

Alfresco 5.1

Administrator Guide



Contents

Preface.....	8
Audience.....	8
Typographic conventions used in this guide.....	8
Installing.....	9
QuickStart install guide.....	9
Installing Alfresco as a single instance.....	9
Installing Alfresco in a distributed environment.....	10
Installing Alfresco using setup wizards.....	12
Using the Alfresco installers on Linux.....	12
Using the Alfresco installers on Windows.....	22
Installing Alfresco One in an unattended mode.....	33
Installing Alfresco manually.....	35
Prerequisites for installing Alfresco.....	35
What you need to install Alfresco.....	40
Installing Alfresco on Tomcat.....	41
Tailoring your Alfresco installation.....	48
Installing an Alfresco Module Package.....	48
Installing additional software for Alfresco.....	51
Testing the Alfresco installation.....	55
Test and familiarize after installing Alfresco.....	55
Test and familiarize after configuring Alfresco.....	56
Test and familiarize after installing Alfresco in a clustered environment.....	56
Troubleshooting the installation.....	56
Uninstalling Alfresco.....	57
Uninstalling an AMP file.....	57
Uninstalling Alfresco on Linux.....	59
Uninstalling Alfresco on Windows.....	59
Installing Alfresco integrations.....	60
Installing and configuring the Alfresco EMC Centera Connector.....	60
Installing and configuring Alfresco Kofax Release script.....	70
Installing and configuring Alfresco Office Services.....	79
Installing and configuring Alfresco Outlook Integration.....	87
Installing and configuring Alfresco S3 Connector.....	87
Installing and configuring the Alfresco Transformation Server.....	90
Installing and configuring Alfresco Web Quick Start.....	97
Installing and configuring Google Docs integration.....	107
Upgrading.....	110
Alfresco upgrade paths.....	110
QuickStart upgrade guide.....	111
Upgrading Alfresco on a single instance.....	111
Upgrading Alfresco in a distributed environment.....	113
Upgrade prerequisites checklist.....	114
Database considerations.....	115
Upgrading Alfresco.....	115
Applying Service Packs and Hot Fixes.....	117
Prerequisites for using the Update Assistant.....	117
Installing Service Packs and Hot Fixes using the Update Assistant.....	118
Troubleshooting the Update Assistant.....	119
Upgrading search.....	120
Issues to consider before upgrading search.....	121
Upgrading from Lucene to Solr 4 search.....	121
Upgrading from Solr 1.4 to Solr 4 search.....	122

Validating an upgrade.....	125
Testing an Alfresco upgrade.....	125
Restoring production data.....	126
Configuring.....	128
Configuration overview.....	128
Using the Admin Console.....	129
Using the Share Admin Tools.....	133
Using the alfresco-global.properties file.....	134
Using a JMX client to change settings dynamically.....	136
Using the Java command line to change settings dynamically.....	140
Customizing Alfresco applications.....	141
Customizing individual configuration items.....	141
System path conventions.....	143
Configuring Alfresco subsystems.....	144
Subsystem categories.....	145
Subsystem configuration files.....	146
Subsystem properties.....	146
Mounting a subsystem.....	147
Mounting a subsystem with composite properties.....	147
Extension classpath.....	148
Configuring databases.....	149
Configuring Amazon RDS databases.....	149
Configuring a DB2 database.....	154
Configuring the MariaDB database connection.....	156
Configuring the MySQL database.....	156
Configuring an Oracle database.....	158
Configuring a PostgreSQL database.....	159
Configuring a SQL Server database.....	160
Advanced database configuration properties.....	162
Validating your database.....	164
Setting up authentication and security.....	165
Alfresco security.....	165
Managing Alfresco keystores.....	166
Cryptographic password hashing.....	171
Encrypting properties.....	172
Authentication subsystems.....	175
Alfresco authentication chain.....	177
Configuring authentication subsystems.....	180
Configuring synchronization.....	225
Managing authentication directories.....	228
Authorities.....	240
Defining permissions.....	243
Access Control Lists.....	246
Modifying access control.....	249
Public services.....	250
Implementation and services.....	253
Admin password in default authentication.....	257
Security policies and filters.....	259
Configuring search.....	264
Configuring search in Alfresco Share.....	264
Solr overview.....	265
Configure Solr search service.....	266
Solr security.....	281
Solr monitoring and troubleshooting.....	288

Solr backup and restore.....	293
Solr replication.....	294
Solr sharding.....	299
Full text search configuration properties for Solr index.....	323
Using Filtered search.....	325
Setting Solr logging.....	328
Calculate the memory needed for Solr nodes.....	328
Transactional metadata query.....	331
Configuring OpenSearch.....	334
Configuring the repository.....	335
Deploying Alfresco with a different context path.....	335
Tuning the JVM.....	336
Command line configuration.....	341
Configuring Alfresco to work with a web proxy.....	341
Configuring server administration properties.....	342
Controlling JVM system properties.....	345
Secure Sockets Layer (SSL) and the Alfresco repository.....	345
Configuring the repository cache.....	350
Adding a MIME type.....	351
Configuring metadata extraction.....	352
About aspects.....	352
About versioning.....	354
Setting up database replication.....	355
Customizing content transformations.....	356
Controlling indexes.....	357
Deferring the start of cron based jobs.....	357
Configuring Share.....	358
Configuring file servers.....	358
Enabling file servers.....	358
Configuring SMB/CIFS server.....	359
Configuring the FTP file server.....	365
Configuring email.....	367
Configuring inbound and outbound email.....	367
Configuring the Activities Feed.....	374
Enabling the Subscription Service.....	374
Configuring the email client with IMAP.....	375
Configuring LibreOffice.....	381
Changing the Office subsystems.....	381
LibreOffice configuration properties.....	382
Configuring OpenOffice transformations in place of LibreOffice.....	383
Configuring Smart Folders.....	383
What is a Smart Folder?.....	384
Prerequisites for using Smart Folders.....	385
Planning and implementing Smart Folders.....	386
Enabling Smart Folders.....	387
Smart Folders tutorial.....	387
Type-based, System, and Custom Smart Folders.....	395
Metadata inheritance.....	397
Smart Folder Template syntax.....	397
Smart Folders global properties settings.....	401
Best practices when using Smart Folders.....	402
Smart Folders technical FAQs.....	403
Content modeling with Model Manager.....	406
Managing models.....	408

Managing custom types, aspects, and properties.....	413
Tutorial: creating and using models in Alfresco Share.....	433
Model Manager video tutorials.....	450
Administering.....	451
Starting and stopping Alfresco.....	451
Starting the Alfresco server.....	451
Stopping the Alfresco server.....	451
Starting Alfresco Share.....	451
Managing Share features.....	452
Changing the Share theme.....	452
Changing the Share logo.....	452
Using templates.....	453
Managing tags.....	454
Managing categories.....	454
Managing cloud syncs.....	455
Managing sites.....	455
Managing users and groups.....	456
Managing users.....	456
Managing groups.....	460
Working with Alfresco licenses.....	462
Uploading a new license.....	463
Authorize users.....	463
Deauthorize an authorized user.....	464
Setting up clustering.....	465
Clustering prerequisites when upgrading to Alfresco One 5.1.....	465
Components of an Alfresco solution.....	466
Setting up Share cluster.....	467
Setting up repository server cluster.....	472
CIFS clustering through load balancer.....	477
Avoiding cluster clash.....	478
Tracking clustering issues.....	479
Setting up multi-tenancy.....	480
Enabling multi-tenancy.....	480
Managing tenants.....	480
Multi-tenancy administration.....	481
Features not supported in a multi-tenant environment.....	482
Creating and managing workflows.....	482
What is a workflow?.....	482
Setting up Hybrid workflow.....	484
Workflow architecture.....	485
Workflow tools.....	488
Process definitions.....	492
Task model.....	498
Setting up Activiti Designer.....	501
Deploying the task model.....	502
Deploying a process definition.....	502
Setting up Enterprise to Cloud Sync.....	502
Enterprise to Cloud Sync overview.....	503
Configuring Enterprise to Cloud Sync.....	503
Troubleshooting Enterprise to Cloud Sync.....	505
Managing transformations.....	507
Standard transform options.....	507
Additional transform options.....	567
File types that support preview and thumbnail generation.....	569

Setting up content stores.....	571
Content stores overview.....	571
Content store types.....	572
Content store selector.....	587
Managing content stores.....	590
Setting up and managing content replication.....	592
Configuring content replication.....	592
Enabling the Replication Service.....	593
Creating a new transfer target for content replication.....	593
Opening locked content in the source repository.....	594
Managing replication jobs.....	595
Importing and transferring files.....	598
Using the Bulk Import tool.....	598
Configuring the File System Transfer Receiver.....	606
Migrating.....	609
Migrating servers.....	609
Monitoring Alfresco.....	609
Downloading the JMX Dump.....	609
JMX monitoring and management extensions.....	610
Scheduled jobs.....	611
Scheduling cleanup of database tables.....	611
Backing up and restoring.....	612
Backing up and restoring the repository.....	612
Troubleshooting.....	615
Setting log levels.....	615
Error messages.....	616
Using the Node Browser.....	616
Debugging an Alfresco installation.....	617
Troubleshooting an upgrade.....	618
Troubleshooting rules and actions.....	618
Troubleshooting clustering.....	619
Troubleshooting LibreOffice subsystems.....	619
Troubleshooting the JMX Dumper.....	620
Troubleshooting CIFS.....	620
Troubleshooting NTLM.....	621
Troubleshooting WebDAV.....	621
OpenLDAP tips.....	622
Active Directory tips.....	625
Troubleshooting SMTP inbound email using StartTLS.....	625
Handling a higher rate of outbound TCP connections.....	615
Troubleshooting IMAP.....	625
Troubleshooting schema-related problems.....	626
Performing schema dumps.....	627
Performing schema validation.....	628
Reference.....	633
Properties available in a JMX client.....	633
JMX bean categories reference.....	636
JMX read-only monitoring beans.....	636
JMX editable management beans.....	670
Alfresco Full Text Search Reference.....	676
Search for a single term.....	676
Search for a phrase.....	676
Search for an exact term.....	677
Search for term expansion.....	677

