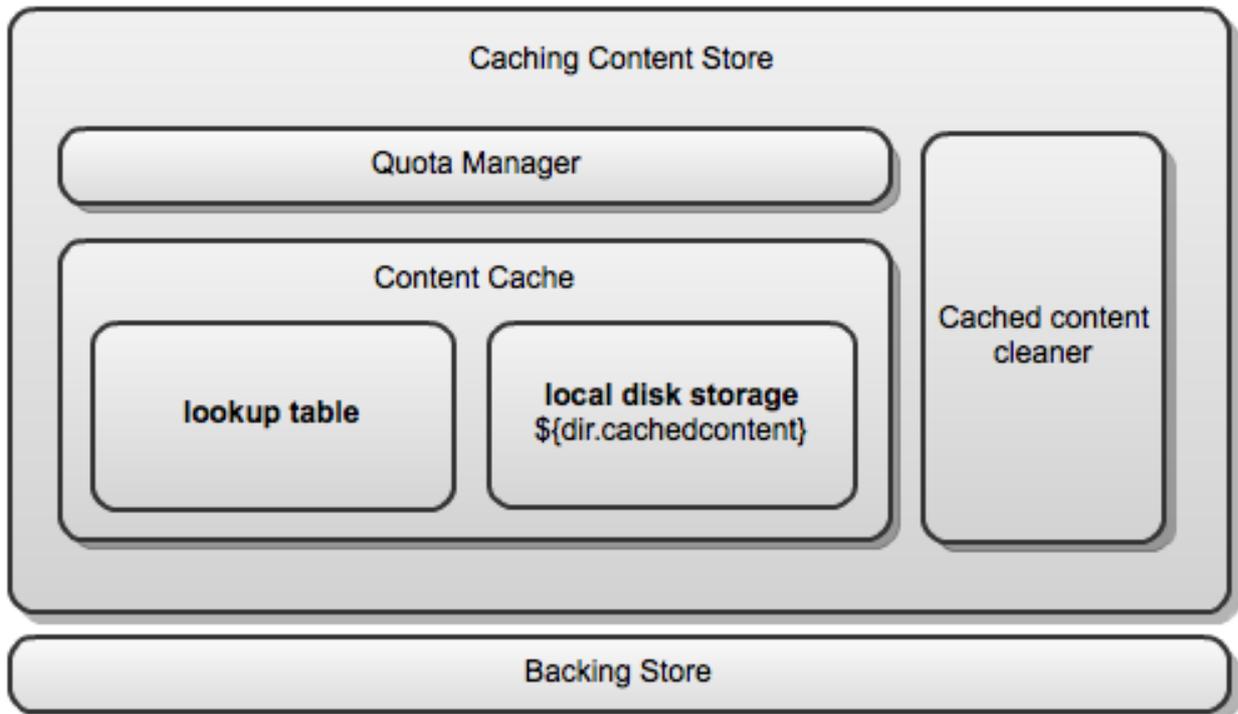
Caching content store (CCS)

This section provides an overview on Caching content store (CCS) and describes how to configure it.

CachingContentStore class overview

The `CachingContentStore` class adds transparent caching to any `ContentStore` implementation. Wrapping a slow `ContentStore` in a `CachingContentStore` improves access speed in many use cases. Example use cases include document storage using a XAM appliance or cloud-based storage, such as Amazon's S3.

The diagram shows the architecture of CCS.



The major classes and interfaces that form the Caching Content Store are:

- **CachingContentStore:** This is the main class that implements the `ContentStore` interface, and can therefore, be used anywhere that a `ContentStore` could be used. The `CachingContentStore` handles all the high level logic of interaction between the cache and the backing store, while the caching itself is provided by a collaborating `ContentCache` object.
- **ContentCache:** This class is responsible for putting items into and getting items from the cache. The single supplied implementation (`ContentCacheImpl`) for this class uses a lookup table to keep track of the files that are being managed by the cache, and a directory on the local file system to store the cached content files. The lookup table itself is a `SimpleCache` implementation instance (for example, `DefaultSimpleCache` or `HazelcastSimpleCache` when running a clustered environment).
- **QuotaManagerStrategy:** The quota managers implement this interface and control how the disk usage is consumed for cached content storage. Alfresco provides two implementations for this: `UnlimitedQuotaStrategy` (does not restrict disk usage, thereby effectively disabling the quota function) and `StandardQuotaStrategy` (attempts to keep usage below the maximum specified in bytes or MB).

The `CachingContentStore` class is highly configurable and many of its components could be exchanged for other implementations. For example, the lookup table could easily be replaced with a different implementation of `SimpleCache` than that supplied.

The cached content cleaner (`CachedContentCleaner`) periodically traverses the directory structure containing the cached content files and deletes the content files that are not in use by the cache. Files are considered not in use by the cache if they have no entry in the lookup table managed by `ContentCacheImpl`. The content cache cleaner is not a part of the architecture but is a helper object for `ContentCacheImpl` and allows it to operate more efficiently.

`CachingContentStore` properties

This topic describes the properties that you can configure for the `CachingContentStore` class.

The following properties are used in the sample context file, `caching-content-store-context.xml.sample` and can be set in the `alfresco-global.properties` file. Their default values are provided in the `repository.properties` file.

system.content.caching.cacheOnInbound=true

Enables write-through caching. If true, an attempt to write the content to the backing store results in the item being cached. Therefore, the first time an item is read (provided the item has not been removed from the cache in the mean time), the file is already cached locally for faster access times. It is recommended that this property is set to `true` for most usage scenarios.

system.content.caching.maxDeleteWatchCount=1

Defines the number of times the file must have been observed as being available for deletion by previous cleanup runs before it is actually deleted. The default value is always set to 1, but can be increased if readers obtained from the cache could not be used due to the underlying file being deleted.

system.content.caching.contentCleanup.cronExpression=0 0 3 * * ?

Specifies how often the cached content cleanup job will run. The supplied value is a quartz expression and is similar to a Unix cron expression. In this case, the cleaner will run at 3 am every morning.

system.content.caching.timeToLiveSeconds=0

Specifies the maximum time in seconds that an item can exist in the cache. After this time elapses, the item will no longer be cached and a request for the content URL will result in the item being fetched from the backing store and cached afresh. A value of 0 means that items will not have a TTL parameter applied to them.

system.content.caching.timeToIdleSeconds=60

Specifies the maximum time an item in the cache can exist without being requested. Each time the item is accessed, the Time To Idle parameter is refreshed and the item will remain in the cache.

system.content.caching.maxElementsInMemory=5000

Applies to the lookup table in the ContentCache. Each content URL requires two entries in the lookup table, so a value of 5000 can allow 2500 content items to be held in memory for the lookup table.

system.content.caching.maxElementsOnDisk=10000

Applies to the lookup table in the ContentCache. Each content URL requires two entries in the lookup table, so a value of 10000 can allow 5000 items to be held on disk.

system.content.caching.minFileAgeInMillis=2000

Specifies that files must be at least this age before they are marked for deletion. This also stops unnecessary checks, such as loading and examining the associated properties file.

system.content.caching.maxUsageMB=4096

Specifies the maximum disk usage in MB that cached content should consume. In other words, this property defines the disk space quota allocated to the `dir.cachedcontent` directory. It is used by the `StandardQuotaStrategy` class as configured in the `caching-content-store-context.xml.sample` file.

system.content.caching.maxFileSizeMB=0

Specifies the maximum size in MB of any individual file of cached content. Content larger than this size can still be retrieved using the `CachingContentStore` class but the content will not be cached. If this property is set to zero, then no size limit will apply to the individual files. This property is used by the `StandardQuotaStrategy` class as configured in the `caching-content-store-context.xml.sample` file.

Configuring `CachingContentStore`

You can configure the `CachingContentStore` class.

To demonstrate step-by-step configuration of the `CachingContentStore` class, the `spring context file, caching-content-store-context.xml.sample is used as a starting point for adding caching to a content store. Once configured, you can activate the sample file by removing the .sample file extension and placing it in your Alfresco installation extension directory at <ALFRESCO_HOME>/tomcat/shared/classes/alfresco/extension.`

1. Define an instance of the `CachingContentStore` class. This is the top level bean that ties together the CCS as a whole.

```
<bean id="fileContentStore"
  class="org.alfresco.repo.content.caching.CachingContentStore" init-
  method="init">
  <property name="backingStore" ref="backingStore"/>
  <property name="cache" ref="contentCache"/>
  <property name="cacheOnInbound"
  value="\${system.content.caching.cacheOnInbound}"/>
  <property name="quota" ref="standardQuotaManager"/>
</bean>
```

In this case, the `fileContentStore` bean is overridden. The `ContentService` bean uses `fileContentStore` bean, so CCS is used automatically. You can also specify a different name and an overridden `contentService` bean. The main collaborators of `backingStore`, `cache` and `quota` refer to the beans for Backing Store, Content Cache and Quota Manager as shown in the diagram in the [CachingContentStore overview](#) topic. Each `CachingContentStore` class should have its own dedicated instances of these collaborators and they should not be shared across other `CachingContentStore` beans, should you have any defined.

2. Define a backing store. This CCS uses this `ContentStore` to provide caching for `TenantRoutingS3ContentStore`.

```
<bean id="tenantRoutingContentStore"
  class="org.alfresco.module.org_alfresco_module_cloud.repo.content.s3store.TenantR
  parent="baseTenantRoutingContentStore">

  <property name="defaultRootDir" value="\${dir.contentstore}" />
  <property name="s3AccessKey" value="\${s3.accessKey}" />
  <property name="s3SecretKey" value="\${s3.secretKey}" />
  <property name="s3BucketName" value="\${s3.bucketName}" />
  <property name="s3BucketLocation" value="\${s3.bucketLocation}" />
  <property name="s3FlatRoot" value="\${s3.flatRoot}" />
  <property name="globalProperties">
    <ref bean="global-properties" />
  </property>
</bean>
```



Remember to change this bean's ID to `backingStore` for use with the preceding XML snippet, or change the `ref` attribute in the `fileContentStore` bean definition to refer to the correct ID (`tenantRoutingContentStore`).

3. Define a `ContentCache`. This object is responsible for placing content into (and retrieving content from) the cache.

```
<bean id="contentCache"
  class="org.alfresco.repo.content.caching.ContentCacheImpl">
  <property name="memoryStore" ref="cachingContentStoreCache"/>
  <property name="cacheRoot" value="\${dir.cachedcontent}"/>
</bean>
```

The `ContentCacheImpl` uses a fast lookup table for determining whether an item is currently cached by the CCS, for controlling the maximum number of items in the cache and their Time To Live (TTL). The lookup table is specified here by the `memoryStore` property. The `ContentCacheImpl` also uses a directory on the local filesystem for storing

binary content data (the actual content being cached). This directory is specified by the `cacheRoot` property. The following code illustrates the bean referencing the specified `memoryStore` reference:

```
<bean id="cachingContentStoreCache" factory-bean="cacheFactory" factory-
method="createCache">
  <constructor-arg value="cache.cachingContentStoreCache" />
</bean>
```

- Now that you have configured the key components of the `CachingContentStore` class, backing store (`ContentStore`) and `ContentCache`, you can optionally specify a quota manager. If you do not wish to specify the quota manager, then the `UnlimitedQuotaStrategy` will be used. The example CCS bean expects this bean to be defined:

```
<bean id="standardQuotaManager"

  class="org.alfresco.repo.content.caching.quota.StandardQuotaStrategy"
    init-method="init"
    destroy-method="shutdown">
  <property name="maxUsageMB" value="4096" />
  <property name="maxFileSizeMB" value="0" />
  <property name="cache" ref="contentCache" />
  <property name="cleaner" ref="cachedContentCleaner" />
</bean>
```

- Finally, to ensure that the disk space is used in a controlled manner, a `CachedContentCleaner` should be configured to clean up cached content files that are no longer being used by the cache.

```
bean id="cachingContentStoreCleanerJobDetail"
  class="org.springframework.scheduling.quartz.JobDetailBean">
  <property name="jobClass">

    <value>org.alfresco.repo.content.caching.cleanup.CachedContentCleanupJob</
value>
    </property>
    <property name="jobDataAsMap">
      <map>
        <entry key="cachedContentCleaner">
          <ref bean="cachedContentCleaner" />
        </entry>
      </map>
    </property>
  </bean>

  <bean id="cachedContentCleaner"

  class="org.alfresco.repo.content.caching.cleanup.CachedContentCleaner"
    init-method="init">
    <property name="minFileAgeMillis"
value="{system.content.caching.minFileAgeMillis}" />
    <property name="maxDeleteWatchCount"
value="{system.content.caching.maxDeleteWatchCount}" />
    <property name="cache" ref="contentCache" />
    <property name="usageTracker" ref="standardQuotaManager" />
  </bean>

  <bean id="cachingContentStoreCleanerTrigger"
  class="org.alfresco.util.CronTriggerBean">
    <property name="jobDetail">
      <ref bean="cachingContentStoreCleanerJobDetail" />
    </property>
    <property name="scheduler">
      <ref bean="schedulerFactory" />
    </property>
    <property name="cronExpression">
```

```

        <value>
    ${system.content.caching.contentCleanup.cronExpression}</value>
    </property>
</bean>

```

Note that both the cleaner and the quota manager limit the usage of disk space but they do not perform the same function. In addition to removing the orphaned content, the cleaner's job is to remove files that are out of use from the cache due to parameters, such as TTL, which sets the maximum time an item should be used by the CCS. The quota manager exists to set specific requirements in terms of allowed disk space.

A number of property placeholders are used in the specified definitions. You can replace them directly in your configuration with the required values, or you can use the placeholders as they are and set the values in the `repository.properties` file. An advantage of using the property placeholders is that the sample file can be used with very few changes and the appropriate properties can be modified to get the CCS running with little effort.

Aggregating content store

An Aggregating content store (`AggregatingContentStore`) is a content store implementation that aggregates a set of stores. This topic provides information on the Aggregating content store and its configuration details.

The Aggregating content store contains a primary store and a set of secondary stores. The order in which the stores appear in the list of participating stores is important. The first store in the list is known as the primary store. Content can be read from any of the stores, as if it were a single store. When the replicator goes to fetch content, the stores are searched from first to last. The stores should therefore, be arranged in order of speed.

For example, if you have a fast (and expensive) local disk, you can use this as your primary store for best performance. The old infrequently used files may be stored on lower cost, slower storage.

When replication is disabled, content is written to the primary store only. The other stores are used to retrieve content and the primary store is not updated with the content.

Example configuration for tiered storage

The following configuration defines an additional tiered storage solution. The default content store is not changed. An additional set of secondary stores is defined (`tier1`, `tier2` and `tier3`). As content ages (old infrequently used files), it can be moved to lower tiers. If the tiered storage is slow, a Caching content store can be placed in front.

1. In your `alfresco-global.properties` file, define three new folder locations:
 - `dir.contentstore1=${dir.root}/tier1`
 - `dir.contentstore2=${dir.root}/tier2`
 - `dir.contentstore3=${dir.root}/tier3`
2. Locate the `<TOMCAT_HOME>/shared/classes/alfresco/extension/aggregating-store-context.xml.sample` file.
3. Remove the `.sample` extension from this file.