Search for conjunctions.....	677
Search for disjunctions.....	677
Search for negation.....	677
Search for optional, mandatory, and excluded elements of a query.....	678
Search for fields.....	678
Search for wildcards.....	679
Search for ranges.....	679
Search for fuzzy matching.....	680
Search for proximity.....	680
Search for boosts.....	680
Search for grouping.....	680
Search for spans and positions.....	681
Escaping characters.....	681
Mixed FTS ID behavior.....	681
Search for order precedence.....	681
Search query syntax APIs.....	682
Search query templates.....	683
Search query literals.....	684
Forms reference.....	684
Administrator best practices.....	689
Tips for getting the most out of Alfresco.....	690
Common mistakes made by Alfresco administrators.....	690
Tips for Alfresco administrators.....	690
Glossary.....	691
A.....	691
B.....	695
C.....	695
D.....	697
E.....	698
F.....	699
G.....	699
H.....	700
I.....	700
J.....	701
K.....	702
L.....	703
M.....	703
N.....	704
O.....	705
P.....	705
Q.....	706
R.....	706
S.....	707
T.....	708
U.....	709
V.....	709
W.....	709
X.....	710
Y.....	711
Z.....	711
Copyright.....	712
Disclaimer.....	713

Preface

The purpose of this guide is to provide guidance on installing, configuring, and administering an Alfresco production environment.

This guide contains the following sections:

- **Installing Alfresco** describes how to install Alfresco and components
- **Upgrading Alfresco** describes how to upgrade Alfresco and components
- **Administering Alfresco** describes how to configure, maintain, and manage the system
- **Troubleshooting** describes how to analyze and troubleshoot various scenarios
- **Reference** provides additional information on topics discussed in this guide

Audience

This guide is intended to assist administrators to install, upgrade, configure, and manage an Alfresco production environment.

No specialist knowledge is assumed to install and configure Alfresco; however, the information provided in this guide assumes that you are familiar with the environment on which you are installing. Some administrative tasks also require knowledge of your environment and configuration processes.

Typographic conventions used in this guide

The following conventions are used in this guide to indicate types of information.

Convention	Type of information
bold	Identifies user interface elements and items to select, such as menu options, command buttons, and items in a list.
<code>monospace</code>	Identifies file and path names, input text, standard output, code, text the user types, and so on.
<i>italics</i>	Emphasizes importance and used for variable expressions, such as parameters. For example: <code>kill -9 <process_id></code>
CAPITALS	Refers to specific keys on the keyboard. For example: SHIFT, CTRL, or ALT
KEY+KEY	Refers to key combinations when you must press and hold down the first key, and then press another key. For example: CTRL+P or ALT+F4
	Refers to a note that provides supplemental information related to a topic.
	Refers to a note that provides important information to remember.
	Refers to a note that warns about the danger of doing or not doing something.
	Refers to a note that provides helpful information or a faster way of doing something.

Installing

Alfresco One ships with three installers:

- Alfresco One Installer: this is appropriate for the majority of users, and installs everything you require to run Alfresco. It corresponds to the installer used in previous versions of Alfresco.
 - Alfresco One Platform Installer: this installs the Alfresco repository, all required third party components (for example, ImageMagick), and links to a variety of developer and admin resources. If you have a clustered environment, you might want to use the Platform installer across these servers.
 - Alfresco One Share Installer: this installs Alfresco Share only, with its own Tomcat instance and the Share Services AMP. You might want to use the Share installer to connect to one or more repositories (that you have installed using the Platform installer).
-  Use the Share installer to connect to a repository that you installed using the Platform installer only. Other setups are not supported.

Alfresco Community Edition is designed to be deployed on a single server. As a result, it is shipped with a single Alfresco Community Edition Installer, which contains both the Alfresco Platform and Alfresco Share components. This is the same approach that is used in previous versions of Alfresco.

Depending on your system, you can install Alfresco using one of the following methods:

- Using a setup wizard, which contains the required software and components you need for evaluating Alfresco
- Using a standard WAR file to deploy Alfresco in a production environment

QuickStart install guide

You can install Alfresco as a single instance and also in a distributed and clustered environment.

This quick start guide provides you with simple instructions on how to download and install the Alfresco One Installer.

It is designed for users who just need a checklist to follow. For detailed step-by-step instructions for installing Alfresco, and more complex options, see [Installing](#).

Installing Alfresco as a single instance

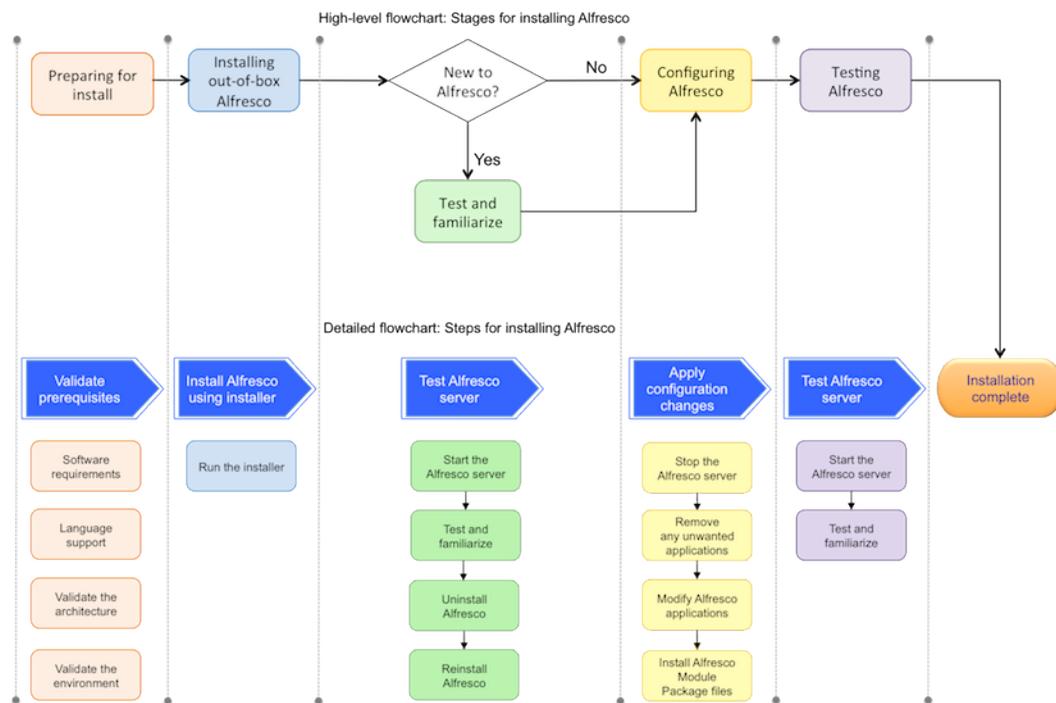
Follow these steps to install a single instance of Alfresco.

The main stages involved in setting up and configuring Alfresco are shown in the diagram. These include preparing your system for installation, installing Alfresco as an out-of-box application, configuring it based on your requirements, and finally, testing and getting familiar to Alfresco.

Each of these main stages consist of sub-steps, as shown in the diagram, which shows the sub-steps that need to be performed in order to complete each main stage.

 Note that the steps shown in the diagrams have a colour code. For example, Preparing for install stage consists of four sub-steps, namely, Software requirements, Language support, Validate the architecture, and Validate the environment.

To get started quickly with installing a single instance of Alfresco, follow the process shown. Click on each step to learn more about it.



1. [Software requirements](#) on page 35
2. [Language support](#) on page 35
3. [Validating the architecture](#) on page 36
4. [Validating the environment](#) on page 38
5. [Installing Alfresco using setup wizards](#) on page 12
6. [Starting the Alfresco server](#) on page 451
7. [Test and familiarize after installing Alfresco](#) on page 55
8. [Uninstalling Alfresco on Linux](#) on page 59
9. [Installing Alfresco using setup wizards](#) on page 12
10. [Stopping the Alfresco server](#) on page 451
11. [Tailoring your Alfresco installation](#) on page 48
12. [Customizing Alfresco applications](#) on page 141
13. [Installing an Alfresco Module Package](#) on page 48
14. [Starting the Alfresco server](#) on page 451
15. [Test and familiarize after configuring Alfresco](#) on page 56

Installing Alfresco in a distributed environment

Use these steps for installing Alfresco in a distributed environment.

The main stages involved in installing Alfresco in a cluster are shown in the diagram. You must install and configure your data on a single node first and then on the second node, and so on.

The main steps involved in the installing process include preparing your system for installation, [installing Alfresco on a single node](#), installing Alfresco on node 2, and finally, testing and getting familiar to Alfresco. Repeat the last two steps on all the other nodes in your system in series.

If you do not need Share on each instance in your cluster, you can use the Alfresco One Platform Installer instead of the Alfresco One Installer. See [Installing Alfresco on Linux using the Platform](#)

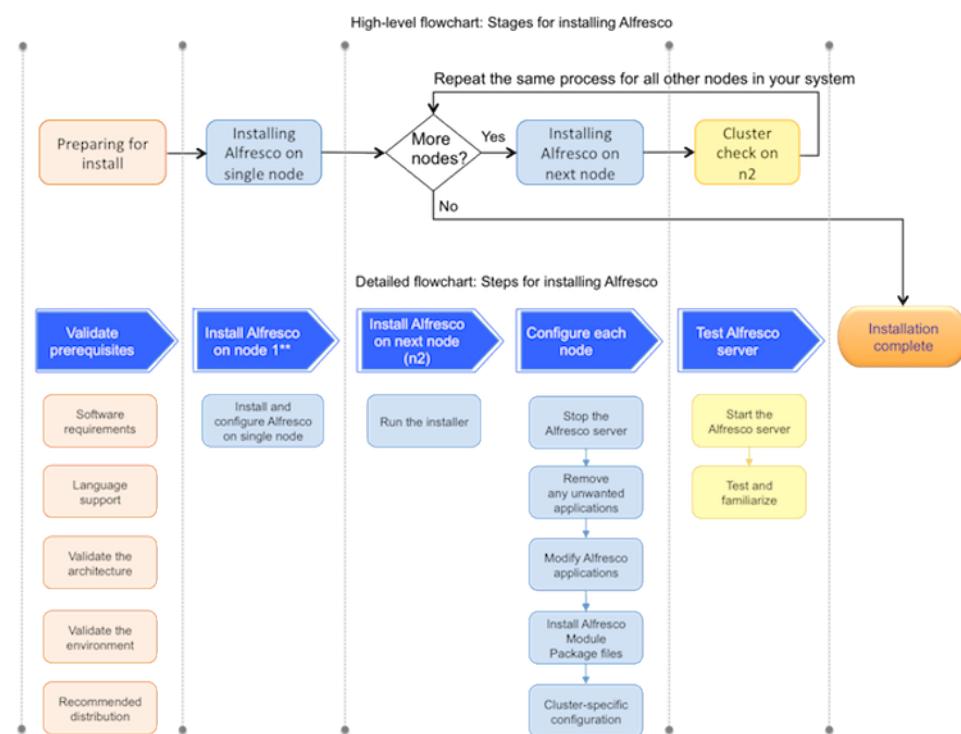
[Installer](#) on page 16 and [Installing Alfresco on Windows using the Platform Installer](#) on page 26 for more information.

Each of these main stages consist of sub-steps, as shown in the diagram, which displays the sub-steps that need to be performed in order to complete each main stage.

- 💡 Note that the steps shown in the diagrams have a colour code. For example, Preparing for install stage consists of four sub-steps, namely, Software requirements, Language support, Validate the architecture, Validate the environment, and Recommended distribution.
- 💡 Make sure you do not install and configure all the nodes in parallel. Follow in the installation process in series for all the nodes in your system.

To get started quickly with installing Alfresco in a distributed environment, follow the process shown.

Click on each task to learn more about it.



** Follow all the steps listed in the 'Installing Alfresco on a single instance' topic, starting from the 'Install Alfresco using installer' step through to 'Test Alfresco server' step.

💡 If you are running a clustered environment, Alfresco recommends that you first scale down your environment to a single node by disabling clustering, and then when you have upgraded, enable clustering by adding the nodes back again.

1. [Software requirements](#) on page 35
2. [Language support](#) on page 35
3. [Validating the architecture](#) on page 36
4. [Validating the environment](#) on page 38
5. [Recommendations for split architecture](#) on page 470
6. [Installing Alfresco as a single instance](#) on page 9
7. [Installing Alfresco using setup wizards](#) on page 12
8. [Stopping the Alfresco server](#) on page 451
9. [Tailoring your Alfresco installation](#) on page 48
10. [Customizing Alfresco applications](#) on page 141

11. [Installing an Alfresco Module Package on page 48](#)
12. [Cluster-specific configuration on page 12](#)
13. [Starting the Alfresco server on page 451](#)
14. [Test and familiarize after installing Alfresco in a clustered environment on page 56](#)

Cluster-specific configuration

If you have a distributed environment and want to implement clustering in Alfresco to improve the availability and performance of various services, you should enable clustering. This information describes the cluster-specific configurations for Alfresco.

Note that this is an optional step. For more information about how to set up a clustered environment, see [Setting up clustering](#).

Installing Alfresco using setup wizards

Use these methods to install Alfresco using the setup wizards.

Using the Alfresco installers on Linux

Use this information to install Alfresco on Linux, using the Alfresco One, Alfresco One Platform, and Alfresco One Share Installers.

Installing Alfresco on Linux using the Alfresco One Installer

The setup wizard for Linux installs all the software and components that you require for running Alfresco. This setup wizard installs Alfresco and additional software, including a Tomcat application server, PostgreSQL database, JDK, and LibreOffice.

 After installation, you must generate and install your own certificates to secure the installation. For more information, see [Generating Secure Keys for Solr 4 Communication](#).

1. Download the installation file: `alfresco-one-installer-201603-linux-x64.bin`
Files are available from the [Support Portal](#).
This Alfresco setup wizard is for 64-bit Linux systems.
2. Execute the downloaded file using the following commands:

```
chmod 777 alfresco-one-installer-201603-linux-x64.bin  
./alfresco-one-installer-201603-linux-x64.bin
```

 You should avoid running applications as the root (Linux administrator) user where possible, however if you must install Alfresco as the root user, then run the `alfresco-one-installer-201603-linux-x64.bin` file with `sudo` specified:

```
chmod 777 alfresco-one-installer-201603-linux-x64.bin  
sudo ./alfresco-one-installer-201603-linux-x64.bin
```

The setup wizard starts.

3. Select the language that you wish to use for the installation.

This sets the language to be used for the setup wizard.

 This does not set the language that is used in Alfresco.

4. On the **Setup - Alfresco One** window, click **Next**.
5. On the **Installation Type** window, choose how you want to use the setup wizard.

There are two types of installation in the setup wizard:

Options	Description
Easy	Easy type installs Alfresco using the default options and configuration. This install type requires you to enter information in only two fields: the Alfresco install location and the administrator password. Choose this route to install Alfresco with the default environment.  If you have previously installed Alfresco and the server is running, when you run this setup wizard again, you might be prompted to enter alternative port numbers for the components and services that you install, for example, for the Tomcat application server, FTP port, and the RMI port.
Advanced	Advanced type installs Alfresco but lets you configure the server ports and service properties. You can also choose which additional components to install.

 When choosing between the **Easy** or **Advanced** installation, consider your basic requirement. If you are installing on a demo system for evaluation purpose only, Alfresco recommends that you use the **Easy** installation option. However, if you want to connect to an existing database server for Alfresco and also, want to see how the various components are being configured, use the **Advanced** installation option.

To complete the **Easy** setup wizard:

- a. Select **Easy**, and then click **Next**.
- b. On the **Installation Folder** window, click **Next** to accept the default location.
 You must use ASCII characters only when setting the installation folder using the Alfresco setup wizard.
- c. On the **Admin Password** window, enter a password for the Administrator user (`admin`).
 You must use ASCII characters only when setting the password using the Alfresco setup wizard. If you need to reset the password (to include non-ASCII characters) after installation, see [Changing a user's password](#) on page 459.
- d. Repeat the password, and then click **Next**.
- e. Click **Next** through the remaining windows in the setup wizard.
- f. Click **Finish** to complete the installation.

Go to the step for the **Completing the Alfresco One Setup Wizard** window and launching Alfresco Share.

To complete the **Advanced** setup wizard, select **Advanced** and then click **Next**.

Follow the remaining steps in this task.

6. On the **Select Components** window, select the components that you want to install. Deselect the components that you do not want to install.
 - Java (this is JRE only)
 - PostgreSQL
 - LibreOffice
 - Solr1
 The Solr 1 option should be used only for migration to Solr 4.
 - Solr4
 - Alfresco Office Services
 - Web Quick Start
 - Google Docs Integration

7. When you have finished selecting the components, click **Next**.
 8. On the **Installation Folder** window, click **Next** to accept the default location.
- For example, the default location is /opt/alfresco-one.

Alternatively, click the  icon to choose another location.

 You must use ASCII characters only when setting the installation folder using the Alfresco setup wizard.

9. On the **Database Server Parameters** window, enter a port number for your database. Enter a suitable port number or click **Next** to accept the default of 5432.
10. On the **Tomcat Port Configuration** window, enter the following Tomcat configuration parameters:

- a. Web Server Domain

For example, the default is 127.0.0.1.

The URL <http://127.0.0.1:8080/share> is based on the web server domain and the Tomcat port number that you specify on the **Tomcat Port Configuration** window. The default of 127.0.0.1 can be used on this machine to verify that Alfresco is running successfully. However, it is not an externally addressable URL, which means that it is not possible for users on other machines to access this URL. To make sure that other users can access the machine where Alfresco is installed, you need to define and create a publicly addressable name.

- b. Tomcat Server Port

For example, the default is 8080.

- c. Tomcat Shutdown Port

For example, the default is 8005.

- d. Tomcat SSL Port

For example, the default is 8443.

- e. Tomcat AJP Port

For example, the default is 8009.

11. (Optional) If you are installing the LibreOffice component, the **LibreOffice Server Port** window displays. Enter a port number on which the LibreOffice server will listen.
12. On the **Sharded Solr installation** window, specify if you are using a sharded Solr installation, and then click **Next**.