The `aggregating-store-context.xml` file enables Aggregating content store. The content of this file is shown below. Place the `aggregating-store-context.xml` file in your `<TOMCAT_HOME>/shared/classes/alfresco/extension` folder.

```

<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" 'http://
www.springframework.org/dtd/spring-beans.dtd'>

<!-- This file enables an aggregating content store. It should be placed in
shared/classes/alfresco/extension -->

```

```

<beans>

  <bean id="defaultContentStore"
class="org.alfresco.repo.content.filestore.FileContentStore">
  <constructor-arg>
    <value>${dir.contentstore}</value>
  </constructor-arg>

  <!-- Uncomment the property below to add content filesize limit.
  <property name="contentLimitProvider" ref="defaultContentLimitProvider"/>
  -->
</bean>

  <bean id="tier1ContentStore"
class="org.alfresco.repo.content.filestore.FileContentStore">
  <constructor-arg>
    <value>${dir.contentstore1}</value>
  </constructor-arg>

  <!-- Uncomment the property below to add content filesize limit.
  <property name="contentLimitProvider" ref="defaultContentLimitProvider"/>
  -->
</bean>

  <bean id="tier2ContentStore"
class="org.alfresco.repo.content.filestore.FileContentStore">
  <constructor-arg>
    <value>${dir.contentstore2}</value>
  </constructor-arg>

  <!-- Uncomment the property below to add content filesize limit.
  <property name="contentLimitProvider" ref="defaultContentLimitProvider"/>
  -->
</bean>

  <bean id="tier3ContentStore"
class="org.alfresco.repo.content.filestore.FileContentStore">
  <constructor-arg>
    <value>${dir.contentstore3}</value>
  </constructor-arg>

  <!-- Uncomment the property below to add content filesize limit.
  <property name="contentLimitProvider" ref="defaultContentLimitProvider"/>
  -->
</bean>

  <!-- this is the aggregating content store - the name fileContentStore
  overrides the alfresco default store -->
  <bean id="fileContentStore"
    class="org.alfresco.repo.content.replication.AggregatingContentStore" >

    <property name="primaryStore" ref="defaultContentStore" />

    <property name="secondaryStores">
      <list>
        <ref bean="tier1ContentStore" />
        <ref bean="tier2ContentStore" />
        <ref bean="tier3ContentStore" />
      </list>
    </property>

  </bean>
</beans>

```

Encrypted Content Store

Alfresco Version 5.0 includes the Encrypted Content Store sub-system, which provides the ability to encrypt content at rest. This section provides an overview of this sub-system, its components and how it is administered.

Encrypted Content Store overview

This section explains Alfresco's implementation of Encrypted Content Store.

- ⚠ Once you make the decision to use Encrypted Content Store, it is irrevocable. This is because when a document is written to the Encrypted Content Store, it is encrypted. If you decide to revert to an unencrypted content store, the content cannot be decrypted.

Alfresco cryptography process

The Encrypted Content Store provides encryption at rest capability. This is done by scrambling plain text into cipher text (encryption) and then back again (decryption) with the help of symmetric and asymmetric keys.

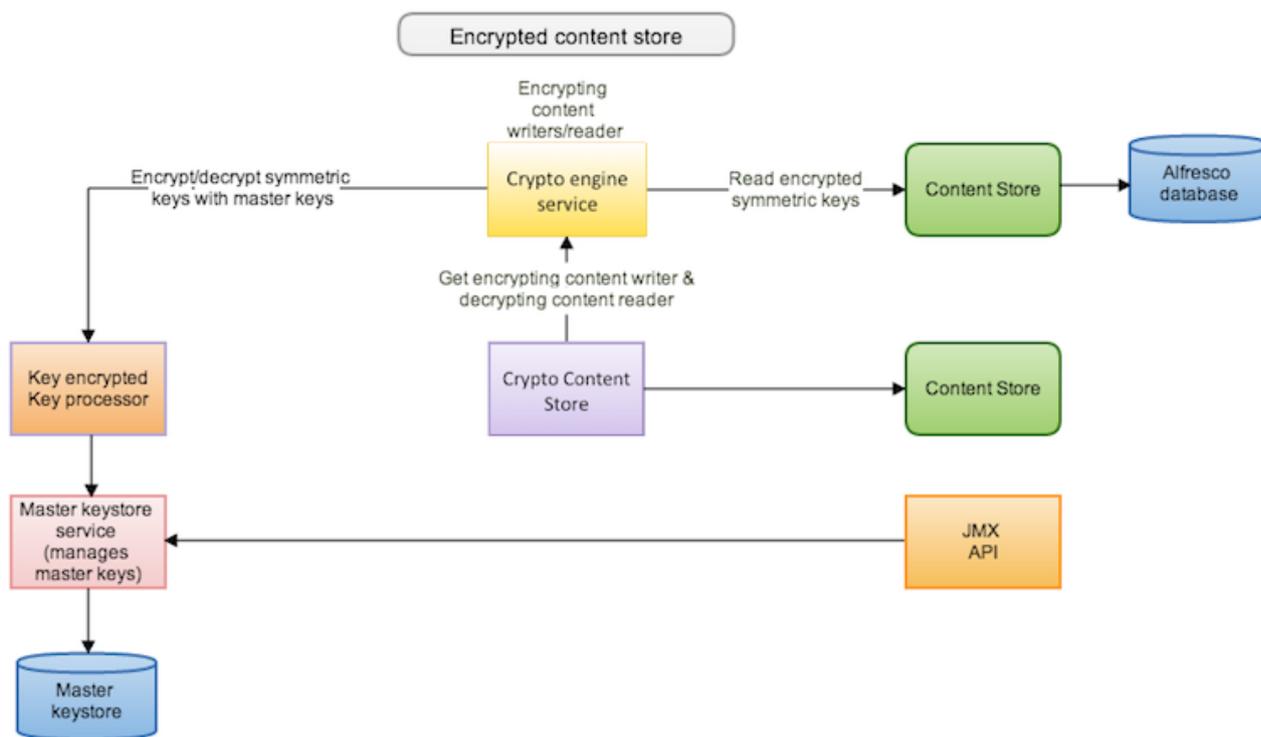
When a document is written to the Encrypted Content Store, the Encrypted Content Store uses symmetric encryption to encrypt the document before it is written to the wrapped content store. A new symmetric key is generated each time a document is written to the content store. This means that every document in the system is encrypted with a different symmetric key. Further more, asymmetric encryption (such as RSA) is used to encrypt/decrypt those symmetric encryption/decryption keys. The asymmetric encryption uses a master key which is selected from a set of configured master keys.

Alfresco uses a set of master keys, which are:

- selected in a random fashion
- stored in a password-protected keystore
- can be retired, in the event of key theft or as part of a standard key retirement process. For more information, see [Encryption-related JMX operations](#).

The repository knows which master key was used to encrypt a given symmetric key so that when a user reads a particular document, the repository can decrypt the symmetric key (using that master key) and then use the decrypted symmetric key to decrypt the document content.

The following diagram shows the application of Encrypted Content Store over your default Alfresco content store.



Issues to consider before using Encrypted Content Store

This topic outlines issues you should consider before using Encrypted Content Store.

- Because encryption is done at the content store level and not just for a few files in the repository, you must make sure whether encryption is needed at the content store level or not.
- Once you make the decision to use Encrypted Content Store, it is irrevocable. This is because when a document is written to the Encrypted Content Store, it is encrypted. If you decide to revert to an unencrypted content store, the content cannot be decrypted.
- Encrypted Content Store is not supported in conjunction with the Alfresco S3 Connector and Centera content store.
- Encrypted Content Store is separately licensed and requires that you receive a license key from Alfresco.
- Multi-tenancy is not supported by Encrypted Content Store.

Installing the Encrypted Content Store

This topic describes the steps to install the Encrypted Content Store to an instance of Alfresco.

To use Encrypted Content Store, you must ensure that you have Alfresco One 5.0 installed on your machine. For more information, see [Installing Alfresco using setup wizards](#).

1. Obtain the license (.lic) file with content encryption enabled from Alfresco.
2. Apply the license into the `<ALFRESCO_HOME>/tomcat/shared/classes/alfresco/extension/license` directory.
3. Generate the RSA master key(s) in a new keystore.

For example, use the following command to generate the master key:

```
keytool -genkey -alias key1 -keyalg RSA -keystore <master keystore path> -keysize 2048
```

4. Follow the instructions for configuring the [Encrypted Content Store](#).

Configuring the Encrypted Content Store

This topic describes how to configure the Encrypted Content Store.

The Encrypted Content Store is configured using the properties in the `alfresco-global.properties` file and can be administered using JMX.

Set up encryption properties using `alfresco-global.properties` file

To configure the Encrypted Content Store, set the configuration properties in the `alfresco-global.properties` file. For example, here is a set of properties which configure the default Java JCE provider implementation with two master keys (override the `cryptodoc.jce.providerName` property to configure a specific provider).

```
filecontentstore.subsystem.name=encryptedContentStore
cryptodoc.jce.keystore.path=/Users/sglover/dev/projects/PLATFORM1-CRYPTO/data/crypto.jks
cryptodoc.jce.keystore.password=password
cryptodoc.jce.key.aliases=mkey1,mkey2
cryptodoc.jce.key.passwords=password,password
cryptodoc.jce.keygen.defaultSymmetricKeySize=128
```

For detailed information on these properties, see [Encrypted Content Store properties](#).



On Enterprise, properties edited using JMX are persisted and will override any settings in the `alfresco-global.properties` file. Use the JMX client to change the configuration properties on Enterprise.

Set up encryption properties using JMX client

You can configure the Encrypted Content Store using the JMX client, such as JConsole on the **JMX MBeans > Alfresco > Configuration > ContentStore > managed > encrypted > Attributes** tab.

The screenshot shows the JConsole interface with the 'MBeans' tab selected. The left-hand tree view shows the hierarchy: Alfresco > Configuration > ContentStore > managed > encrypted > Attributes. The right-hand pane displays the 'Attribute values' for the selected MBean.

Name	Value
\$type	encrypted
cryptodoc.jce.key.aliases	mkey2,mkey3,mkey4
cryptodoc.jce.key.passwords	password,password,password
cryptodoc.jce.keygen.defaultSymmetricAlgorithm	AES
cryptodoc.jce.keygen.defaultSymmetricKeySize	128
cryptodoc.jce.keystore.password	password
cryptodoc.jce.keystore.path	/Users/sglover/dev/projects/PLATFORM1-CRYPTO/data/c...
cryptodoc.jce.keystore.type	jceks
cryptodoc.jce.providerName	
cryptodoc.symmetricKey.reencryption.batch.size	200
cryptodoc.symmetricKey.reencryption.numThreads	4
instancePath	[managed, encrypted]

A 'Refresh' button is located at the bottom right of the attribute values pane.

Encrypted Content Store properties

This topic describes the properties that need to be set for the Encrypted Content Store.

Set these properties in the `alfresco-global.properties` file.

filecontentstore.subsystem.name=encryptedContentStore

Enables the Encrypted Content Store subsystem.

cryptodoc.jce.providerName

Specifies the Java security provider name. If left blank, it indicates using the default provider. You can also select your own provider by setting this property to the provider class name. If a specific provider name is not set, the system selects the most preferred provider.

cryptodoc.jce.keystore.type

Specifies the keystore type (for example, `jceks`)

cryptodoc.jce.keystore.path

Specifies the path to the keystore containing the master keys.

cryptodoc.jce.keystore.password

Specifies the keystore password.

cryptodoc.jce.key.aliases

Specifies a comma-separated list of the aliases/names of the master keys in the master keystore.

cryptodoc.jce.key.passwords

Specifies a comma-separated list of passwords that Alfresco will use to load the keys from the master key store. The position of the password matches the position of the corresponding key alias in the `cryptodoc.jce.key.aliases` property.

cryptodoc.jce.keygen.defaultSymmetricKeySize

Specifies the key size to use for the symmetric keys that are used to encrypt/decrypt document content.



The default symmetric key size is 128 bits. Users who want better key strength should download and install the [Java Cryptography Extension \(JCE\) Unlimited Strength Jurisdiction Policy Files](#) for the JRE.

cryptodoc.jce.keygen.defaultSymmetricAlgorithm

Specifies the symmetric key algorithm.

The following properties are used to re-encrypt symmetric keys (for master key revocation).

cryptodoc.symmetricKey.reencryption.batch.size=200

Specifies the number of symmetric keys re-encrypted in each batch.

cryptodoc.symmetricKey.reencryption.numThreads=4

Specifies the number of threads to use to perform re-encryption.

The keystore path, password, aliases and their password are the common properties you can overwrite to configure Encrypted Content Store using the `alfresco-global.properties` file.

The JMX interface exposes these properties and allows the user to change them for a running system. For more information, see [Encryption-related JMX operations](#).

Encryption-related JMX operations

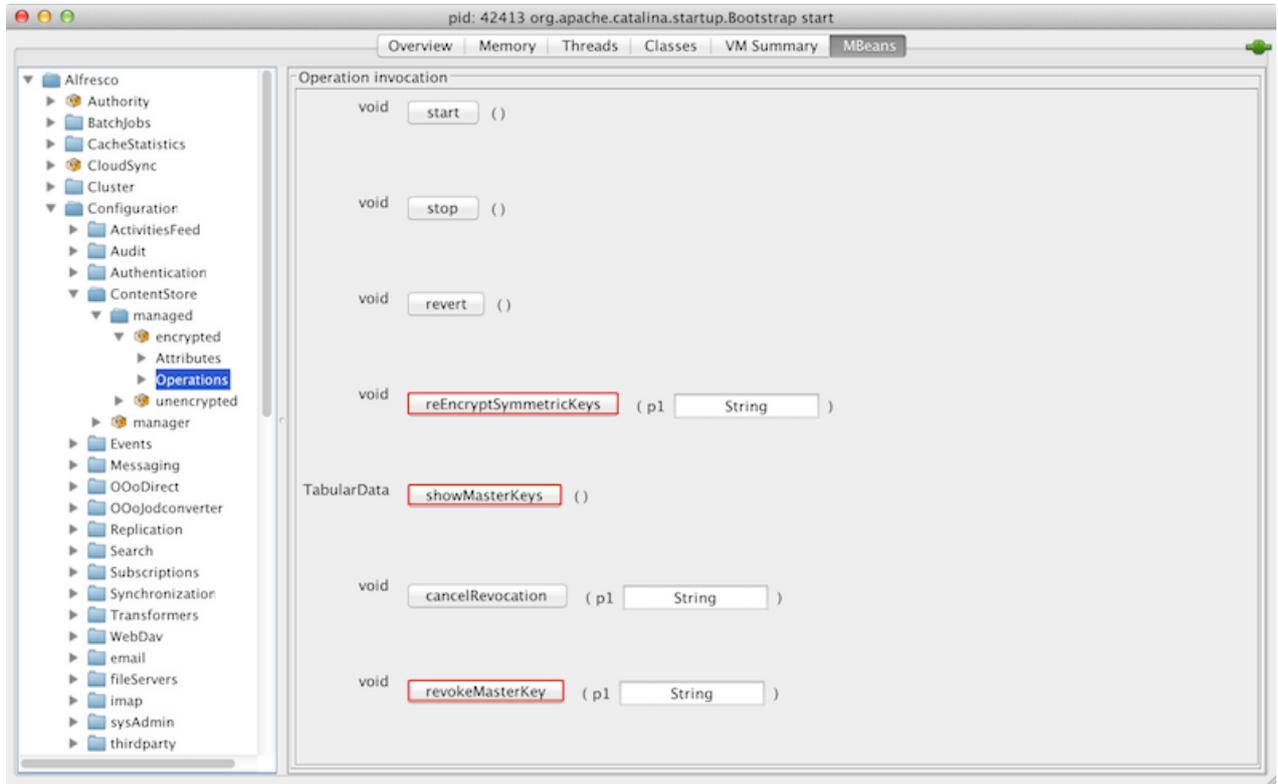
This topic describes the use of JMX operations for performing some of the most common tasks for Encrypted Content Store.