 When using the installer, the templates used to create shards do not use the port specified in the installer. To set the port manually when creating a shard, see [Installing and configuring Solr shards](#).

13. On the **Alfresco FTP Port** window, enter a port number for the Alfresco FTP server, and then click **Next**.
14. On the **Alfresco RMI Port** window, enter a port number for Alfresco to execute remote commands, and then click **Next**.
15. On the **Admin Password** window, type a password. Repeat the password, and then click **Next**.

This sets the password for the Alfresco Administrator user account (`admin`).

 You must use ASCII characters only when setting the password using the Alfresco setup wizard. If you need to reset the password (to include non-ASCII characters) after installation, see [Changing a user's password](#) on page 459.

16. On the **Warning** window, review the list of environment notifications for your installation of Alfresco installation.

This list of environment notifications is based on an evaluation of your installation environment while the setup wizard is running.

17. On the **Ready to Install** window, click **Next**.

The **Installing** window displays, showing the progress of the installation.

18. On the **Completing the Alfresco One Setup Wizard** window, click **Finish**.

This window shows check boxes that determine whether you will see the Readme file, the [Getting Started](#) web page, and also whether to launch Alfresco One. By default, these options are selected and will launch when you click **Finish**. If you do not want to start Alfresco at this point, deselect the **Launch Alfresco One** check box.

-  If you are installing the S3 connector as part of your Alfresco installation, deselect the **Launch Alfresco One** check box. You must not start Alfresco before applying the S3 AMP file.

19. Click **OK** to close the Readme.

The Alfresco server starts and then Alfresco launches in your default browser.

-  It can take several minutes to start the Alfresco server and to launch Alfresco. Your browser opens and tries to connect to <http://127.0.0.1:8080/share>.

20. Log on to Alfresco as the `admin` user. Enter the password that you specified in the **Admin Password** window.

The Alfresco server is launched automatically as a service called `alfresco`. This service comprises the following individual services:

- `postgresql`
- `Tomcat Server`

If you did not automatically launch Alfresco at the end of the setup wizard, to start Alfresco, you need to start all the services.

21. Manually start the Alfresco server:

Browse to `/opt/alfresco-one/` (the installation folder that you created in 8 on page 14). As an administrator, run

```
./alfresco.sh start
```

22. To fully stop Alfresco, you must stop all the services:

Browse to `/opt/alfresco-one/` (the installation folder that you created in 8 on page 14). As an administrator, run

```
./alfresco.sh stop
```

Installing Linux libraries manually

Use this information to install Linux libraries manually on supported Linux distributions, such as Ubuntu, SUSE and RedHat.

LibreOffice requires the following libraries to be installed on your system:

- `libfontconfig`
- `libICE`
- `libSM`
- `libXrender`
- `libXext`

- libxinerama
- libcups
- libGLU

On some Linux distributions, such as Ubuntu, SUSE, and RedHat, the Alfresco setup wizard will validate whether or not the required libraries are present. If the required libraries are missing, you will get a warning message. You can install them using your package manager from the command line.

If LibreOffice does not start up normally with Alfresco, test manually; for example, by running this startup script:

```
start ex.
{installdir}/libreoffice/scripts/libreoffice_ctl.sh start
status ex. {installdir}
/libreoffice/scripts/libreoffice_ctl.sh status
```

If you receive errors that indicate that a library missing, work with your system administrator to add the missing library or its equivalent from your configured repositories.

Installing Alfresco on Linux using the Platform Installer

The Alfresco One Platform Installer for Linux installs the Alfresco repository and all the software and components that you require for running the Alfresco platform; for example, a Tomcat application server, PostgreSQL database, JDK, LibreOffice, Solr4 and other required software like ImageMagick. It does not install Alfresco Share.

-  The Alfresco One Installer is recommended for most purposes. See [Installing Alfresco on Linux using the Alfresco One Installer](#) on page 12 for more information. Use the Alfresco One Platform Installer only if you have a specific requirement for it.
-  After installation, you must generate and install your own certificates to secure the installation. For more information, see [Generating Secure Keys for Solr 4 Communication](#).

1. Download the following installation file:

```
alfresco-one-platform-installer-5.1-linux-x64.bin
```

Files are available from the [Support Portal](#).

This Alfresco setup wizard is for 64-bit Linux systems.

2. Execute the downloaded file using the following commands:

```
chmod 777 alfresco-one-platform-installer-5.1-linux-x64.bin
./alfresco-one-platform-installer-5.1-linux-x64.bin
```

 You should avoid running applications as the root (Linux administrator) user where possible, however if you must install Alfresco as the root user, then run the `alfresco-one-platform-installer-5.1-linux-x64.bin` file with `sudo` specified:

```
chmod 777 alfresco-one-platform-installer-5.1-linux-x64.bin
sudo ./alfresco-one-platform-installer-5.1-linux-x64.bin
```

The setup wizard starts.

3. Select the language that you wish to use for the installation.

This sets the language to be used for the setup wizard.

 This does not set the language that is used in Alfresco.

4. On the **Setup - Alfresco One Platform** window, click **Next**.

5. On the **Installation Type** window, choose how you want to use the setup wizard.

There are two types of installation in the setup wizard:

Options	Description
Easy	Easy type installs Alfresco using the default options and configuration. This install type requires you to enter information in only two fields: the Alfresco install location and the administrator password. Choose this route to install Alfresco with the default environment.  If you have previously installed Alfresco and the server is running, when you run this setup wizard again, you might be prompted to enter alternative port numbers for the components and services that you install, for example, for the Tomcat application server, FTP port, and the RMI port.
Advanced	Advanced type installs Alfresco but lets you configure the server ports and service properties. You can also choose which additional components to install.

 When choosing between the **Easy** or **Advanced** installation, consider your basic requirement. If you are installing on a demo system for evaluation purpose only, Alfresco recommends that you use the **Easy** installation option. However, if you want to connect to an existing database server for Alfresco and also, want to see how the various components are being configured, use the **Advanced** installation option.

To complete the **Easy** setup wizard:

- a. Select **Easy**, and then click **Next**.
- b. On the **Installation Folder** window, click **Next** to accept the default location.
 You must use ASCII characters only when setting the installation folder using the Alfresco setup wizard.
- c. On the **Admin Password** window, enter a password for the Administrator user (admin).
 You must use ASCII characters only when setting the password using the Alfresco setup wizard. If you need to reset the password (to include non-ASCII characters) after installation, see [Changing a user's password](#) on page 459.
- d. Repeat the password, and then click **Next**.
- e. Click **Next** through the remaining windows in the setup wizard.
- f. Click **Finish** to complete the installation.

Go to the step for the **Completing the Alfresco One Platform Setup Wizard** window and launching Alfresco Platform.

To complete the **Advanced** setup wizard, select **Advanced** and then click **Next**.

Follow the remaining steps in this task.

6. On the **Select Components** window, select the components that you want to install. Deselect the components that you do not want to install.
 - Java (this is JRE only)
 - PostgreSQL
 - LibreOffice
 - Solr4
 - Alfresco Office Services
 7. When you have finished selecting the components, click **Next**.
 8. On the **Installation Folder** window, click **Next** to accept the default location.
- For example, the default location is /opt/alfresco-one-platform.



Alternatively, click the icon to choose another location.

You must use ASCII characters only when setting the installation folder using the Alfresco setup wizard.

9. On the **Database Server Parameters** window, enter a port number for your database. Enter a suitable port number or click **Next** to accept the default of 5432.
10. On the **Tomcat Port Configuration** window, enter the following Tomcat configuration parameters:
 - a. Web Server Domain
For example, the default is 127.0.0.1.
The URL <http://127.0.0.1:8080/share> is based on the web server domain and the Tomcat port number that you specify on the **Tomcat Port Configuration** window. The default of 127.0.0.1 can be used on this machine to verify that Alfresco is running successfully. However, it is not an externally addressable URL, which means that it is not possible for users on other machines to access this URL. To make sure that other users can access the machine where Alfresco is installed, you need to define and create a publicly addressable name.
 - b. Tomcat Server Port
For example, the default is 8080.
 - c. Tomcat Shutdown Port
For example, the default is 8005.
 - d. Tomcat SSL Port
For example, the default is 8443.
 - e. Tomcat AJP Port
For example, the default is 8009.
11. (Optional) If you are installing the LibreOffice component, the **LibreOffice Server Port** window displays. Enter a port number on which the LibreOffice server will listen.
12. On the **Sharded Solr installation** window, specify if you are using a sharded Solr installation, and then click **Next**.

When using the installer, the templates used to create shards do not use the port specified in the installer. To set the port manually when creating a shard, see [Installing and configuring Solr shards](#).
13. On the **Alfresco FTP Port** window, enter a port number for the Alfresco FTP server, and then click **Next**.
14. On the **Alfresco RMI Port** window, enter a port number for Alfresco to execute remote commands, and then click **Next**.
15. On the **Admin Password** window, type a password. Repeat the password, and then click **Next**.
This sets the password for the Alfresco Administrator user account (`admin`).

You must use ASCII characters only when setting the password using the Alfresco setup wizard. If you need to reset the password (to include non-ASCII characters) after installation, see [Changing a user's password](#) on page 459.
16. On the **Service Startup Configuration** window, select whether you want the service to start up manually or automatically.

17. On the **Warning** window, review the list of environment notifications for your installation of Alfresco installation.

This list of environment notifications is based on an evaluation of your installation environment while the setup wizard is running.

18. On the **Ready to Install** window, click **Next**.

The **Installing** window displays, showing the progress of the installation.

19. On the **Completing the Alfresco One Platform Setup Wizard** window, click **Finish**.

This window shows check boxes that determine whether you will see the Readme file, the [Getting Started](#) web page, and also whether to launch the Alfresco Platform. By default, these options are selected and will launch when you click **Finish**. If you do not want to start Alfresco at this point, deselect the **Launch Alfresco One Platform** check box.

 If you are installing the S3 connector as part of your Alfresco installation, deselect the **Launch Alfresco One Platform** check box. You must not start Alfresco before applying the S3 AMP file.

20. Click **OK** to close the Readme.

The Alfresco server starts and then Alfresco Platform launches in your default browser.

The browser page that is displayed gives you links to resources such as the online documentation, Alfresco WebDav, Alfresco WebScripts, the Alfresco Admin Console, Alfresco Support, and CMIS 1.0 and 1.1 documentation.

 It can take several minutes to start the Alfresco server and to launch the page. Your browser opens and tries to connect to <http://127.0.0.1:8080/alfresco>.

The Alfresco server is launched automatically as a service called `alfresco`. This service comprises the following individual services:

- `postgresql`
- `Tomcat Server`

21. To access resources such as the Alfresco Admin Console, log on as the `admin` user. Enter the password that you specified in the **Admin Password** window.

If you did not automatically launch Alfresco at the end of the setup wizard, to start Alfresco, you need to start all the services.

22. Manually start the Alfresco server:

Browse to `/opt/alfresco-one-platform/` (the installation folder that you created in 8 on page 17). As an administrator, run

```
./alfresco.sh start
```

23. To fully stop Alfresco, you must stop all the services:

Browse to `/opt/alfresco-one-platform/` (the installation folder that you created in 8 on page 17). As an administrator, run

```
./alfresco.sh stop
```

Installing Alfresco on Linux using the Share Installer

The Alfresco One Share Installer for Linux installs Alfresco Share only, with its own Tomcat application server and the Share Services AMP.

 The Alfresco One Installer is recommended for most purposes. See [Installing Alfresco on Linux using the Alfresco One Installer](#) on page 12 for more information. Use the Alfresco One Share Installer only if you have a specific requirement for it; for example, if you are linking to an Alfresco repository on a different server, that has been installed using the Alfresco One Platform Installer.

- Download the following installation file:

`alfresco-one-share-installer-5.1-linux-x64.bin`

Files are available from the [Support Portal](#).

This Alfresco setup wizard is for 64-bit Linux systems.

- Execute the downloaded file using the following commands:

```
chmod 777 alfresco-one-share-installer-5.1-linux-x64.bin
./alfresco-one-share-installer-5.1-linux-x64.bin
```

 Avoid running applications as the root (Linux administrator) user where possible, however if you must install Alfresco as the root user, then run the `alfresco-one-share-installer-5.1-linux-x64.bin` file with `sudo` specified:

```
chmod 777 alfresco-one-share-installer-5.1-linux-x64.bin
sudo ./alfresco-one-share-installer-5.1-linux-x64.bin
```

The setup wizard starts.

- Select the language that you wish to use for the installation.

This sets the language to be used for the setup wizard.

 This does not set the language that is used in Alfresco.

- On the **Setup - Alfresco One Share** window, click **Next**.

- On the **Installation Type** window, choose how you want to use the setup wizard.

There are two types of installation in the setup wizard:

Options	Description
Easy	<p>Easy type installs Alfresco One Share using the default options and configuration. This install type requires you to enter locations for where you want to install Share, and which repository you want to connect to. Choose this route to install with the default environment.</p> <p> If you have previously installed Alfresco and the server is running, when you run this setup wizard again, you might be prompted to enter alternative port numbers for the components and services that you install, for example, for the Tomcat application server, FTP port, and the RMI port.</p>
Advanced	Advanced type installs Alfresco One Share but asks you to specify the components that you want to install, and the Tomcat configuration parameters that you require.

 When choosing between the **Easy** or **Advanced** installation, consider your basic requirement. The Alfresco One Share Installer provides a Tomcat server to run Alfresco Share, so if you have your Alfresco repository on the same machine as your Alfresco Share installation, you must use the Advanced setup, so that you can specify alternative Tomcat ports for the Share Tomcat server.

To complete the **Easy** setup wizard:

- Select **Easy**, and then click **Next**.

- On the **Installation Folder** window, click **Next** to accept the default location.

 You must use ASCII characters only when setting the installation folder using the Alfresco setup wizard.

- On the **Connect Alfresco One Share to the repository** window, enter the location of your Alfresco repository, in the format: `http://yourserver:port/alfresco` or accept the default setting `http://localhost:8080/alfresco`

- On the **Ready to Install** window, click **Next**.

The **Installing** window displays, showing the progress of the installation.

7. On the **Completing the Alfresco One Share Setup Wizard** window, click **Finish**.

This window shows check boxes that determine whether you will see the Readme file, the [Getting started](#) web page, and also whether to launch Alfresco Share. By default, these options are selected and will launch when you click **Finish**. If you do not want to start Alfresco at this point, deselect the **Launch Alfresco One Share** check box.

8. Click **Finish** to complete the installation.
9. To complete the **Advanced** setup wizard, select **Advanced** and then click **Next**.
Follow the remaining steps in this task.
10. On the **Select Components** window, select the components that you want to install. Deselect the components that you do not want to install.
Java is the only option available to select or deselect.
11. When you have finished selecting the components, click **Next**.
12. On the **Installation Folder** window, click **Next** to accept the default location.
For example, the default location is `/opt/alfresco-one-share`.



Alternatively, click the icon to choose another location.

 You must use ASCII characters only when setting the installation folder using the Alfresco setup wizard.

13. On the **Tomcat Port Configuration** window, enter the following Tomcat configuration parameters:

 The Alfresco One Share Installer provides its own Tomcat server to run Alfresco Share, so if you have your Alfresco repository on the same machine as your Alfresco Share installation, you must specify alternative Tomcat ports for the Share Tomcat server.

- a. Web Server Domain

For example, the default is 127.0.0.1

The URL `http://127.0.0.1:8080/share` is based on the web server domain and the Tomcat port number that you specify on the **Tomcat Port Configuration** window. The default of 127.0.0.1 can be used on this machine to verify that Alfresco is running successfully. However, it is not an externally addressable URL, which means that it is not possible for users on other machines to access this URL. To make sure that other users can access the machine where Alfresco is installed, you need to define and create a publicly addressable name.

- b. Tomcat port

For example, the default is 8080.

- c. Tomcat Shutdown port

For example, the default is 8005.

- d. Tomcat SSL Port

For example, the default is 8443.

- e. Tomcat AJP Port

For example, the default is 8009.

14. On the **Connect Alfresco One Share to the repository** window, enter the location of your Alfresco repository, in the format: `http://yourserver:port/alfresco` or accept the default setting `http://localhost:8080/alfresco`
15. On the **Ready to Install** window, click **Next**.
The **Installing** window displays, showing the progress of the installation.
16. On the **Completing the Alfresco One Share Setup Wizard** window, deselect the **Launch Alfresco One Share** check box, and click **Finish**.
This window shows check boxes that determine whether you will see the Readme file, the [Getting Started](#) web page, and also whether to launch Alfresco Share. By default, these options are selected and will launch when you click **Finish**. We recommend that you do not launch Alfresco Share yet, because you need to complete the next steps before starting your servers.
17. Click **OK** to close the Readme.
18. Link your Share instance to a repository, by applying the Share Services AMP to your chosen Alfresco Platform instance.
 -  If you do not apply the Share Services AMP to the Alfresco repository, Alfresco Share will not work correctly, and when you start up Alfresco Share, you will see the message:
`Alfresco is running without Share Services. See your System Administrator for more details.`
 - a. Navigate to the `alfresco-one-share/amps` directory, and locate the `alfresco-share-services.amp` file.
 - b. Copy the `alfresco-share-services.amp` file to your Alfresco instance (`alfresco-one-platform/amps`), on the machine that hosts your Alfresco Platform repository.
 - c. Use the guidance in [Installing an Alfresco Module Package](#) on page 48 to apply the AMP to your repository.
 - d. Restart the Alfresco Platform instance where you have installed the Share Services AMP to see that the changes have been applied.
19. Manually start your Tomcat server for the instance that you have just installed with the Alfresco One Share Installer:
From the `alfresco-one-share` installation directory: `service alfresco start`
20. Log on to Alfresco Share (`http://localhost:port/share`) as the **admin** user. Enter the password that you specified in the **Admin Password** window.
21. Check for error messages as you open Alfresco Share.
Use the information in [Troubleshooting the installation](#) on page 56 to help you.
If you need to change any settings after installation, these are stored in the `share-config-custom.xml` file. See [Configuring Share with the share-config-custom.xml file](#) for more information.
22. To fully stop Alfresco, you must stop all the services:
`service alfresco stop`

Using the Alfresco installers on Windows

Use this information to install Alfresco on Windows, using the Alfresco One, Alfresco One Platform, and Alfresco One Share Installers.

Installing Alfresco on Windows using the Alfresco One Installer

The setup wizard for Microsoft Windows installs all the software and components that you require for running Alfresco. This setup wizard installs Alfresco and additional software, including a Tomcat application server, PostgreSQL database, JDK, and LibreOffice.

-  After installation, you must generate and install your own certificates to secure the installation. For more information, see [Generating Secure Keys for Solr 4 Communication](#).