The JMX client, JConsole allows the user to see the set of current master keys and the total number of symmetric keys encrypted by each master key. It also enables the users to revoke a master key and to add a new master key alias.

Retire a master key

To retire a master key, follow the sequence of JMX operations below:

1. On the **JConsole** window, select the **MBeans** tab.
The available managed beans are displayed in JConsole.
2. Navigate to **Alfresco > Configuration > ContentStore > managed > encrypted > Operations**.
The **Operation invocation** window is displayed.



3. Click **revokeMasterKey** to stop the relevant master key from being used for encryption.
The master key is now no longer available for encryption.
4. Click **reEncryptSymmetricKeys** to reencrypt the symmetric keys of this master key with a new master key.
5. Click **showMasterKeys** to check that there are no outstanding symmetric keys for the revoked master key and that the total number of files that were encrypted using the revoked master key is zero.
6. Click **stop** to stop the Encrypted Content Store subsystem.
7. Remove the relevant alias and related password from **MBeans > Alfresco > Configuration > ContentStore > managed > encrypted > Attributes > Attribute values** window.
8. Click **start** to restart and reinitialize the Encrypted Content Store subsystem.

 If you update or remove a master key using the JMX client on an Enterprise installation, those updates override the values in the `alfresco-global.properties` file. Alternatively, one can delete the master key alias and password by editing the `alfresco-global.properties` file and restarting the repository.

Add a new master key

To add a master key, follow the steps below:

1. Add the new master key to the master `keystore` file.
2. Define the new master key alias and password by one of the following ways:
 - Add the key alias and password in the `alfresco-global.properties` file; or
 - Add the key alias and password by using the JMX operations. Follow the sequence of steps from Step 3 onwards.

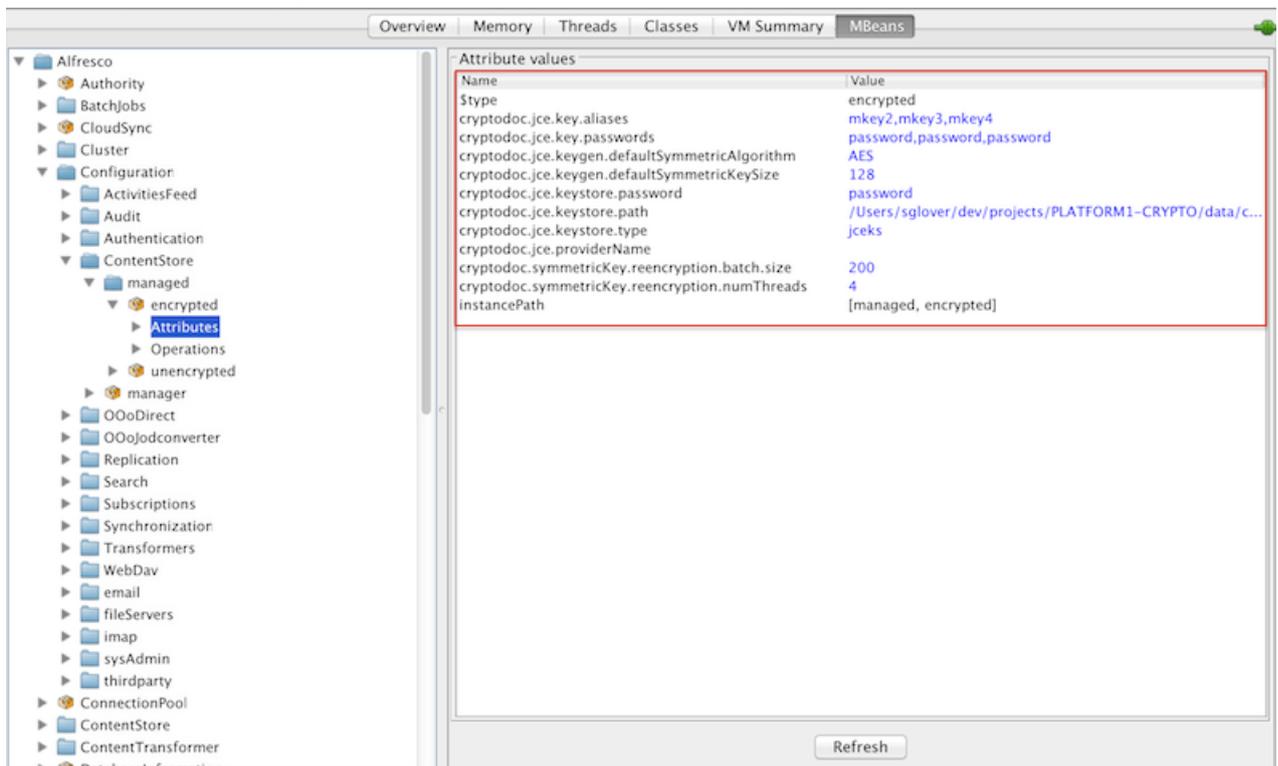
 The values set on a subsystem will mean that the property values from configuration files may be overwritten or ignored. Use the JMX client to set the configuration properties.

3. On the **JConsole** window, select the **MBeans** tab.

The available managed beans are displayed in JConsole.

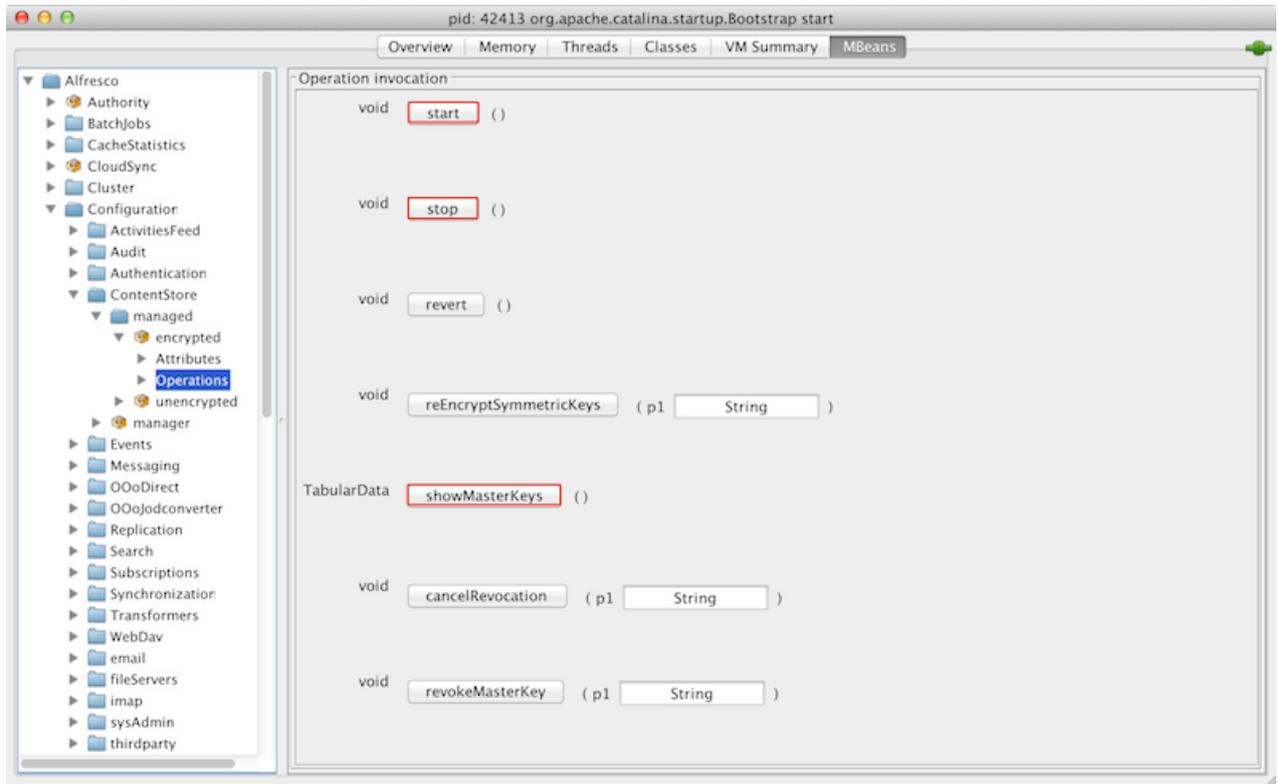
4. Navigate to **Alfresco > Configuration > ContentStore > managed > encrypted > Attributes**.

The **Attribute values** window is displayed.



Name	Value
Stype	encrypted
cryptodoc.jce.key.aliases	mkey2,mkey3,mkey4
cryptodoc.jce.key.passwords	password,password,password
cryptodoc.jce.keygen.defaultSymmetricAlgorithm	AES
cryptodoc.jce.keygen.defaultSymmetricKeySize	128
cryptodoc.jce.keystore.password	password
cryptodoc.jce.keystore.path	/Users/sglover/dev/projects/PLATFORM1-CRYPTO/data/c...
cryptodoc.jce.keystore.type	jceks
cryptodoc.jce.providerName	
cryptodoc.symmetricKey.reencryption.batch.size	200
cryptodoc.symmetricKey.reencryption.numThreads	4
instancePath	[managed, encrypted]

5. On the **Operation invocation** window, click **stop** to stop the Alfresco subsystem.



6. On the **Attribute values** window, add a new key alias in the **cryptodoc.jce.key.aliases** field and its password in the **cryptodoc.jce.key.passwords** field. Both these fields accept comma-separated list of values.

 While adding a new master key alias, if you add the alias but not the password, the master key will fail to register.

7. Click **start** to restart and reinitialize the Encrypted Content Store subsystem.
8. Click **showMasterKeys** to check that the new master key is now being used.

Expiry of a master key

The Encrypted Content Store sub-system does not support automatic expiry of the master key. When a master key expires from the keystore, you must follow the sequence of JMX operations mentioned in the **Retire a master key** section above to manually retire the master key.

Additional JMX operations

- Click **cancelRevocation** to cancel revocation of the master key. This ensures that the previously revoked master key is now being used.
- Click **reloadMasterKeys** to reload the master keys from the `keystore` file.

Centera content store

The Centera Connector module provides integration between Alfresco and Content Addressable Storage (CAS) systems. This topic provides an overview on the Centera content store.

CAS systems store and locate files using addresses based on the file's content, rather than a physical location address. CAS systems are typically used for long-term storage of content that does not require frequent access or where it is stored for regulatory purposes.

When a CAS system stores content, it generates a unique key or hash, which is based on the content. The hash is generated from the content properties, such as the name, date, or content itself.

An example hash might be `EQM2GC012MC77e72B24N2MMFU59G418ACSAIE70BAS340TN3E1JJL`. This hash is then used as the address of the stored content, and which is then used to retrieve the content. When a request is made to the CAS using this address, it returns the associated content.

The benefits of using CAS systems are:

- Content can be located easily even in large volumes of data
- Content integrity: if stored content has been altered then there is a mismatch between the hash passed as the address and hash computed on the fly
- Avoids redundancy by recognizing that the hash is already present and so does not store it again

For more information on installing and configuring Centera Connector, and setting up `CenteraContentStore` as your main content store, see [Installing and configuring Centera Connector](#).

Content store selector

The content store selector provides a mechanism to control the store used for the content file associated with a particular content item.

By applying the `cm:storeSelector` aspect and setting its `cm:storeName` property to the name of a selectable store, the content will be automatically moved from its current location to the new store. The store does not, therefore, store content itself, it defines and manages those stores that are available for selection.

This allows storage policies to be implemented to control which underlying physical storage is used, based on your applications needs or business policies.

Content store selector configuration example

The following example defines two file stores, in addition to the standard default file store. By setting the `cm:storeName` property to either of these new stores or the default store, the content is automatically moved from its existing store to the relevant new store.

1. Create a `sample-content-store-selector-context.xml` file in the `<extension>` directory.
2. Define the new file stores by adding the following bean definitions:

```
<bean id="firstSharedFileContentStore"
  class="org.alfresco.repo.content.filestore.FileContentStore">
  <constructor-arg>
    <value>${dir.root}/storeA</value>
  </constructor-arg>
</bean>

<bean id="secondSharedFileContentStore"
  class="org.alfresco.repo.content.filestore.FileContentStore">
  <constructor-arg>
    <value>${dir.root}/storeB</value>
  </constructor-arg>
</bean>
```

This configuration snippet defines two new stores. The physical location is relative to the `dir.root` property defined in the `alfresco-global.properties` file.

3. Declare the `storeSelectorContentStore` to be the primary content store by adding the following bean definition:

```
<bean id="contentService" parent="baseContentService">
  <property name="store">
    <ref bean="storeSelectorContentStore" />
  </property>
</bean>
```

```

    </property>
  </bean>

```

4. Declare the mapping between store names and store instances.

```

<bean id="storeSelectorContentStore"
  parent="storeSelectorContentStoreBase">
  <property name="defaultStoreName">
    <value>default</value>
  </property>
  <property name="storesByName">
    <map>
      <entry key="default">
        <ref bean="fileContentStore" />
      </entry>
      <entry key="storeA">
        <ref bean="firstSharedFileContentStore" />
      </entry>
      <entry key="storeB">
        <ref bean="secondSharedFileContentStore" />
      </entry>
    </map>
  </property>
</bean>

```

The list of stores is defined by the `<property name="storesByName">` property. Any stores you want to be available to the `storeSelectorContentStore` should be listed under this property.

5. Add the extra stores to the list to be handled by the `eagerContentStoreCleaner`.

Using the new content store

The new content store is set using the `cm:storeName` property.

The `cm:storeName` property can be set in number of ways:

- Manually, by exposing this property so its value can be set by Share
- Running a script action that sets the `cm:storeName` property value within the script
- Using a rule that runs a script action to set the property

The expected behavior is as follows:

- When the `cm:storeSelector` aspect is not present or is removed, the content is copied to a new location in the 'default' store
- When the `cm:storeSelector` aspect is added or changed, the content is copied to the named store
- Under normal circumstances, a trail of content will be left in the stores, just as it would be if the content were being modified. The normal processes to clean up the orphaned content will be followed.