1. Download the following installation file:

`alfresco-one-installer-201603-win-x64.exe`

Files are available from the [Support Portal](#).

The Alfresco setup wizard is for 64-bit Windows systems.

2. Double-click the downloaded file.
3. Select the language that you wish to use for the installation.

This sets the language to be used for the setup wizard.

 This does not set the language that is used in Alfresco.

4. On the **Setup - Alfresco One** window, click **Next**.
5. On the **Installation type** window, choose how you want to use the setup wizard.

There are two types of installation in the setup wizard:

Options	Description
Easy	<p>Easy type installs Alfresco using the default options and configuration. This install type requires only two fields: install location and administrator password. Choose this route to install Alfresco with the default environment.</p> <p> If you have previously installed Alfresco and the server is running, when you run this setup wizard again, you might be prompted to enter alternative port numbers for the components and services that you install, for example, for the Tomcat application server, FTP port, and the RMI port.</p>
Advanced	<p>Advanced type installs Alfresco but lets you configure the server ports and service properties. You can also choose which additional components to install.</p>

 When choosing between the **Easy** or **Advanced** installation, consider your basic requirement. If you are installing on a demo system for evaluation purpose only, Alfresco recommends that you use the **Easy** installation option. However, if you want to connect to an existing database server for Alfresco and also, want to see how the various components are being configured, use the **Advanced** installation option.

To complete the **Easy** setup wizard:

- a. Select **Easy**, and then click **Next**.
- b. On the **Installation Folder** window, click **Next** to accept the default location.
- c. On the **Admin Password** window, enter a password for the Administrator user (admin).
- d. Repeat the password, and then click **Next**.

- e. Click **Next** through the remaining windows in the setup wizard.
 - f. Click **Finish** to complete the installation.
- Go to the step for the **Completing the Alfresco One Setup Wizard** window and launching Alfresco Share.

To complete the **Advanced** setup wizard, select **Advanced** and then click **Next**.

Follow the remaining steps in this task.

6. On the **Select Components** window, select the components that you want to install. Deselect the components that you do not want to install.

- Java (this is JRE only)
- PostgreSQL
- LibreOffice
- Solr1

 The Solr 1 option should be used only for migration to Solr 4.

- Solr4
- Alfresco Office Services
- Web Quick Start
- Google Docs Integration

7. When you have finished selecting the components, click **Next**.

8. On the **Installation Folder** window, click **Next** to accept the default location.

For example, the default location is C:\alfresco-one.



Alternatively, click the  icon to choose another location.

 There is a known problem related to the use of virtual Windows drives and permission restrictions on the C drive in Windows. If you select a virtual Windows drive, and you have selected additional components in step 6, you might receive an error message during installation:

```
09160000 An IO error was encountered during deployment of the AMP
into the WAR
```

The installation will complete, but additional components will not be added to the Alfresco installation. The error does not occur if the virtual drive is associated with a folder that is not on the C drive (that is, any other physical or virtual hard disk on the Windows machine). You have two options:

- Do not use the SUBST command for folders on the C drive (in other words, use SUBST on other drives instead)
- Use a Virtual Hard Disk (VHD), which behaves like a physical disk, in place of the virtual Windows drive for the install. For guidance, see [Create and use a Virtual Hard Disk](#).

9. On the **Database Server Parameters** window, enter a port number for your database.

Enter a suitable port number or click **Next** to accept the default of 5432.

10. On the **Tomcat Port Configuration** window, enter the following Tomcat configuration parameters, and then click **Next**.

- a. Enter the Web Server domain number.

For example, the default is 127.0.0.1.

The URL <http://127.0.0.1:8080/share> is based on the web server domain and the Tomcat port number that you specify on the **Tomcat Port Configuration** window. The default of 127.0.0.1 can be used on this machine to verify that Alfresco is running successfully. However, it is not an externally addressable URL, which means that it is not possible for users on other machines to access this URL. To make sure that other users can access the machine where Alfresco is installed, you need to define and create a publicly addressable name.

- b. Enter the port number for the Tomcat web application.
For example, the default is 8080.
 - c. Enter the Tomcat Shutdown port number.
For example, the default is 8005.
 - d. Enter the Tomcat SSL port number.
For example, the default is 8443.
 - e. Enter the Tomcat AJP Port number.
For example, the default is 8009.
11. (Optional) If you are installing the LibreOffice component, the **LibreOffice Server Port** window displays. Enter a port number on which the LibreOffice server will listen.
 12. On the **Sharded Solr installation** window, specify if you are using a sharded Solr installation, and then click **Next**.
 - ⚠ When using the installer, the templates used to create shards do not use the port specified in the installer. To set the port manually when creating a shard, see [Installing and configuring Solr shards](#).
 13. On the **Alfresco FTP Port** window, enter a port number for the Alfresco FTP server, and then click **Next**.
 14. On the **Alfresco RMI Port** window, enter a port number for the RMI service, and then click **Next**.
 15. On the **Admin Password** window, enter a password. Repeat the password, and then click **Next**.
This sets the password for the Alfresco Administrator user account (`admin`).
 - ⚠ You must use ASCII characters only when setting the password using the Alfresco setup wizard. If you need to reset the password (to include non-ASCII characters) after installation, see [Changing a user's password](#) on page 459.
 16. On the **Service Startup Configuration** window, select whether you want the service to start up manually or automatically.
 17. On the **Warning** window, review the list of environment notifications for your installation of Alfresco installation.
This list of environment notifications is based on an evaluation of your installation environment while the setup wizard is running.
 18. On the **Ready to Install** window, click **Next**.
The **Installing** window displays, showing the progress of the installation.
 19. On the **Completing the Alfresco One Setup Wizard** window, click **Finish**.
This window shows check boxes that determine whether you will see the Readme file, the [Getting Started](#) web page, and also whether to launch Alfresco One. By default, these options are selected and will launch when you click **Finish**. If you do not want to start Alfresco at this point, deselect the **Launch Alfresco One** check box.

-  If you are installing the S3 connector as part of your Alfresco installation, deselect the **Launch Alfresco One** check box. You must not start Alfresco before applying the S3 AMP file.
20. Click **OK** to close the Readme.

The Alfresco server starts and then Alfresco launches in your default browser.

-  It can take several minutes to start the Alfresco server and to launch Alfresco. Your browser opens and tries to connect to <http://127.0.0.1:8080/share>.
21. Log on to Alfresco as the `admin` user. Enter the password that you specified in the **Admin Password** window.

The Alfresco server is launched as a Windows service. To manage the server, open the Control Panel **Services** window. The services that will be running for an Alfresco install using the default options are:

- `alfrescoPostgreSQL`
- `alfrescoTomcat`

If you did not automatically launch Alfresco at the end of the installation wizard, to start Alfresco, you need to start all the services. Use the `servicerun start` script in the installation directory or select **All Programs > Alfresco One > Alfresco One Service > Start Alfresco One Service**.

22. To fully stop Alfresco, you must stop all the services. Use the `servicerun stop` script in the installation directory or select **All Programs > Alfresco One > Alfresco One Service > Stop Alfresco One Service**.

Installing Alfresco on Windows using the Platform Installer

The Alfresco One Platform Installer for Windows installs the Alfresco repository and all the software and components that you require for running the Alfresco platform; for example, a Tomcat application server, PostgreSQL database, JDK, LibreOffice, Solr4 and other required software like ImageMagick. It does not install Alfresco Share.

-  The Alfresco One Installer is recommended for most purposes. See [Installing Alfresco on Linux using the Alfresco One Installer](#) on page 12 for more information. Use the Alfresco One Platform Installer only if you have a specific requirement for it.
-  After installation, you must generate and install your own certificates to secure the installation. For more information, see [Generating Secure Keys for Solr 4 Communication](#).

1. Download the following installation file:

`alfresco-one-platform-installer-5.1-win-x64.exe`

Files are available from the [Support Portal](#).

The Alfresco setup wizard is for 64-bit Windows systems.

2. Double-click the downloaded file.

3. Select the language that you wish to use for the installation.

This sets the language to be used for the setup wizard.

 This does not set the language that is used in Alfresco.

4. On the **Setup - Alfresco One Platform** window, click **Next**.

5. On the **Installation Type** window, choose how you want to use the setup wizard.

There are two types of installation in the setup wizard:

Options	Description
Easy	Easy type installs Alfresco using the default options and configuration. This install type requires you to enter information in only two fields: the Alfresco install location and the administrator password. Choose this route to install Alfresco with the default environment.  If you have previously installed Alfresco and the server is running, when you run this setup wizard again, you might be prompted to enter alternative port numbers for the components and services that you install, for example, for the Tomcat application server, FTP port, and the RMI port.
Advanced	Advanced type installs Alfresco but lets you configure the server ports and service properties. You can also choose which additional components to install.

 When choosing between the **Easy** or **Advanced** installation, consider your basic requirement. If you are installing on a demo system for evaluation purpose only, Alfresco recommends that you use the **Easy** installation option. However, if you want to connect to an existing database server for Alfresco and also, want to see how the various components are being configured, use the **Advanced** installation option.

To complete the **Easy** setup wizard:

- a. Select **Easy**, and then click **Next**.
- b. On the **Installation Folder** window, click **Next** to accept the default location.
- c. On the **Admin Password** window, enter a password for the Administrator user (admin).
- d. Repeat the password, and then click **Next**.
- e. Click **Next** through the remaining windows in the setup wizard.
- f. Click **Finish** to complete the installation.

Go to the step for the **Completing the Alfresco One Platform Setup Wizard** window and launching Alfresco.

To complete the **Advanced** setup wizard, select **Advanced** and then click **Next**.

Follow the remaining steps in this task.

6. On the **Select Components** window, select the components that you want to install. Deselect the components that you do not want to install.

- Java (this is JRE only)
- PostgreSQL
- LibreOffice
- Solr4
- Alfresco Office Services

7. When you have finished selecting the components, click **Next**.

8. On the **Installation Folder** window, click **Next** to accept the default location.

For example, the default location is `C:\alfresco-one-platform`.



Alternatively, click the  icon to choose another location.

-  There is a known problem related to the use of virtual Windows drives and permission restrictions on the C drive in Windows. If you select a virtual Windows

drive, and you have selected additional components in step 6, you might receive an error message during installation:

```
09160000 An IO error was encountered during deployment of the AMP
into the WAR
```

The installation will complete, but additional components will not be added to the Alfresco installation. The error does not occur if the virtual drive is associated with a folder that is not on the C drive (that is, any other physical or virtual hard disk on the Windows machine). You have two options:

- Do not use the SUBST command for folders on the C drive (in other words, use SUBST on other drives instead)
 - Use a Virtual Hard Disk (VHD), which behaves like a physical disk, in place of the virtual Windows drive for the install. For guidance, see [Create and use a Virtual Hard Disk](#).
9. On the **Database Server Parameters** window, enter a port number for your database. Enter a suitable port number or click **Next** to accept the default of 5432.
 10. On the **Tomcat Port Configuration** window, enter the following Tomcat configuration parameters:
 - a. Web Server Domain
For example, the default is 127.0.0.1.
The URL <http://127.0.0.1:8080/share> is based on the web server domain and the Tomcat port number that you specify on the **Tomcat Port Configuration** window. The default of 127.0.0.1 can be used on this machine to verify that Alfresco is running successfully. However, it is not an externally addressable URL, which means that it is not possible for users on other machines to access this URL. To make sure that other users can access the machine where Alfresco is installed, you need to define and create a publicly addressable name.
 - b. Tomcat Server Port
For example, the default is 8080.
 - c. Tomcat Shutdown Port
For example, the default is 8005.
 - d. Tomcat SSL Port
For example, the default is 8443.
 - e. Tomcat AJP Port
For example, the default is 8009.
 11. (Optional) If you are installing the LibreOffice component, the **LibreOffice Server Port** window displays. Enter a port number on which the LibreOffice server will listen.
 12. On the **Sharded Solr installation** window, specify if you are using a sharded Solr installation, and then click **Next**.
 - ⓘ When using the installer, the templates used to create shards do not use the port specified in the installer. To set the port manually when creating a shard, see [Installing and configuring Solr shards](#).
 13. On the **Alfresco FTP Port** window, enter a port number for the Alfresco FTP server, and then click **Next**.
 14. On the **Alfresco RMI Port** window, enter a port number for Alfresco to execute remote commands, and then click **Next**.

15. On the **Admin Password** window, type a password. Repeat the password, and then click **Next**.

This sets the password for the Alfresco Administrator user account (`admin`).

 You must use ASCII characters only when setting the password using the Alfresco setup wizard. If you need to reset the password (to include non-ASCII characters) after installation, see [Changing a user's password](#) on page 459.

16. On the **Service Startup Configuration** window, select whether you want the service to start up manually or automatically.
17. On the **Warning** window, review the list of environment notifications for your installation of Alfresco installation.

This list of environment notifications is based on an evaluation of your installation environment while the setup wizard is running.

18. On the **Ready to Install** window, click **Next**.

The **Installing** window displays, showing the progress of the installation.

19. On the **Completing the Alfresco One Platform Setup Wizard** window, click **Finish**.

This window shows check boxes that determine whether you will see the Readme file, the [Getting Started](#) web page, and also whether to launch Alfresco One Platform. By default, these options are selected and will launch when you click **Finish**. If you do not want to start Alfresco at this point, deselect the **Launch Alfresco One Platform** check box.

 If you are installing the S3 connector as part of your Alfresco installation, deselect the **Launch Alfresco One Platform** check box. You must not start Alfresco before applying the S3 AMP file.

20. Click **OK** to close the Readme.

The Alfresco server starts and then Alfresco One Platform launches in your default browser.

The browser page that is displayed gives you links to resources such as the online documentation, Alfresco WebDav, Alfresco WebScripts, the Alfresco Admin Console, Alfresco Support, and CMIS 1.0 and 1.1 documentation.

 It can take several minutes to start the Alfresco server and to launch the page. Your browser opens and tries to connect to <http://127.0.0.1:8080/alfresco>.

21. To access resources such as the Alfresco Admin Console, log on as the `admin` user. Enter the password that you specified in the **Admin Password** window.

The Alfresco server is launched as a Windows service. If you did not automatically launch Alfresco at the end of the installation wizard, to start Alfresco, you need to start all the services. From the **Start** menu, select **All Programs > Alfresco One Platform > Alfresco One Platform Service > Start Alfresco One Platform service**, or from a command prompt, navigate to the Alfresco installation directory (`C:/alfresco-one-platform`) and run `servicerun START` as an administrator.

22. To fully stop Alfresco, you must stop all the services. Use the Alfresco One Platform options (see the previous step) to manage the services, or use the scripts in the installation directory to start or stop the services: `servicerun START` and `servicerun STOP`. You need administrator rights to run these commands.

Installing Alfresco on Windows using the Share Installer

The Alfresco One Share Installer for Windows installs Alfresco Share only, with its own Tomcat application server and the Share Services AMP.

 The Alfresco One Installer is recommended for most purposes. See [Installing Alfresco on Linux using the Alfresco One Installer](#) on page 12 for more information. Use the Alfresco One Share Installer only if you have a specific requirement for it; for example, if you are linking to an Alfresco repository on a different server, that has been installed using the Alfresco One Platform Installer.

1. Download the following installation file:

`alfresco-one-share-installer-5.1-win-x64.exe`

Files are available from the [Support Portal](#).

The Alfresco setup wizard is for 64-bit Windows systems.

2. Double-click the downloaded file.
3. Select the language that you wish to use for the installation.

This sets the language to be used for the setup wizard.

 This does not set the language that is used in Alfresco.

4. On the **Setup - Alfresco One Share** window, click **Next**.
5. On the **Installation type** window, choose how you want to use the setup wizard.

There are two types of installation in the setup wizard:

Options	Description
Easy	<p>Easy type installs Alfresco One Share using the default options and configuration. This install type requires you to enter locations for where you want to install Share, and which repository you want to connect to. Choose this route to install with the default environment.</p> <p> If you have previously installed Alfresco and the server is running, when you run this setup wizard again, you might be prompted to enter alternative port numbers for the components and services that you install, for example, for the Tomcat application server, FTP port, and the RMI port.</p>
Advanced	<p>Advanced type installs Alfresco One Share but asks you to specify the components that you want to install, and the Tomcat configuration parameters that you require.</p>

 When choosing between the **Easy** or **Advanced** installation, consider your basic requirement. The Alfresco One Share Installer provides a Tomcat server to run Alfresco Share, so if you have your Alfresco repository on the same machine as your Alfresco Share installation, you must use the Advanced setup, so that you can specify alternative Tomcat ports for the Share Tomcat server.

To complete the **Easy** setup wizard:

- a. Select **Easy**, and then click **Next**.
- b. On the **Installation Folder** window, click **Next** to accept the default location.
- c. Click **Next** through the remaining windows in the setup wizard.
- d. On the **Connect Alfresco One Share to the repository** window, enter the location of your Alfresco repository, in the format: `http://yourserver:port/alfresco` or accept the default setting `http://localhost:8080/alfresco`

6. On the **Ready to Install** window, click **Next**.
The **Installing** window displays, showing the progress of the installation.
7. On the **Completing the Alfresco One Share Setup Wizard** window, click **Finish**.
This window shows check boxes that determine whether you will see the Readme file, the [Getting started](#) web page, and also whether to launch Alfresco One Share. By default, these options are selected and will launch when you click **Finish**. If you do not want to start Alfresco at this point, deselect the **Launch Alfresco One Share** check box.
8. Click **Finish** to complete the installation.
9. To complete the **Advanced** setup wizard, select **Advanced** and then click **Next**.
Follow the remaining steps in this task.
10. On the **Select Components** window, select the components that you want to install.
Deselect the components that you do not want to install.
Java is the only option available to select or deselect.
11. When you have finished selecting the components, click **Next**.
12. On the **Installation Folder** window, click **Next** to accept the default location.
For example, the default location is `C:\alfresco-one-share`.

 Alternatively, click the folder icon to choose another location.
13. On the **Tomcat Port Configuration** window, enter the following Tomcat configuration parameters:
 -  The Alfresco One Share Installer provides its own Tomcat server to run Alfresco Share, so if you have your Alfresco repository on the same machine as your Alfresco Share installation, you must specify alternative Tomcat ports for the Share Tomcat server.
 - a. Web Server Domain
For example, the default is 127.0.0.1
The URL `http://127.0.0.1:8080/share` is based on the web server domain and the Tomcat port number that you specify on the **Tomcat Port Configuration** window. The default of 127.0.0.1 can be used on this machine to verify that Alfresco is running successfully. However, it is not an externally addressable URL, which means that it is not possible for users on other machines to access this URL. To make sure that other users can access the machine where Alfresco is installed, you need to define and create a publicly addressable name.
 - b. Tomcat port
For example, the default is 8080.
 - c. Tomcat Shutdown port
For example, the default is 8005.
 - d. Tomcat SSL Port
For example, the default is 8443.
 - e. Tomcat AJP Port
For example, the default is 8009.
14. On the **Connect Alfresco One Share to the repository** window, enter the location of your Alfresco repository, in the format: `http://yourserver:port/alfresco` or accept the default setting `http://localhost:8080/alfresco`
15. On the **Ready to Install** window, click **Next**.