Content Store Selector full configuration example

The following example shows the full definition of creating new stores using the Content Store Selector.

This configuration must be saved as an extension, for example, `<extension>\sample-content-store-selector-context.xml`.



The list of stores available can be set by updating the list under the `<property name="storesByName">` property.

```

<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" 'http://
www.springframework.org/dtd/spring-beans.dtd'>

```

```

<beans>

  <bean id="firstSharedFileContentStore"
class="org.alfresco.repo.content.filestore.FileContentStore">
    <constructor-arg>
      <value>${dir.root}/storeA</value>
    </constructor-arg>
  </bean>

  <bean id="secondSharedFileContentStore"
class="org.alfresco.repo.content.filestore.FileContentStore">
    <constructor-arg>
      <value>${dir.root}/storeB</value>
    </constructor-arg>
  </bean>

  <bean id="storeSelectorContentStore" parent="storeSelectorContentStoreBase">
    <property name="defaultStoreName">
      <value>default</value>
    </property>
    <property name="storesByName">
      <map>
        <entry key="default">
          <ref bean="fileContentStore" />
        </entry>
        <entry key="storeA">
          <ref bean="firstSharedFileContentStore" />
        </entry>
        <entry key="storeB">
          <ref bean="secondSharedFileContentStore" />
        </entry>
      </map>
    </property>
  </bean>

<!-- Point the ContentService to the 'selector' store -->
  <bean id="contentService" parent="baseContentService">
    <property name="store">
      <ref bean="storeSelectorContentStore" />
    </property>
  </bean>

  <!-- Add the other stores to the list of stores for cleaning -->
  <bean id="eagerContentStoreCleaner"
class="org.alfresco.repo.content.cleanup.EagerContentStoreCleaner" init-
method="init">
    <property name="eagerOrphanCleanup" >
      <value>${system.content.eagerOrphanCleanup}</value>
    </property>
    <property name="stores" >
      <list>
        <ref bean="fileContentStore" />
        <ref bean="firstSharedFileContentStore" />
        <ref bean="secondSharedFileContentStore" />
      </list>
    </property>
    <property name="listeners" >
      <ref bean="deletedContentBackupListeners" />
    </property>
  </bean>

</beans>

```

The following example shows the web-client-config-custom.xml file:

```

<!-- Configuring in the cm:storeSelector aspect -->
<config evaluator="aspect-name" condition="cm:storeSelector">

```

```

    <property-sheet>
      <show-property name="cm:storeName" />
    </property-sheet>
  </config>
  <config evaluator="string-compare" condition="Action Wizards">
    <aspects>
      <aspect name="cm:storeSelector"/>
    </aspects>
  </config>
  ...

```

Managing content stores

This topic describes how to effectively manage the File content store and Deleted content store.

The **File content store** saves the files or content items on a file system under the root directory. The `${dir.contentstore}` property points to the root location on the file system. Files are organised by time to assist with incremental backup.

The **Deleted content store** saves orphaned files that are removed (nightly, by default) by the content store cleaner. The `${dir.contentstore.deleted}` property points to the location where deleted files are stored. The default deleted content store is a file content store.

When you create a file, a `.bin` file is stored in the default file content store and there is a reference on that `.bin` file in the database. When you delete the document, Alfresco updates the database. When you purge the deleted items, Alfresco destroys all references to that `.bin` file in database. When the scheduled job runs, it scans the database and the `contentstore` directory and moves everything that is not referenced in the database to the `<ALFRESCO_HOME>\alf_data\contentstore.deleted` directory. The content of the `contentstore.deleted` directory is not referenced anywhere. So, you can always delete the contents of this directory (normally just after a backup). You can have your own Operating System cron job that purges contents of this folder periodically.

The `repository.properties` file defines the `fileContentStore` and `deletedContentStore` properties.

```

# The location of the content store
dir.contentstore=${dir.root}/contentstore
dir.contentstore.deleted=${dir.root}/contentstore.deleted

```

You can configure these properties by overriding them in the `alfresco-global.properties` file.

 You can use a remote file system but you cannot use the UNC mapped network path with it, for example:

```

dir.contentstore=//server1/c/contentstore/contentstore
dir.contentstore.deleted=//server1/c/contentstore/contentstore.deleted

```

You need to use a Windows or DOS path.

To select a content stores, you have to choose the required subsystem:

```

filecontentstore.subsystem.name=unencryptedContentStore

```

The default, unencrypted store is a simple file storage store with its root in `dir.contentstore=${dir.root}/contentstore`. A date-time file structure is used, which makes the store easy to backup and browse. Most commonly, the `dir.contentstore` points to a shared file system when Alfresco is deployed in a cluster. This is fully supported. Any regular file system backup procedure will work without the danger of corruption or loss of data. As a good practice, you should take the database backup before you take the file system backup.

Cleaning up orphaned content (purge)

This topic describes how to delete or purge orphaned content from the content store while the system is running.

The `contentStoreCleaner` bean identifies and deletes the orphaned content. In the default configuration, the `contentStoreCleanerTrigger` calls the `contentStoreCleaner` bean.

```
<bean id="contentStoreCleaner"
  class="org.alfresco.repo.content.cleanup.ContentStoreCleaner" >
  ...
  <property name="protectDays" >
    <value>14</value>
  </property>
  <property name="stores" >
    <list>
      <ref bean="fileContentStore" />
    </list>
  </property>
  <property name="listeners" >
    <list>
      <ref bean="deletedContentBackupListener" />
    </list>
  </property>
</bean>
```

Properties:

- **protectDays**

This property specifies the minimum time that content binaries should be kept in the `contentStore`.

In the above example, if a file is created and immediately deleted, it will not be cleaned from the `contentStore` for at least 14 days. The value should be adjusted to account for backup strategies, average content size, and available disk space. Setting this value to zero will result in a system warning as it breaks the transaction model. Also, it is possible to lose content if the orphaned content cleaner runs while the content is being loaded into the system. If the system backup strategy is just to make regular copies, then this value should be greater than the number of days between successive backup runs.

- **store**

This property displays a list of `ContentStore` beans to scour for orphaned content.

- **listeners**

This property specifies the listeners, which are notified when an orphaned content is located.

In the above example, the `deletedContentBackupListener` copies the orphaned content to a separate `deletedContentStore`. Note that this configuration will not actually remove the files from the file system but rather move them to the designated `deletedContentStore`, usually `contentstore.deleted`. Once an appropriate backup has been performed, the files can be removed from the `deletedContentStore` via script or cron job.

Configuring templated nodes and space templates

With templated nodes and space templates you can store content and folder templates in Alfresco repositories that users can then use to create content.

Templated nodes provide a convenient way for users to quickly create content based on a pre-determined style, such as documents already formatted to company guidelines. Once a template has been stored in the Alfresco repository users can create new content items based upon it from the **Create** menu in the **Document Library**.

Space templates let users quickly create folders. Content or other folder structures can be contained in the folder and are replicated when a folder is created from the template. Once a

template has been stored in the Alfresco repository users can create new folders based upon it from the **Create** menu in the **Document Library**.

You can have an unlimited number of templated nodes and space templates.

Setting files as templates

In the Node Templates folder you can store documents that users can then use as document templates.

1. In the Alfresco **Repository** open the **Data Dictionary** then **Node Templates**.
2. Either drag and drop a content item that you want to use as a template onto the **Node Templates** drag and drop area, or click **Upload** and browse to and select the required file.



The standard Alfresco **Create** options are also available, so that you can create templates directly from Alfresco, in just the same way as a user would create new content.

If you already have templated nodes set up, you can select **Create document from template** from the **Create** menu and create additional templates based on your existing templates.

3. Click **OK** when the upload is complete.

The file is now available to your users as a template when they select to create content **Create document from template**.

Setting folders as templates

In the Space Templates folder you can store folders that users can then use as folder templates.

1. In the Alfresco **Repository** open the **Data Dictionary** then **Space Templates**.
2. Click **Create** then **Folder**.

The new folder is added to the Space Templates folder. You can add content or further folders to this folder to create an entire folder structure. If users create a folder from this template the whole structure and all its contents will be replicated.



If you already have space templates set up, you can select **Create folder from template** from the **Create** menu and create additional templates based on your existing templates.

The file is now available to your users as a template when they select to create content **Create document from template**.

Setting up Alfresco multi-tenancy

Alfresco also supports multi-tenancy (MT) features that enable Alfresco to be configured as a true single-instance, multi-tenant environment. Multi-tenancy allows multiple, independent tenants to be hosted on a single instance, which can be installed either on a single server or across a cluster of servers. The Alfresco instance is logically partitioned such that it will appear to each tenant that they are accessing a completely separate instance of Alfresco.

Enabling multi-tenancy

In Alfresco, the multi-tenancy feature is pre-configured out-of-the-box, although it is not enabled by default.

When you install Alfresco, multi-tenancy is disabled. The multi-tenancy feature is automatically enabled when the first tenant is created.

 Only an Administrator user can create tenants.

However, if you wish to disable multi-tenancy, you need to delete all the tenants. See the [Managing tenants](#).

Multi-tenancy administration

For example, if a tenant/organization called `acme` is created, the tenant administrator can log in as `admin@acme` and create users such as `alice@acme`, and `bob@acme`.

The administration features available to the tenant administrator include:

- Manage system users (including user Usages and Quotas)
- Manage user groups
- Category management
- Export and import
- System information
- Node browser

For more information on administration, see [Using the Admin Console](#) and [Using the Share Admin Tools](#).

Multi-tenancy export and import

 Repository export does not apply to certain areas, such as in-flight workflows. A repository import must be into the same version of Alfresco from which the export was performed.

1. Log in to Alfresco as the `admin` user and access: `http://localhost:8080/alfresco/faces/jsp/admin/tenantadmin-console.jsp`

2. Use the export feature to export a tenant:

```
export <tenant domain> <destination directory>
```

This exports the tenant to a set of repository export files in a given destination directory. Export file names will be suffixed with `<tenant domain>_.`

3. Use the import feature to import a tenant:

```
import <tenant domain> <source directory> [<root contentstore dir>]
```

This creates a tenant by importing the tenant files from the given source directory. The import file names must be suffixed with `<tenant domain>_.`

 If an existing tenant needs to be re-imported, the tenant must be deleted first. To cleanly delete a tenant, the server must be restarted to clear the index threads. The tenant-specific index directories and tenant-specific content directories must also be manually deleted before starting the import.

Multi-tenancy implementation

All Alfresco-related services are partitioned including node services, security services, workflow services, search and index services, and dictionary services. To support Alfresco Share in a multi-tenant environment, additional partitioned services include site services, activity services, invite services, and AVM services.

The metadata is logically partitioned within the database schema.

Logging enables nested diagnostic context (NDC). For a single tenant environment, the log output will show the user name context. For a multi-tenant environment, the log output also shows the tenant context.

Modules

Alfresco supports the ability to pre-package AMPs (Alfresco Module Packages) into the Alfresco WAR, which are installed into the default domain on start up. In a multi-tenant environment, the module is also installed into each tenant domain when the tenant is created or imported.

Features not supported in a multi-tenant environment

There are some features and components that are not supported in a multi-tenant production environment.

Using multi-tenancy you can configure multiple, independent tenants on a single Alfresco instance. However, multi-tenancy is not supported in the following methods and modules:

- Record Management
- Solr
- CIFS
- Portlets
- LDAP, NTLM and authentication methods other than `alfresco`
- Inbound email
- Content replication
- IMAP
- Encrypted Content Store

Setting up content replication

Content replication is designed to assist geographically distributed deployments where performance may be affected by network latency or bandwidth limitations.

- Fast access by serving content from local servers
- High availability - removes the single point of failure

For network administrators replication provides:

- Reduced network overhead

By default, replicated content is read-only on the target repository. This ensures the integrity of the content is not compromised by uncontrolled updates. A option is provided in the Alfresco Share user interface for users to navigate to the content's source repository to make any updates.

The replication service controls content replication between different Alfresco repositories. The replication service is responsible for persisting replication jobs that specify what is to be replicated, to where, and when. In addition, it monitors the status of currently executing replication jobs and enables replications to be canceled.

The replication service finds the nodes that need to be transferred, and then it delegates the transfer of content to the transfer service.

Replication jobs are managed in the Share Admin Tools.

Configuring content replication

You can configure Alfresco to use content replication.

Alternatively, you can enable or disable content replication from the Admin Console. See [Admin Console: Working with the replication service](#) for more information.

1. Open the `alfresco-global.properties` file.
2. Set the `replication.enabled` property to `true`.

```
replication.enabled=true
```



If this line is not present in the `alfresco-global.properties` file or the value is set to `false`, you will not be able to run any replication job. As a result, you will see a similar error message in Share:

Status Refresh

⚠ Last job failed. Job started: Sun 12 Feb 2012 14:36:25 Ended: Sun 12 Feb 2012 14:36:25

⚠ 01120127 Unable to replicate. The replication service is not enabled

3. Save the file.
4. Restart the Alfresco server.

Creating a new transfer target for content replication

The **Transfer Target Groups** space contains the transfer target definitions that specify where transfers go to. There is a group level below the folder which is used to classify different sets of transfer targets. There is only a single group called **Default Group**.

You can add transfer targets by creating a new transfers folder.

1. Create a folder in **Company Home > Data Dictionary > Transfers > Transfer Target Groups > Default Group**.
2. A rule defined on the **Default Group** folder specializes the type of any folder created within it.

The type is set to `trx:transferTarget`, which you can then complete through the user interface. The new node contains the properties you can fill in through the user interface to set up your target.

Opening locked content in the source repository

1. Locate the `repositoryId` by browsing to the remote server's CMIS landing page using the following URL:

```
http://{server}:{port}/alfresco/service/cmisis/index.html
```

The `repositoryId` field is displayed in the **CMIS Repository Information** panel.

2. Open the `<web-extension>\share-config-custom.xml` file.
3. Locate the following example configuration:

```
<config evaluator="string-compare" condition="Replication">
  <share-urls>
    To discover a Repository Id, browse to the remote server's
    CMIS landing page at:
    http://{server}:{port}/alfresco/service/cmisis/index.html
    The Repository Id field is found under the "CMIS Repository
    Information" expandable panel.

    Example config entry:
    <share-url repositoryId="622f9533-2a1e-48fe-af4e-
    ee9e41667ea4">http://new-york-office:8080/share/</share-url>
  </share-urls>
</config>
```

4. Modify the `repositoryId` in the following line:

```
<share-url repositoryId="622f9533-2a1e-48fe-af4e-ee9e41667ea4">http://
new-york-office:8080/share/</share-url>
```

5. Copy this configuration setting to your `share-config-custom.xml` file.

Configuring the File System Transfer Receiver

The Transfer Service is accessible as a bean named `TransferService`, and it can be defined, along with other related beans, in the `transfer-service-context.xml` spring context file.

A file system transfer target is marked by specializing a normal transfer target to the type `trx:fileTransferTarget`. It allows you to specify which folder node corresponds to the root folder of the file system receiver by associating the transfer target with a folder (the `trx:fileTransferRootFolder` association).

It supports sync mode transfer, so it can also be used by the replication service. It includes an embedded Derby database to keep track of data (NodeRef to file path mappings, for example), and it runs as a web application in an embedded Tomcat 7 instance using the Web Script Framework and MyBatis.

Setting up the File System Transfer Receiver

The File System Transfer Receiver is delivered as a compressed zip file.

1. Download the following file from the Alfresco Support Portal:
`alfresco-file-transfer-receiver-5.0.zip`
2. Extract the can file into a relevant directory.

The File System Transfer Receiver can file extracts into the following directory structure:

```
classes
lib
webapps
file-transfer-receiver.jar
```

The following files are contained within the subdirectories.

`/classes`

```
ftr-custom-context.xml
ftr-custom.properties
ftr-launcher-context.xml
ftr-launcher.properties
log4j.properties
```

`/lib`

various library files

`/webapps`

```
file-transfer-receiver.war
```

Start File System Transfer Receiver

This section describes how to start the File System Transfer Receiver.

1. Ensure that you have expanded the File System Transfer Receiver zip file:
`alfresco-file-transfer-receiver-5.0.zip`
2. To run the File System Transfer Receiver, enter the following command:

```
java -jar file-transfer-receiver.jar
```

You see information messages to indicate that the Tomcat web application server is starting.

File System Transfer Receiver launch properties

The launch properties for the File System Transfer Receiver are available in the `ftr-launcher.properties` file.

Property	Description
<code>ftr.tomcat.baseDir=</code>	Specifies the base directory in which the embedded Tomcat web application server is installed. This can either be an absolute path or a path relative to where the server is being started from. The default value of <code>\${user.dir}</code> means that the Tomcat base directory is taken to be the user's current working directory.
<code>ftr.tomcat.portNum=</code>	Specifies the port number on which the FSTR Tomcat web application server is to listen. The default is 9090.

File System Transfer Receiver custom properties

The custom properties for the File System Transfer Receiver are available in the `ftr-custom.properties` file.

Property	Description
<code>fileTransferReceiver.stagingDirectory=</code>	The staging directory is where the FSTR will temporarily store the files that it receives from the source repository during a transfer. These files include the manifest file that describes the metadata of the nodes being transferred as well as the actual content files associated with those nodes. All of these files are staged in the directory referenced by this property prior to being moved to their correct location below the root directory. The default is <code>./ftr-staging</code>
<code>fileTransferReceiver.rootDirectory=</code>	Specifies the location of the directory on the local file system that is the top level of the transferred tree of nodes. A node that is a child of the nominated root node of the transfer in the source repository will be placed in the directory referenced by this property when it's transferred. The default is <code>./ftr-root</code>
<code>fileTransferReceiver.jdbcUrl=jdbc:derby:./derbyDB;create=true;user=alfresco;password=alfresco</code>	The FSTR contains an embedded Apache Derby database that it uses to keep track of which nodes it receives and which file on the file system corresponds to which node. This property specifies the connection URL for this embedded database. It is unlikely that it will need to be changed.  Alfresco recommends that you do not store FSTR database on a network file system location, such as an NFS volume. The database must be on a local disk to ensure data integrity.

Property	Description
<code>fileTransferReceiver.username=</code>	The user name that the source repository will have to declare when initiating a transfer to this FSTR. This property must correspond with the user name property stored on the transfer target in the source repository. The default is set to <code>admin</code> .
<code>fileTransferReceiver.password=</code>	The password that the source repository will have to declare when initiating a transfer to this FSTR. This property must correspond with the password property stored on the transfer target in the source repository. The default is set to <code>admin</code> .

File System Transfer Receiver log file properties

You can debug the File System Transfer Receiver issues using log4jproperties.

For example:

```
log4j.logger.org.alfresco.repo.transfer.fsr=warn
log4j.logger.org.alfresco.repo.web.scripts.transfer=warn
```

Monitoring Alfresco

This section describes the various methods for monitoring Alfresco.

JMX monitoring and management extensions

This section describes the JMX-based monitoring and management functionality.

The monitoring and management extensions can be subdivided into three categories:

Read-only monitoring beans

Expose a variety of real-time metrics for monitoring health and throughput of your Alfresco server.

Configuration beans

Provide an easily navigable view of key system configuration for support and diagnostic purposes.

Management beans

Allow control over various subsystems.

For more information on these categories of bean, refer to the reference section [JMX bean categories](#).

Coexistence with other MBeans

If there is an MBean server already running on the Java Virtual Machine (JVM) that Alfresco is running on, Alfresco will export its MBeans to that server. Otherwise, Alfresco will start up its own MBean server. This means that, for example, on Tomcat, the Alfresco beans will complement those provided by the application server and will be navigable in the same context with a suitable JMX client.

Activating the Sun JMX agent and local JMX connectivity

When using Tomcat and a Sun JVM together for monitoring, you can configure Alfresco and Tomcat to share the JVM's own platform MBean server. The pre-registered MBeans give a detailed view of the JVM's health, usage and throughput; in areas including class loading, hot spot compilation, garbage collection, and thread activity.

Sun's MBean server also provides a convenient local connection method, allowing the Alfresco process to be automatically 'discovered' by a JMX client such as JConsole without manual configuration of connection details.

The Sun JMX agent can also be activated in remote mode (where a connection is made through an RMI lookup). However, since Alfresco is always preconfigured to allow a secure remote JMX connection on any JVM, it is most likely that you will choose to activate the Sun JMX agent in local mode. This will mean the platform MBean Server will be shared by Alfresco and still be available for remote connections through the RMI connector.

- To activate the Sun JMX agent in local mode, ensure that the following system property is set:

```
com.sun.management.jmxremote
```

For example, in your Tomcat startup script, you could use the following line:

```
export JAVA_OPTS="${JAVA_OPTS} -Dcom.sun.management.jmxremote"
```

- Refer to the Sun documentation for more information on all the possible configuration options.

Scheduled jobs

Alfresco runs a number of scheduled jobs that assist in the maintenance of a production environment.

The scheduled jobs are defined in the [scheduled-jobs-context.xml](#) file.

Scheduled job	Description
<code>contentStoreCleanerTrigger</code>	Launches the <code>contentStoreCleaner</code> bean, which identifies, and deletes or purges orphaned content from the content store while the system is running. Content is said to be orphaned when all references to a content binary have been removed from the metadata. By default, this job is triggered at 4:00 am each day. In a clustered environment, this job could be enabled on a headless (non-public) node only, which will improve efficiently.
<code>nodeServiceCleanupTrigger</code>	Performs cleanup operations on DM node data, including old deleted nodes and old transactions. In a clustered environment, this job could be enabled on a headless (non-public) node only, which will improve efficiently.
<code>tempFileCleanerTrigger</code>	Cleans up all Alfresco temporary files that are older than the given number of hours. Subdirectories are also emptied and all directories below the primary temporary subdirectory are removed. The job data must include the <code>protectHours</code> property, which is the number of hours to protect a temporary file from deletion since its last modification.

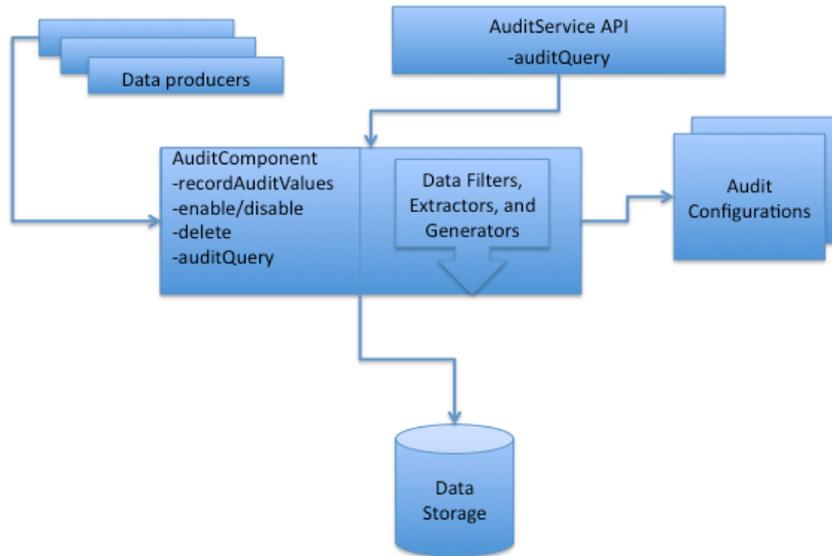
Auditing Alfresco

Alfresco provides the ability to audit activity. This section describes how Alfresco generates, stores, and retrieves auditing information.

-  The auditing mechanism prior to Version 3.4.0 has been removed but the old tables remain in the system. You can access the previous audit data but any new audit data will be

directed to the new tables. Any customizations of the auditing feature must be rewritten using the new configuration files. All SQL-based queries used previously must be replaced by calls to the supplied APIs. The use of low-level SQL statements to retrieve data is not supported.

The architecture of the auditing features comprises the following components:



A data producer defines the components that produce data that might be audited. Data producers do not need to know anything about how the data is stored. Data is generated and sent to the `AuditComponent.recordAuditValues` component. The only requirement is that each packet of data is a *Map* of data keyed by logical path names, which are specific to the producers.

The **AuditService** search should be used for data retrieval; however, for completeness, the following tables are used:

- Tables exclusive to the new audit (*AlfrescoPostCreate-3.2-AuditTables.sql*)
 - `alf_audit_model`: Contains the record of the audit configuration files.
 - `alf_audit_application`: Contains an entry for each logical application. There can be several audit applications defined in a single audit model.
 - `alf_audit_entry`: Contains an entry for each call to `AuditComponent.recordAuditValues`. There is a reference to a property.
- Shared tables (*AlfrescoPostCreate-3.2-PropertyValueTables.sql*)
 - `alf_prop_root`: Entry point for properties: shared values are fully indexed; arbitrarily-deep collections; quick data query and reconstruction.

Audit configuration and environment

Configuration and environment	Details
Tomcat environment	<ul style="list-style-type: none"> Set the configuration properties in the <code>alfresco-global.properties</code> file. Log4J settings can be added in a file <code><tomcat>/shared/classes/alfresco/extension/audit-log4j.properties</code>.
View the available web scripts and details	Use the following scripts: <ul style="list-style-type: none"> Script index: http://localhost:8080/alfresco/service/ Audit scripts: http://localhost:8080/alfresco/service/index/package/org/alfresco/repository/audit
HTTP client	<ul style="list-style-type: none"> <code>curl</code> will be used as the HTTP client
Sample files	<ul style="list-style-type: none"> Audit sample files are distributed in the <code><extension>/audit</code> directory. Activate the sample files by removing the <code>.sample</code> extension.

Check the state of auditing on the server:

```
% curl -u admin:admin "http://localhost:8080/alfresco/service/api/audit/control"
{
  "enabled" : false,
  "applications":
  [
  ]
}
```

Audit filters

This section describes how to use Alfresco global properties to **filter audit data** generated by any audit data producer.

Audit data producers call `AuditComponent.recordAuditValues(rootPath, auditMap)` once for each event to be audited. Filters are applied to reject events so that their values are never used by **audit configurations**. The `rootPath` identifies the data producer and the `auditMap` is the event data. The `rootPath` value and keys in the map represent a tree structure.

Example rootPath and auditMap

The last component in the `rootPath` is considered by the `AuditFilter` to be the event **action**. The keys in an audit map identify each audit value. Global properties can be defined to accept or reject each value. If any value in an audit map is rejected, the whole map is rejected. So that one does not have to define too many properties, a *default* event action property can be defined. This will be inherited by all actions unless a property is defined for a particular event action.

```
rootPath:
  /alfresco-access/transaction

auditMap:
  "action"          => "MOVE"
  "node"            => "workspace://SpacesStore/90a398d1-8e0d-462a-8c3b-f0b17a2d1143"
  "move/from/node" => "workspace://SpacesStore/a82446e9-4dca-49d2-9ce0-4526687fb310"
  "move/from/path" => "/app:company_home/st:sites/cm:fred/cm:documentLibrary/cm:folder1"
  "move/from/type" => "cm:folder"
```

```

    "move/to/node" => "workspace://
SpacesStore/517bd4d0-99bc-47ad-8cd7-5d425f94c7db"
    "move/to/path" => "/app:company_home/st:sites/cm:fred/cm:documentLibrary"
    "move/to/type" => "cm:folder"
    "path" => "/app:company_home/st:sites/cm:fred/cm:documentLibrary/
cm:Word 123.docx"
    "sub-actions" => "moveNode readContent"
    "type" => "cm:content"
    "user" => "admin"

```

Example filter

Each property value defines a list of regular expressions that will be used to match the actual audit map values.

```

audit.filter.alfresco-access.default.enabled=true
audit.filter.alfresco-access.default.user=~System;.*
audit.filter.alfresco-access.default.type=cm:folder;cm:content
audit.filter.alfresco-access.default.path=/app:company_home/. *
audit.filter.alfresco-access.transaction.user=
audit.filter.alfresco-access.login.user=jblogs
...

```

In the example, events created by any user except for the internal user *"System"* will be recorded by default for all event actions. However the property for the *transaction* event action overrides this to record even *"System"* events.

For any filters to be applied to an event action, that action's filters must be enabled with an *"enabled"* property set to *"true"*. However this can also be done by using the *default* event action, as shown in the example. Property names have a *"audit.filter."* prefix and use *'.'* as a separator where as components of *rootPath* and keys in the audit map use *'/'*.

Lists are evaluated from left to right allowing one flexibility to accept or reject different combinations of values. If no match is made by the end of the list the value is rejected. If there is not a property for a given value or an empty list is defined (as specified for the *"user"* value on a *"transaction"* action) any value is accepted. Each regular expression in the list is separated by a semicolon (*'.'*). Expressions that include a semicolon can be escaped using a *'\'*. An expression that starts with a *'~'* indicates that any matching value should be rejected. If the first character of an expression needs to be a *'~'*, it too can be escaped with a *'\'*.

A property value can be a reference to another property, which saves having multiple copies of the same regular expression. This is indicated by a *'\$'* as the first character of the property value. If the first character of an expression needs to be a *'\$'* it too can be escaped with a *'\'*.

Redirected properties

```

audit.filter.alfresco-access.transaction.type=$transaction.content.types
transaction.content.types=$general.content.types
general.content.types=cm:folder;cm:content

```

Debug information

The *PropertyAuditFilter* provides *log4j* debug information (in the *alfresco.log* file) when it rejects values. Turning on this debugger can generate large volumes of output.

Enable debug

```

# Change file appender to include debug from any source
log4j.appender.File.Threshold=debug

# Enable debug from the PropertyAuditFilter
log4j.logger.org.alfresco.repo.audit.PropertyAuditFilter=debug

```

Audit filter customizations

You can define additional filter properties and override predefined filter values.

If you are using the Tomcat web application server, add the additional properties to the `<tomcat>/shared/classes/alfresco-global.properties` file.

Content auditing

This section describes how to use Alfresco to audit actions performed on your content and folders, including a technical overview, and also examples of how to customize the standard configuration.

Content auditing technical overview

The data producer `org.alfresco.repo.audit.access.AccessAuditor` gathers together lower events into user recognizable events. For example, the download or preview of content is recorded as a single read. Similarly the upload of a new version of a document is recorded as a single create version. By contrast the `AuditMethodInterceptor` data producer typically would record multiple events.

A default audit configuration file located at `<alfresco.war>/WEB-INF/classes/alfresco/audit/alfresco-audit-access.xml` is provided that persists audit data for general use. This can be enhanced to extract additional data of interest to specific installations. For ease of use, login success, login failure and logout events are also persisted by the default configuration.

Default audit filter settings are also provided for the `AccessAuditor` data producer, so that internal events are not reported. These settings can be customized (by setting global properties) to include or exclude auditing of specific areas of the repository, users or some other value included in the audit data created by `AccessAuditor`.

No additional functionality is provided for the retrieval of persisted audit data, as all data is stored in the standard way, and so is accessible using the `AuditService` search, audit web scripts, and database queries.

Example audit trail

User actions from Share:

1. Create a new folder called **My Documents**.
2. Upload a document (`The fox.odt`).
3. Preview the document.
4. Update the meta data.
5. Upload a new version.
6. Copy the document to a folder called **MyPictures**.
7. Delete the copy of the document.

In the example, the property values show "..." to indicate that they are truncated.

```
1. /alfresco-access/transaction/action=CREATE
   /alfresco-access/transaction/aspects/add=[cm:titled]
   /alfresco-access/transaction/path=/app:company_home/st:sites/
cm:mysite/cm:documentLibrary/cm:My Documents
   /alfresco-access/transaction/properties/add=...
   /alfresco-access/transaction/sub-actions=createNode
updateNodeProperties addNodeAspect
   /alfresco-access/transaction/type=cm:folder
   /alfresco-access/transaction/user=admin

2. /alfresco-access/transaction/action=CREATE
```

```

    /alfresco-access/transaction/aspects/add=[cm:titled, cm:author]
    /alfresco-access/transaction/path=/app:company_home/st:sites/
cm:mysite/cm:documentLibrary/cm:My Documents/cm:The fox.odt
    /alfresco-access/transaction/properties/add=...
    /alfresco-access/transaction/sub-actions=createNode
updateNodeProperties readContent createContent updateContent
addNodeAspect
    /alfresco-access/transaction/type=cm:content
    /alfresco-access/transaction/user=admin

3. /alfresco-access/transaction/action=READ
    /alfresco-access/transaction/path=/app:company_home/st:sites/
cm:mysite/cm:documentLibrary/cm:My Documents/cm:The fox.odt
    /alfresco-access/transaction/sub-actions=readContent
    /alfresco-access/transaction/type=cm:content
    /alfresco-access/transaction/user=admin

4. /alfresco-access/transaction/action=updateNodeProperties
    /alfresco-access/transaction/aspects/add=[cm:taggable]
    /alfresco-access/transaction/path=/app:company_home/st:sites/
cm:mysite/cm:documentLibrary/cm:My Documents/cm:The fox.odt
    /alfresco-access/transaction/properties/add=...
    /alfresco-access/transaction/properties/from={cm:modified=Mon Jun 13
15:34:05 BST 2011}
    /alfresco-access/transaction/properties/to={cm:modified=Mon Jun 13
15:39:35 BST 2011}
    /alfresco-access/transaction/sub-actions=updateNodeProperties
addNodeAspect readContent
    /alfresco-access/transaction/type=cm:content
    /alfresco-access/transaction/user=admin

5. /alfresco-access/transaction/action=CHECK IN
    /alfresco-access/transaction/aspects/add=[cm:versionable]
    /alfresco-access/transaction/copy/from/path=/app:company_home/
st:sites/cm:mysite/cm:documentLibrary/cm:My Documents/cm:The fox
(Working Copy).odt
    /alfresco-access/transaction/path=/app:company_home/st:sites/
cm:mysite/cm:documentLibrary/cm:My Documents/cm:The fox.odt
    /alfresco-access/transaction/properties/add=...
    /alfresco-access/transaction/properties/from=...
    /alfresco-access/transaction/properties/to=...
    /alfresco-access/transaction/sub-actions=updateNodeProperties
addNodeAspect createVersion readContent deleteNodeAspect updateContent
copyNode checkIn
    /alfresco-access/transaction/type=cm:content
    /alfresco-access/transaction/user=admin
    /alfresco-access/transaction/version=2.0

6. /alfresco-access/transaction/action=COPY
    /alfresco-access/transaction/aspects/add=[cm:titled, cm:copiedfrom,
cm:author, cm:taggable]
    /alfresco-access/transaction/copy/from/path=/app:company_home/
st:sites/cm:mysite/cm:documentLibrary/cm:My Documents/cm:The fox.odt
    /alfresco-access/transaction/path=/app:company_home/st:sites/
cm:mysite/cm:documentLibrary/cm:My Pictures/cm:The fox.odt
    /alfresco-access/transaction/properties/add=...
    /alfresco-access/transaction/sub-actions=createNode readContent
createContent updateNodeProperties addNodeAspect copyNode
    /alfresco-access/transaction/type=cm:content
    /alfresco-access/transaction/user=admin

7. /alfresco-access/transaction/action=DELETE
    /alfresco-access/transaction/path=/app:company_home/st:sites/
cm:mysite/cm:documentLibrary/cm:My Pictures/cm:The fox.odt
    /alfresco-access/transaction/sub-actions=deleteNode
    /alfresco-access/transaction/type=cm:content
    /alfresco-access/transaction/user=admin

```

Enabling auditing of content

1. Open the `alfresco-global.properties` file.
2. Add the following properties:

```
# Enable audit in general
audit.enabled=true

# Enable the alfresco-access audit application
audit.alfresco-access.enabled=true

# Enable the auditing of sub-actions. Normally disabled as these values
# are
# not normally needed by audit configurations, but may be useful to
# developers
#audit.alfresco-access.sub-actions.enabled=true
```

Default audit filter settings

These values result in events only being recorded for common actions initiated by users of the system. These values can be overridden if required.

```
audit.filter.alfresco-access.default.enabled=true
audit.filter.alfresco-access.transaction.user=~System;~null;.*
audit.filter.alfresco-access.transaction.type=cm:folder;cm:content;st:site
audit.filter.alfresco-access.transaction.path=~/*sys:archivedItem;~/ver:;.*
```

Audit data generated by AccessAuditor

The `/sub-action/<sequence>` structure holds cut down details of each sub-action, but are only included if the global property `audit.alfresco-access.sub-actions.enabled=true`.

The structure of the audit data is shown as follows:

```
/alfresco-access
  /transaction
    /action=<actionNamegt
    /sub-actions=<sub action listgt
    /path=<prefixPathgt
    /type=<prefixTypegt
    /node=<nodeRefgt
    /user=<usergt
    /copy
      /from
        /node=<nodeRefgt
        /path=<prefixPathgt
        /type=<prefixTypegt
    /move
      /from
        /node=<nodeRefgt
        /path=<prefixPathgt
        /type=<prefixTypegt
    /properties
      /from=<mapOfValuesgt
        /<propertyNamegt=<propertyValuegt
        ...
      /to=<mapOfValuesgt
        /<propertyNamegt=<propertyValuegt
        ...
      /add=<mapOfValuesgt
        /<propertyNamegt=<propertyValuegt
        ...
      /delete=<mapOfValuesgt
        /<propertyNamegt=<propertyValuegt
        ...
    /aspects
      /add=<mapOfNamesgt
        /<aspectNamegt=null
```

```

    ...
    /delete=<mapOfNamesgt
    /<aspectNamegt=null
    ...
  /version-properties=<mapOfValuesgt
  /sub-action/<sequencegt
  /action=<actionNamegt
  /copy
    ...
  /move
    ...
  /properties
    ...
  /aspects
    ...

```

An example of audit data is shown as follows:

Inbound audit values:

```

  /alfresco-access/transaction/action=MOVE
  /alfresco-access/transaction/node=workspace://
SpacesStore/74a5985a-45dd-4698-82db-8eaeff9df8d7
  /alfresco-access/transaction/move/from/node=workspace://SpacesStore/
d8a0dfd8-fe45-47da-acc2-fd8df9ea2b2e
  /alfresco-access/transaction/move/from/path=/app:company_home/st:sites/
cm:abc/cm:documentLibrary/cm:folder1/cm:Word 123.docx
  /alfresco-access/transaction/move/from/type=cm:folder
  /alfresco-access/transaction/path=/app:company_home/st:sites/cm:abc/
cm:documentLibrary/cm:folder2/cm:Word 123.docx
  /alfresco-access/transaction/sub-actions=moveNode readContent
  /alfresco-access/transaction/type=cm:content
  /alfresco-access/transaction/user=admin
  /alfresco-access/transaction/sub-action/00/action=moveNode
  /alfresco-access/transaction/sub-action/00/move/from/node=workspace://
SpacesStore/d8a0dfd8-fe45-47da-acc2-fd8df9ea2b2e
  /alfresco-access/transaction/sub-action/00/move/from/path=/
app:company_home/st:sites/cm:abc/cm:documentLibrary/cm:folder1/cm:folder1/
cm:Word 123.docx
  /alfresco-access/transaction/sub-action/00/move/from/type=cm:folder
  /alfresco-access/transaction/sub-action/01/action=readContent

```

Persisted audit data

The default structure of the persisted audit data is shown as follows:

```

/alfresco-access
  /login/user=<usergt
  /loginFailure/user=<usergt
  /logout/user=<usergt
  /transaction/
  /action=<actionNamegt
  /sub-actions=<sub action listgt
  /path=<prefixPathgt
  /type=<prefixTypegt
  /user=<usergt
  /version=<versiongt
  /copy/from/path=<prefixPathgt
  /move
    /from/path=<prefixPathgt
  /properties
    /from=<mapOfValuesgt
    /to=<mapOfValuesgt
    /add=<mapOfValuesgt
    /delete=<mapOfValuesgt
    /fromName=<oldNamegt
    /toName=<newNamegt
  /aspects
    /add=<mapOfNamesgt
    /delete=<mapOfNamesgt

```

The `version` value is sourced from either the `add/cm:versionLabel` or `to/cm:versionLabel` values.

The exception is the property `name`, individual property and aspect changes are not included, as it is not possible to know all possible names. The map values of all changes is however included. The individual property `name` value is included as it is a well known property, which changes if content or a folder is renamed within the same parent folder.

Content auditing customizations

There are two customizations available: the custom audit filter, and the custom configuration.

Custom audit filter

The most common customization is to change the default audit filter values.

These filter values are used to include or exclude selected events. Global property names identifies elements in the generated audit data. Each property value is a list of regular expressions that either accept or reject the generated data value. If any value is rejected in a set of data the whole set is rejected. For example, to audit the users "jblogs" and any user that starts with "temp" other than "tempmanager", override the following global property value. If using tomcat, add a value to the `<tomcat>/shared/classes/alfresco-global.properties` file.

The following is an example custom filter:

```
audit.filter.alfresco-access.transaction.user=~tempManager;test.*;jblogs
```

The list is semicolon separated. Any regular expression that starts with a '~' indicates that a matching value should be rejected. The list is evaluated from left to right until there is a match. If no match is made the value is rejected. If the list is empty (zero length) all values are accepted. It is possible to filter on any of the generated data values. Refer to the audit filtering section for a more detailed description of filter properties.

Custom audit configuration

The most common reason to customize the audit configuration is if there is a need to extract individual property or aspect values that have special meaning to a particular Alfresco installation.

For example, a security clearance level has been added to content and it is important to include that clearly in the persisted audit data, rather than having to find it deep within a map of all properties. The default configuration includes an example. It extracts the `name` property. It is generally a good idea to create a new audit configuration file that includes a mapped path to avoid confusion with the default. If running under Tomcat place the audit configuration file in the `<tomcat>/shared/classes/alfresco/extension/audit` directory. The following example is simply a cut down version of the default with the path mapped to a new value.

The following is an example of the `myApp.xml` file.

```
<?xml version="1.0" encoding="UTF-8"?>
<Audit xmlns="http://www.alfresco.org/repo/audit/model/3.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.alfresco.org/repo/audit/model/3.2
  alfresco-audit-3.2.xsd">

  <DataExtractors>
    <DataExtractor name="simpleValue"
      registeredName="auditModel.extractor.simpleValue"/>
  </DataExtractors>

  <PathMappings>
    <PathMap source="/alfresco-access" target="/my-app" />
  </PathMappings>
</Audit>
```

```

</PathMappings>

<Application name="my-app" key="my-app">
  <RecordValue
    key="action" dataExtractor="simpleValue"
    dataSource="/my-app/transaction/action"
    dataTrigger="/my-app/transaction/action" />
  <RecordValue
    key="user" dataExtractor="simpleValue"
    dataSource="/my-app/transaction/user"
    dataTrigger="/my-app/transaction/user" />
  <RecordValue
    key="path" dataExtractor="simpleValue"
    dataSource="/my-app/transaction/path"
    dataTrigger="/my-app/transaction/path" />
</Application>

</Audit>

```

The following shows the AccessAuditor debug for a move action.

```

Audit data:
  /my-app/action=MOVE
  /my-app/path=/app:company_home/st:sites/cm:fred/cm:documentLibrary/cm:Word
  123.docx
  /my-app/user=admin

Inbound audit values:
  /alfresco-access/transaction/action=MOVE
  /alfresco-access/transaction/node=workspace://
  SpacesStore/90a398d1-8e0d-462a-8c3b-f0b17a2d1143
  /alfresco-access/transaction/move/from/node=workspace://SpacesStore/
  a82446e9-4dca-49d2-9ce0-4526687fb310
  /alfresco-access/transaction/move/from/path=/app:company_home/st:sites/
  cm:fred/cm:documentLibrary/cm:folder1/cm:Word 123.docx
  /alfresco-access/transaction/move/from/type=cm:folder
  /alfresco-access/transaction/path=/app:company_home/st:sites/cm:fred/
  cm:documentLibrary/cm:Word 123.docx
  /alfresco-access/transaction/sub-action/00/action=moveNode
  /alfresco-access/transaction/sub-action/00/move/from/node=workspace://
  SpacesStore/a82446e9-4dca-49d2-9ce0-4526687fb310
  /alfresco-access/transaction/sub-action/00/move/from/path=/
  app:company_home/st:sites/cm:fred/cm:documentLibrary/cm:folder1/cm:Word
  123.docx
  /alfresco-access/transaction/sub-action/00/move/from/type=cm:folder
  /alfresco-access/transaction/sub-action/01/action=readContent
  /alfresco-access/transaction/sub-actions=moveNode readContent
  /alfresco-access/transaction/type=cm:content
  /alfresco-access/transaction/user=admin

```

Audit sample files

Samples can also be downloaded directly from the following location in svn:

```
http://svn.alfresco.com/repos/alfresco-open-mirror/alfresco/HEAD/root/projects/
repository/config/alfresco/extension/audit/
```

When using a sample file, remove the `.sample` extension.

Enabling auditing

Generation of audit data is disabled by default. To enable auditing permanently, settings must be added to the Alfresco global properties file as shown in the following text.

To enable auditing permanently, add the following settings to the `alfresco-global.properties` file:

```
audit.alfresco-access.enabled=true
```



Auditing is enabled by default. The `audit.enabled` property provides a way to globally enable or disable the auditing framework. However, enabling this property does not necessarily result in the generation of audit data.

To enable generation of audit data that you can view in Share, you will need to enable the `audit.alfresco-access.enabled` property.

Once changes to the global properties file have been saved, you will need to restart the Alfresco server, for auditing to be fully enabled.

You can check the status of auditing conveniently from the command line by using a tool such as `curl` to access the Audit Control web script.

To check the global status of auditing issue a command such as:

```
curl -u admin:password "http://localhost:8080/alfresco/service/api/audit/control"
```

This invokes the web script with a GET request. This will result in a JSON response such as the following if auditing is currently enabled:

```
{
  "enabled" : true,
  "applications":
  [
    {
      "name": "Alfresco Tagging Service",
      "path" : "/tagging",
      "enabled" : true
    },
    '
    {
      "name": "alfresco-access",
      "path" : "/alfresco-access",
      "enabled" : true
    }
  ]
}
```

While this does return the global status of the auditing framework, audit data will only be generated if `audit.alfresco-access.enabled` is enabled.

If auditing is currently disabled the response will be:

```
{
  "enabled" : false
}
```

In this case no audit data will be generated as the audit framework is disabled.

Auditing can also be globally enabled or disabled using the control web script. To do this a POST request is sent to the web script. For example, using `curl`, auditing can be enabled using the following command:

```
curl -u admin:password -d "" "http://localhost:8080/alfresco/service/api/audit/control?enable=true"
```

This results in the following response:

```
{
  "enabled" : true
}
```

}

To disable auditing issue the following command:

```
curl -u admin:password -d "" "http://localhost:8080/alfresco/service/api/audit/control?enable=false"
```

This results in the following response:

```
{
  "enabled" : false
}
```

While the global status of the auditing framework can be switched on and off in this manner, audit data will only be generated if `audit.alfresco-access.enabled` is enabled in the global properties file.



Enabling or disabling auditing using the Audit Control web script only remains valid in force while the server is running; the setting will not be retained following a server restart, but will subsequently be set according to the values in `alfresco-global.properties`.

Using JMX to control auditing

A JMX client can be used to access global properties. The properties can be modified using the JMX client. A server restart will be required for changes to properties to take effect.

Auditing examples

This section describes some auditing examples.

Audit data passed to `recordAuditValues()`:

```
Root path:
  /alfresco-api/post/NodeService/createStore
Map:
  args/protocol = "workspace"
  args/identifier = "SpacesStore"
  result = StoreRef[workspace://SpacesStore]
```

If the root path passes the initial filtration phase - there is at least one component interested in auditing the information - then the map is expanded.

Expanded audit data:

```
Map:
  /alfresco-api/post/NodeService/createStore/args/protocol = "workspace"
  /alfresco-api/post/NodeService/createStore/args/identifier =
  "SpacesStore"
  /alfresco-api/post/NodeService/createStore/result = StoreRef[workspace://
  SpacesStore]
```

The filtered data is then passed through the path mappings, generating a new "Map" of data for each application.

Path-mapped audit data:

```
Map:
  /MyApp/createStore = StoreRef[workspace://SpacesStore]
```

This data is then passed to any extractors and generators to produce a final "Map" of data that will be persisted.

Persisted audit data:

```
Map:
  /MyApp/createStore/value = StoreRef[workspace://SpacesStore]
  /MyApp/createStore/rootNode = NodeRef[workspace://SpacesStore/fd123...]
```

Audit configuration files

This section describes the location and basic structure of the audit configuration files.

The XML schema is located at [alfresco-audit-3.2.xsd](#).

The configuration file structure is divided into four basic sections:

<DataExtractors>

In this section, `DataExtractors` are declared for use in the `<Application>` sections of the configuration files. A `DataExtractor` is a component that uses input data to produce some output, either transforming the data or outputting the data verbatim. The simplest extractor is the `SimpleValueDataExtractor`, which returns whatever data is passed in. A more complex extractor is the `NodeNameDataExtractor`, which is able to produce the `cm:name` value of a node, assuming the data passed in is a `NodeRef`. For the complete set of built-in generators, see the `org.alfresco.repo.audit.extractor` package, or the `auditModel.extractor.*` beans, which are declared in `alfresco/audit-services-context.xml`.

The extractors can be declared in-line, for example:

```
<DataExtractors>
  <DataExtractor name="simpleValue"
class="org.alfresco.repo.audit.extractor.SimpleValueDataExtractor"/>
  ...
</DataExtractors>
```

Or they can be declared in Spring configuration and referenced in the audit configuration (see the `alfresco/audit-services-context.xml` file), for example:

```
<DataExtractors>
  <DataExtractor name="simpleValue"
registeredName="auditModel.extractor.simpleValue"/>
  ...
</DataExtractors>
```

<DataGenerators>

In this section, `DataGenerators` are declared for use in the `<Application>` sections of the configuration files. A `DataGenerator` is a component that produces data without any input, that is, data is produced when a data path is active, but is independent of the values at that path. Examples of generators are the `AuthenticatedUserDataGenerator` component, which produces the name of the currently-authenticated user (user in context) and the `AuthenticatedPersonDataGenerator` component, which produces the full name of the currently-authenticated user (person in context). For the complete set of built-in generators, see the `org.alfresco.repo.audit.generator` package or the `auditModel.generator.*` beans, which are declared in the `alfresco/audit-services-context.xml` file.

The generators can be declared in-line, for example:

```
<DataGenerators>
  <DataGenerator name="currentUser"
class="org.alfresco.repo.audit.generator.AuthenticatedUserDataGenerator"/>
  <DataGenerator name="personFullName"
class="org.alfresco.repo.audit.generator.AuthenticatedPersonDataGenerator"/>
  >
</DataGenerators>
```

Or they can be declared in Spring configuration and referenced in the audit configuration (see the `alfresco/audit-services-context.xml` file), for example:

```
<DataGenerators>
  <DataGenerator name="currentUser"
registeredName="auditModel.generator.user"/>
  <DataGenerator name="personFullName"
registeredName="auditModel.generator.personFullName"/>
</DataGenerators>
```

<PathMappings>

The expanded map coming from the Data Producers is passed through the path mappings. This is a raw remapping of the input data based on the path names in the data map.

```
<PathMappings>
  <PathMap source="/DOD5015" target="/DOD5015"/>
  <!-- Force the fullName generator to trigger -->
  <PathMap source="/DOD5015/event/node" target="/DOD5015/event/
person"/>
  <PathMap source="/alfresco-api/post/AuthenticationService/
authenticate" target="/DOD5015/login"/>
</PathMappings>
```

In this example, all paths starting with `/DOD5015` are mapped verbatim, but without the declaration, the data paths starting with `/DOD5015` are discarded. A small subset of the Alfresco API data is used (only the `AuthenticationService.authenticate` call) by mapping all values starting with that path to `/DOD5015/login`.

<Application>

This section defines how the mapped data is to be used by DataGenerators or by DataExtractors.

```
<Application name="DOD5015" key="DOD5015">
  <AuditPath key="login">
    <AuditPath key="args">
      <AuditPath key="userName">
        <RecordValue key="value" dataExtractor="simpleValue"/>
      </AuditPath>
    </AuditPath>
    <AuditPath key="no-error">
      <GenerateValue key="fullName"
dataGenerator="personFullName"/>
    </AuditPath>
    <AuditPath key="error">
      <RecordValue key="value" dataExtractor="nullValue"/>
    </AuditPath>
  </AuditPath>
</Application>
```

Built-in data producers

The following are built-in data producers.

- `org.alfresco.repo.audit.AuditMethodInterceptor`: Generates audit data for all public service API calls. Refer to the javadocs for the data structure.
- `org.alfresco.repo.node.NodeAuditor`: Generates audit data for `beforeDeleteNode`

It is possible for any server-side component to pass data to the `auditComponent` bean.

To see what information is available to audit, enable the following logging:

```
log4j.logger.org.alfresco.repo.audit.inbound=DEBUG
```

DataExtractors and DataGenerators

This section provides a description of DataExtractors and DataGenerators.

It is possible for any server-side component to pass data to the `auditComponent` bean.

DataExtractor

Uses an inbound mapped value as the source of the data. *AuditExampleLogin1* records values quite literally using the *simpleValue* data extractor.

DataGenerator

Activates when an inbound mapped path is present, but is not dependent on the value on that path. *AuditExampleLogin2* triggers the *personFullName* generator when the *authenticate/no-error* path is present; this records the full name of the currently-authenticated user even though the inbound data for *authenticate/no-error* is *null*.

Look at the data recorded for the two sample applications:

```
% curl -u admin:admin "http://localhost:8080/alfresco/service/api/audit/query/AuditExampleLogin1?verbose=true&forward=false&limit=1"
{
  "count":1,
  "entries":
  [
    {
      "id":137,
      "application":AuditExampleLogin1,
      "user":admin,
      "time":"2010-09-20T17:37:14.699+01:00",
      "values":
      {
        "\/auditexamplelogin1\/login\/no-error\/user":"admin"
      }
    }
  ]
}
% curl -u admin:admin "http://localhost:8080/alfresco/service/api/audit/query/AuditExampleLogin2?verbose=true&forward=false&limit=1"
{
  "count":1,
  "entries":
  [
    {
      "id":138,
      "application":AuditExampleLogin2,
      "user":admin,
      "time":"2010-09-20T17:37:23.101+01:00",
      "values":
      {
        "\/auditexamplelogin2\/login\/user":"Administrator"
      }
    }
  ]
}
```

Locating the audit code

This section describes the location of audit code.

1. For DEBUG logging, to see which data is being produced, rejected, or recorded, enable DEBUG for:

```
log4j.logger.org.alfresco.repo.audit.AuditComponentImpl=DEBUG
```

2. For JUnit code, the unit test code demonstrates use of the Audit APIs and configuration:

```
org.alfresco.repo.audit.AuditComponentTest
```

- `alfresco-audit-test-authenticationservice.xml`: This is used by the test to capture both successful and unsuccessful login attempts in the audit data.
- `testAuditAuthenticationService`: This demonstrates the use of the `auditSearch` method.

- For Records Management (DOD5015) and auditing, the module pulls in audit data from the 'AuthenticationService' but adds more data around the individual actions that take place during Records Management processes.

```
org.alfresco.module.org_alfresco_module_dod5015.audit.*
```

- `RecordsManagementAuditServiceImpl$RMAuditTxnListener`: This transaction listener generates Records Management-specific data for events (it is a `Data Producer`). It generates node property deltas.
- `config/alfresco/module/org_alfresco_module_dod5015/audit/rm-audit.xml`: This defines how the data produced by the `AuthenticationService` and the Records Management module is persisted. There are some custom `DataGeneratorS` and `DataRecorderS`.
- `RecordsManagementAuditServiceImpl.getAuditTrailImpl`: This method demonstrates how the Records Management use-case searches the audit data. Further query extensions are required to extend the search filters available using the `auditQuery` API.

Defining the audit application

This section describes the audit applications.

Data producers have no knowledge of how or whether data will be stored. Different use cases need to store or modify inbound data independently, therefore the use cases are separated into audit applications. Each application defines how data is mapped, extracted, and recorded without affecting data required by other applications.

For example, the Records Management module records before and after values when specific nodes are modified, whereas the CMIS standard requires a slightly different set of data to be recorded. Additionally, each of the audit logs can be enabled and disabled independently within the same server. Usually, each audit application is defined in its own configuration file, but for demonstration purposes, multiple application definitions can be defined in one configuration file.

- Enable the sample file by removing the `.sample` extension.
`alfresco/extensions/audit/alfresco-audit-example-login.xml.sample`
- Restart the Alfresco server.
- Ensure that the applications have been registered properly and are enabled:

```
% curl -u admin:admin "http://localhost:8080/alfresco/service/api/audit/control"
{
  "enabled" : true,
  "applications":
  [
    {
      "name": "AuditExampleLogin1",
      "path" : "/auditexamplelogin1",
      "enabled" : true
    },
    {
      "name": "AuditExampleLogin2",
      "path" : "/auditexamplelogin2",
      "enabled" : true
    },
    {
      "name": "CMISChangeLog",
      "path" : "/CMISChangeLog",
      "enabled" : true
    }
  ]
}
```

```
]
}
```

4. At an application level, auditing is enabled or disabled for specific paths; changes made to an application's audit state are persisted. To disable all auditing for an application, disable the root path; in this case, disable the root path for the `CMISChangeLog` application. If you restart the server you will see that the application remains disabled.

```
% curl -u admin:admin -d "" "http://localhost:8080/alfresco/service/api/audit/control/CMISChangeLog/CMISChangeLog?enable=false"
{
  "enabled" : false
}
```

Simple audit query

You can use or edit this simple audit query example.

1. Generate some auditing data for the sample applications.
2. Connect to Alfresco One.
3. Log in as the `admin` user.
4. Log out from Alfresco One.
5. Log in as the `admin` user but use an illegal password.

The following examples are two queries to return results: without and with full-audited values respectively. Some entries have been replaced with a (...) for brevity.

```
% curl -u admin:admin "http://localhost:8080/alfresco/service/api/audit/query/AuditExampleLogin1"
{
  "count":4,
  "entries":
  [
    {
      "id":69,
      "application":AuditExampleLogin1,
      "user":admin,
      "time":"2010-09-20T14:45:28.998+01:00",
      "values":
null
    },
    ...
  ]
}
% curl -u admin:admin "http://localhost:8080/alfresco/service/api/audit/query/AuditExampleLogin1?verbose=true"
{
  "count":5,
  "entries":
  [
    ...
    {
      "id":72,
      "application":AuditExampleLogin1,
      "user":null,
      "time":"2010-09-20T14:45:43.884+01:00",
      "values":
      {
        "\/auditexamplelogin1\/login\/error\/user":"admin"
      }
    },
    ...
    {
      "id":76,
      "application":AuditExampleLogin1,
```

```

        "user":admin,
        "time":"2010-09-20T14:46:23.319+01:00",
        "values":
        {
            "\/auditexamplelogin1\/login\/no-error\/
user":"admin"
        }
    }
]
}

```

There is no count function in the search API. This is by design; use the `limit` parameter instead.

6. Assume that a client wants to see the details of the latest two results but knows of the existence of the next eight results. In this case, it would be pointless pulling back full (`verbose=true`) results for the latest 10 entries. Instead, pull back the last two results with values and then pull back the next eight results without values.

Notice that the response contains a count of the number of entries returned; the individual entries are provided so that the entry IDs can be used for further result retrieval.

```

% curl -u admin:admin "http://localhost:8080/alfresco/service/api/audit/
query/AuditExampleLogin1?verbose=true&limit=2&forward=false"
{
  "count":2,
  "entries":
  [
    {
      "id":98,
      "application":AuditExampleLogin1,
      "user":admin,
      "time":"2010-09-20T15:10:04.043+01:00",
      "values":
      {
        "\/auditexamplelogin1\/login\/no-error\/
user":"admin"
      }
    },
    {
      "id":96,
      "application":AuditExampleLogin1,
      "user":admin,
      "time":"2010-09-20T15:09:50.117+01:00",
      "values":
      {
        "\/auditexamplelogin1\/login\/no-error\/
user":"admin"
      }
    }
  ]
}
% curl -u admin:admin "http://localhost:8080/alfresco/service/api/audit/
query/AuditExampleLogin1?verbose=false&limit=8&forward=false&toId=96"
{
  "count":8,
  "entries":
  [
    {
      "id":94,
      "application":AuditExampleLogin1,
      "user":admin,
      "time":"2010-09-20T15:09:47.606+01:00",
      "values":

```

```

null
  },
  ...
  {
    "id":80,
    "application":AuditExampleLogin1,
    "user":admin,
    "time":"2010-09-20T14:58:34.305+01:00",
    "values":
null
  }
]
}

```

Advanced audit query

You can use or edit this advanced audit query example.

This type of query URL makes use of a data path within the audit application. This allows entries to be found that match specific audited values. By default, query values are treated as case-sensitive string types, but it is possible to specify the type to query against.

1. Generate some audit data.
2. Connect to Alfresco One.
3. Attempt a failed login as joe.

```

% curl -u admin:admin "http://localhost:8080/alfresco/service/api/audit/query/AuditExampleLogin1/auditexamplelogin1/login/error/user?verbose=true&value=joe"
{
  "count":1,
  "entries":
  [
    {
      "id":101,
      "application":AuditExampleLogin1,
      "user":null,
      "time":"2010-09-20T15:13:57.947+01:00",
      "values":
      {
        "\/auditexamplelogin1\/login\/error\/user":"joe"
      }
    }
  ]
}
% curl -u admin:admin "http://localhost:8080/alfresco/service/api/audit/query/AuditExampleLogin1/auditexamplelogin1/login/error/user?verbose=true&value=JOE"
{
  "count":0,
  "entries":
  [
  ]
}

```

Understanding PathMappings

To create an audit configuration file, it is necessary to know which data can be audited and how the data is mapped onto your application.

1. Turn on debugging for the inbound data. For a better understanding, you can turn on debug logging for the mapping components as well, although this is more verbose.

```

% cat <tomcatgt/shared/classes/alfresco/extension/audit-log4j.properties
log4j.logger.org.alfresco.repo.audit.AuditComponentImpl=DEBUG
log4j.logger.org.alfresco.repo.audit.inbound=DEBUG

```

2. Tail the log file and watch the output.

a. Log in as admin.

```
16:47:37,434 DEBUG [repo.audit.inbound]
Inbound audit values:
 /alfresco-api/pre/AuthenticationService/authenticate/args/
userName=admin
16:47:37,443 User:admin DEBUG [repo.audit.inbound]
Inbound audit values:
 /alfresco-api/post/AuthenticationService/authenticate/no-error=null
 /alfresco-api/post/AuthenticationService/authenticate/args/
userName=admin
```

b. From the inbound values (and if you have the AuditComponentImpl debugging on):

```
16:47:37,445 User:System DEBUG [repo.audit.AuditComponentImpl]
Extracted audit data:
 Application: AuditApplication[ name=AuditExampleLogin2, id=7,
disabledPathsId=7]
 Raw values: {/auditexamplelogin2/login=null}
 Extracted: {}
16:47:37,447 User:admin DEBUG [repo.audit.AuditComponentImpl] New
audit entry:
 Application ID: 7
 Entry ID: 130
 Values: {/auditexamplelogin2/login=null}
 Audit Data: {/auditexamplelogin2/login/user=Administrator}
16:47:37,447 User:System DEBUG [repo.audit.AuditComponentImpl]
Extracted audit data:
 Application: AuditApplication[ name=AuditExampleLogin1, id=6,
disabledPathsId=6]
 Raw values: {/auditexamplelogin1/login/no-error=null, /
auditexamplelogin1/login/args/userName=admin}
 Extracted: {/auditexamplelogin1/login/no-error/user=admin}
16:47:37,449 User:admin DEBUG [repo.audit.AuditComponentImpl] New
audit entry:
 Application ID: 6
 Entry ID: 131
 Values: {/auditexamplelogin1/login/no-error=null, /
auditexamplelogin1/login/args/userName=admin}
 Audit Data: {/auditexamplelogin1/login/no-error/user=admin}
```

You can see that the `AuthenticationService.authenticate` method generate two sets of "inbound" data: the `/alfresco-api/pre/AuthenticationService/authenticate` data is passed through before the service call is processed; the `/alfresco-api/post/AuthenticationService/authenticate` data is passed through after the service call has been processed. When logging in successfully, the post-call data is generated with a `no-error` path.

c. Perform a failed log in with user joe.

```
17:02:09,697 DEBUG [repo.audit.inbound]
Inbound audit values:
 /alfresco-api/pre/AuthenticationService/authenticate/args/
userName=joe
17:02:09,704 DEBUG [repo.audit.inbound]
Inbound audit values:
 /alfresco-api/post/AuthenticationService/authenticate/error=08200014
Failed to authenticate
Started at:

org.alfresco.repo.security.authentication.AbstractChainingAuthenticationServi
...
```

This is translated and recorded:

```
17:02:09,704 User:System DEBUG [repo.audit.AuditComponentImpl]
Extracted audit data:
```

```

Application: AuditApplication[ name=AuditExampleLogin1, id=6,
disabledPathsId=6]
Raw values: {/auditexamplelogin1/login/error=08200014 Failed to
authenticate
Started at:

org.alfresco.repo.security.authentication.AbstractChainingAuthenticationServi
...
17:02:09,704 DEBUG [repo.audit.AuditComponentImpl] New audit entry:
Application ID: 6
Entry ID: 135
Values: {/auditexamplelogin1/login/error=08200016 Failed to
authenticate
Started at:

org.alfresco.repo.security.authentication.AbstractChainingAuthenticationServi
...
Audit Data: {/auditexamplelogin1/login/error/user=joe}

```

- d. Notice that the failed log in did not generate any data for audit application `AuditExampleLogin2`. To understand this, look at the `PathMappings` section of the example:

```

<PathMappings>
  <PathMap source="/alfresco-api/post/AuthenticationService/
authenticate" target="/auditexamplelogin1/login"/>
  <PathMap source="/alfresco-api/post/AuthenticationService/
authenticate/no-error" target="/auditexamplelogin2/login"/>
</PathMappings>

```

Before any data is considered for persistence, the inbound data paths are remapped using the `PathMappings` configuration. The `/auditexamplelogin2/login` path is mapped onto `.../no-error` only, so failed logins were not recorded for the `AuditExampleLogin2` audit application, while the `AuditExampleLogin1` application recorded both successful and failed logins.

Audit recording values

The `RecordValue` element makes use of the `DataExtractor` definitions, but specifies when to be activated (`dataTrigger`) and where to get the data from (`dataSource`). Both the `dataTrigger` and `dataSource` attributes default to the path of the `RecordValue` element. Data is always written to the path where the `RecordValue` is declared. So, it is possible to trigger the `RecordValue` when a data path is present (such as a null value) and then to read a value from a completely different location.

1. Activate sample `/audit/alfresco-audit-example-extractors.xml` file.
2. Restart Alfresco (or restart the Audit subsystem).
3. Tail the log to capture `createNode` calls:

```
tail -f ../logs/catalina.out | grep -G "createNode" -A 200 -B 20
```

4. Log in to Explorer and add some content under **Company Home**.

```

20:18:52,817 User:admin DEBUG [repo.audit.AuditComponentImpl]
New audit entry:
Application ID: 8
Entry ID: 177
Values:
  /auditexampleextractors/args/properties=...
  /auditexampleextractors/args/assocQName={http://www.alfresco.org/model/
content/1.0}alfresco.log
  /auditexampleextractors/args/parentRef=workspace://
SpacesStore/37884669-0607-4527-940d-cb34b4f07d75
  /auditexampleextractors/no-error=null
  /auditexampleextractors/args/assocTypeQName={http://www.alfresco.org/
model/content/1.0}contains

```

```

/auditexampleextractors/args/nodeTypeQName={http://www.alfresco.org/
model/content/1.0}content
/auditexampleextractors/result=workspace://
SpacesStore/37884669-0607-4527-940d-cb34b4f07d75|workspace://SpacesStore/
c0fab6d-903f-4317-87d1-ec62de37089c|...
Audit Data:
/auditexampleextractors/create/out/a=workspace://
SpacesStore/37884669-0607-4527-940d-cb34b4f07d75|workspace://SpacesStore/
c0fab6d-903f-4317-87d1-ec62de37089c|...
/auditexampleextractors/create/derived/parent-node-name=Company Home
/auditexampleextractors/create/derived/parent-node-null=null
/auditexampleextractors/create/in/c={http://www.alfresco.org/model/
content/1.0}contains
/auditexampleextractors/create/in/d={http://www.alfresco.org/model/
content/1.0}alfresco.log
/auditexampleextractors/create/in/a=workspace://
SpacesStore/37884669-0607-4527-940d-cb34b4f07d75
/auditexampleextractors/create/derived/parent-node-type={http://
www.alfresco.org/model/content/1.0}folder
/auditexampleextractors/create/in/b={http://www.alfresco.org/model/
content/1.0}content

```

5. View the audited data using the query API:

```

% curl -u admin:admin "http://localhost:8080/alfresco/service/api/audit/
query/AuditExampleExtractors?limit=1&forward=false&verbose=true"
{
  "count":1,
  "entries":
  [
    {
      "id":177,
      "application":AuditExampleExtractors,
      "user":admin,
      "time":"2010-09-20T20:18:52.761+01:00",
      "values":
      {
        "\auditexampleextractors\create\out\
a":"workspace:\\\SpacesStore\37884669-0607-4527-940d-cb34b4f07d75|
workspace:\\\SpacesStore\c0fab6d-903f-4317-87d1-ec62de37089c|...
        ,"\auditexampleextractors\create\derived\parent-
node-name":"Company Home"
        ,"\auditexampleextractors\create\in\c":"{http:\\
\www.alfresco.org\model\content\1.0}contains"
        ,"\auditexampleextractors\create\in\d":"{http:\\
\www.alfresco.org\model\content\1.0}alfresco.log"
        ,"\auditexampleextractors\create\in\
a":"workspace:\\\SpacesStore\37884669-0607-4527-940d-cb34b4f07d75"
        ,"\auditexampleextractors\create\derived\parent-
node-type":"{http:\\\www.alfresco.org\model\content\1.0}folder"
        ,"\auditexampleextractors\create\in\b":"{http:\\
\www.alfresco.org\model\content\1.0}content"
      }
    }
  ]
}

```

The `/no-error` path was used as the `dataTrigger` to activate all the `RecordValue` elements, that is, the presence of the path triggered the data rather than any specific value. `/create/derived/...` audit values show how the parent node reference was used to record values that were not part of the inbound data set.

Using the example, to search for values that are not strings, use the following:

```

% curl -u admin:admin "http://localhost:8080/alfresco/service/api/audit/query/
AuditExampleExtractors/ \
auditexampleextractors/create/derived/parent-node-type?
\

```

```

        valueType=org.alfresco.service.namespace.QName&
        \
        value=%7Bhttp://www.alfresco.org/model/content/1.0%7Dfolder"
    {
        "count":1,
        "entries":
        [
            {
                "id":177,
                "application":AuditExampleExtractors,
                "user":admin,
                "time":"2010-09-20T20:18:52.761+01:00",
                "values":
null
            }
        ]
    }

% curl -u admin:admin "http://localhost:8080/alfresco/service/api/audit/query/
AuditExampleExtractors/ \
    auditexampleextractors/create/in/a?
    \
    valueType=org.alfresco.service.cmr.repository.NodeRef&
    \
    value=workspace://SpacesStore/37884669-0607-4527-940d-
cb34b4f07d75"
{
    "count":1,
    "entries":
    [
        {
            "id":177,
            "application":AuditExampleExtractors,
            "user":admin,
            "time":"2010-09-20T20:18:52.761+01:00",
            "values":
null
        }
    ]
}

```



It is not possible to restrict results to a specific value path. The path AND the value are enough to return a result. This does not usually yield duplicate results but it is not as restrictive as it should be. For example, generate the audit data and query for verbose output. Choose to search based on a path and a value and check that you get the correct number of results. Now choose a different path in the value list and query with that, that is, use a path and value that are not related.

Using values that have changed in a post method call

When using the `org.alfresco.repo.audit.AuditMethodInterceptor` Data Producer, which generates audit data for all public service API calls, it is sometimes useful to be able to audit before and after values in a 'post' call application, or to include values from before the call.

For example, the `nodeName` data extractor can only be called on a node that exists, so calling it after a delete has no effect.

The output of 'pre' call applications is available to 'post' call applications, which can be seen in the following example. The example shows auditing the deletion of nodes and includes the node name. The `nodeName` is evaluated in the 'pre' call application and copied in the 'post' call application.

```

<?xml version='1.0' encoding='UTF-8'?>
<Audit
  xmlns="http://www.alfresco.org/repo/audit/model/3.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

```

```

xsi:schemaLocation="http://www.alfresco.org/repo/audit/model/3.2 alfresco-
audit-3.2.xsd" >

  <DataExtractors>
    <DataExtractor name="simpleValue"
registeredName="auditModel.extractor.simpleValue"/>
    <DataExtractor name="nodeNameValue"
registeredName="auditModel.extractor.nodeName"/>
  </DataExtractors>

  <PathMappings>
    <PathMap source="/alfresco-api/pre/NodeService/deleteNode" target="/
preDelete" />
    <PathMap source="/alfresco-api/post/NodeService/deleteNode" target="/
postDelete" />
  </PathMappings>

  <Application name="PreCallDataDelete" key="preDelete">
    <RecordValue key="nodeName" dataExtractor="nodeNameValue" dataSource="/
preDelete/args/nodeRef" dataTrigger="/preDelete/args/nodeRef" />
  </Application>

  <Application name="PostDelete" key="postDelete">
    <RecordValue key="error" dataExtractor="simpleValue" dataSource="/
postDelete/error" dataTrigger="/postDelete/error" />
    <AuditPath key="deleteDetails">
      <RecordValue key="deletedNodeRef" dataExtractor="simpleValue"
dataSource="/postDelete/args/nodeRef" dataTrigger="/postDelete/args/nodeRef" /
>
      <RecordValue key="nodeName" dataExtractor="simpleValue" dataSource="/
postDelete/preCallData/preDelete/nodeName" dataTrigger="/postDelete/
preCallData/preDelete/nodeName" />
    </AuditPath>
  </Application>
</Audit>

```



The `dataSource` attribute of the final `<RecordValue>` element includes the output path of the 'pre' call application ("`preDelete/nodeName`"). This is prefixed by `preCallData/` much like the `args/` prefix for method arguments. To avoid 'pre' call applications from generating audit records themselves, rather than just generating output for the 'post' call applications, give them a name that starts with `PreCallData`.