The **Installing** window displays, showing the progress of the installation.

16. On the **Warning** window, review the list of environment notifications for your installation of Alfresco installation.

This list of environment notifications is based on an evaluation of your installation environment while the setup wizard is running.

17. On the **Completing the Alfresco One Share Setup Wizard** window, deselect the **Launch Alfresco One Share** check box, and click **Finish**.

This window shows check boxes that determine whether you will see the Readme file, the [Getting Started](#) web page, and also whether to launch Alfresco Share. By default, these options are selected and will launch when you click **Finish**. We recommend that you do not launch Share yet, because you need to complete the next steps before starting your servers.

18. Click **OK** to close the Readme.

19. Link your Share instance to a repository, by applying the Share Services AMP to your chosen Alfresco Platform instance.

 If you do not apply the Share Services AMP to the Alfresco repository, Alfresco Share will not work correctly, and when you start up Alfresco Share, you will see the message:

Alfresco is running without Share Services. See your System Administrator for more details.

- a. Navigate to the `alfresco-one-share/amps` directory, and locate the `alfresco-share-services.amp` file.
- b. Copy the `alfresco-share-services.amp` file to your Alfresco instance (`alfresco-one-platform/amps`), on the machine that hosts your Alfresco Platform repository.
- c. Use the guidance in [Installing an Alfresco Module Package](#) on page 48 to apply the AMP to your repository.
- d. Restart the Alfresco instance where you have installed the Share Services AMP to see that the changes have been applied.

20. Log on to Alfresco Share (`http://localhost:port/share`) as the `admin` user. Enter the password that you specified in the **Admin Password** window.

The Alfresco server is launched as a Windows service. If you did not automatically launch Alfresco One Share at the end of the installation wizard, to start Alfresco, you need to start all the services. From the **Start** menu, select **All Programs > Alfresco One Share > Alfresco One Share Service > Start Alfresco One Share service**, or from a command prompt, navigate to the Alfresco installation directory (`C:/alfresco-one-share`) and run `servicerun START` as an administrator.

21. Check for error messages as you open Share.

Use the information in [Troubleshooting the installation](#) on page 56 to help you.

If you need to change any settings after installation, these are stored in the `share-config-custom.xml` file. See [Configuring Share with the share-config-custom.xml file](#) for more information.

22. To fully stop Alfresco, you must stop all the services. Use the Alfresco One Share options (see the previous step) to manage the services, or use the scripts in the installation directory to start or stop the services: `servicerun START` and `servicerun STOP`. You need administrator rights to run these commands.

Installing Alfresco One in an unattended mode

Alfresco distributes binary all-in-one installers which include a setup wizard built with Bitrock technology. You can automate the installation process by running the installers in an unattended mode.

These installers contain all the dependencies that you need to quickly get Alfresco up and running. For example, the Alfresco installers install and configure all the necessary software, such as Java, Apache Tomcat, a PostgreSQL database, LibreOffice, and ImageMagick. The resulting install is optimized for demonstration and initial evaluation. The installer configures an Alfresco service on Windows and Linux (if run as root) for easier startup.

 There is no support for installing on the Mac platform in unattended mode.

To automate the installation process and customize it for your environment, you can run the install wizard in an **unattended mode** and provide an option file. The available options can be listed by calling the installer executable on the command line and passing it the `--help` flag. You can pass an option file called `install_opts` to the installer using a command, for example:

For Unix:

```
sudo ./alfresco-one-installer-201603-linux-x64.bin --optionfile install_opts
```

For Windows:

```
alfresco-one-installer-201603-win-x64.exe --optionfile install_opts
```

either as an administrator, or by clicking yes on User Account Control window that pops up.

This is an example option file, which installs most components, uses an external database, and does not install the start-up scripts:

```
mode=unattended
components=javaalfresco,alfrescowcmqs,libreofficecomponent
disable-components=postgres

# Use JDBC settings for an existing database
jdbc_url=jdbc:postgresql://localhost/alfresco
jdbc_driver=org.postgresql.Driver
jdbc_database=alfresco
jdbc_username=alfresco
jdbc_password=alfresco

# Install location
prefix=/opt/alfresco_51

alfresco_admin_password=admin

# Don't install init scripts
baseunixservice_install_as_service=0
```

The full list of options available are listed:

Table 1: Options available for the unattended installer

Option	Information
<code>--help</code>	Displays the list of valid options
<code>--version</code>	Displays the product version and information
<code>--unattendedmodeui <option></code>	Unattended Mode User Interface. Default is <code>none</code> . Options are <code>none</code> , <code>minimal</code> , <code>minimalWithDialogs</code>
<code>--optionfile <option></code>	Installation option file
<code>--debuglevel <option></code>	Debugging information. Default is 2. Options are 0, 1, 2, 3, 4.
<code>--mode <option></code>	Installation mode. Default is <code>gtk</code> . Options are <code>gtk</code> , <code>xwindow</code> , <code>text</code> , <code>unattended</code> .

Option	Information
--debugtrace <option>	Debugging file name
--enable-components <option>	Comma-separated list of components. Default is javaalfresco,postgres,alfresco,alfrescosolr4,alfrescogoogle. Options are javaalfresco,postgres,alfrescosolr,alfrescosolr4,alfrescowcmqs,alfrescogoogledocs,libreofficecomponent
--disable-components <option>	Comma-separated list of components. Default is alfrescosolr,alfrescowcmqs Options are javaalfresco,postgres,alfrescosolr,alfrescosolr4,alfrescowcmqs,alfrescogoogledocs,libreofficecomponent
--installer-language <option>	Language selection. Default is en. Options are en, fr, es, it, de, ja, nl, ru, zh_CN, no, pt_BR
--prefix <option>	Select a folder
--jdbc_url <option>	JDBC URL identifier. Default is jdbc:postgresql://localhost/alfresco
--jdbc_driver <option>	JDBC driver. Default is org.postgresql.Driver
--jdbc_database <option>	Database name. Default is alfresco
--jdbc_username <option>	User name
--jdbc_password <option>	Password
--postgres_port <option>	Database server port. Default is 5432
--tomcat_installation_type <option>	Tomcat installation type Setting tomcat_installation_type=existing prevents the tomcat binaries from being installed.
--tomcat_server_domain <option>	Web server domain
--tomcat_server_port <option>	Tomcat server port. Default is 8080
--tomcat_server_shutdown_port <option>	Tomcat shutdown port. Default is 8005
--tomcat_server_ssl_port <option>	Tomcat SSL port. Default is 8443
--tomcat_server_ajp_port <option>	Tomcat AJP port. Default is 8009
--libreoffice_port <option>	LibreOffice server port. Default is 8100
--alfresco_ftp_port <option>	FTP port. Default is 21
--alfresco.rmi.port <option>	RMI port. default is 50500
--alfresco_admin_password <option>	Admin password
--baseunixservice_install_as_service <option>	Option to install Alfresco as a service. Default is 1
--baseunixservice_script_name <option>	Service script name. Default is alfresco
--alfrescocustomstack_services_demand <option>	Option to automatically start up Alfresco custom services. Default is demand. Options are demand, auto

-  After installation, you must generate and install your own certificates to secure the installation. For more information, see [Generating Secure Keys for Solr 4 Communication](#).

Installing Alfresco manually

Use this information to manually install Alfresco One.

Prerequisites for installing Alfresco

Use this information to review your system before you manually install Alfresco.

Software requirements

Use this information to understand the required software that must be on your system for manually installing Alfresco.

Component	Recommendation
Java SE Development Kit (JDK)	Either Oracle JDK 7 or JDK 8 is required; Alfresco supports both Java 7 and Java 8. The <code>JAVA_HOME</code> environment variable must be set to the location of the JDK installation.
Application server	Alfresco runs in Tomcat.
Database	Alfresco comes preconfigured with a PostgreSQL 9.4.4 database. If you intend to use Alfresco in a production environment, you can use one of the supported databases. For the latest information on supported databases, refer to the Alfresco website. For information on configuring the database settings, refer to Configuring databases .
LibreOffice	Alfresco uses LibreOffice 4.4.5 for transforming documents from one format to another, for example, a text file to a PDF file.
ImageMagick	Alfresco uses ImageMagick to manipulate images for previewing.
GhostScript	Alfresco uses GhostScript in conjunction with ImageMagick to manipulate images for previewing.

Language support

The Alfresco interface is supported for use with a number of languages that have been through an Engineering quality assurance (QA) and linguistic testing cycle.

Alfresco is supported with the following languages:

- US English (en)
- German (de)
- Spanish (es)
- French (fr)
- Italian (it)
- Japanese (ja)
- Dutch (nl)
- Simplified Chinese (zh_CN)
- Russian (ru)
- Norwegian Bokmål (nb)
- Brazilian Portuguese (pt_BR)

If you select a language when you install Alfresco using the setup wizards, your selected language will be used for the installation instructions. To use a localized version of Alfresco, ensure that you configure the correct language in your browser settings.

The source-localized files are encoded in ASCII, and the special and accented characters are displayed using escape sequences. The source files have been renamed using the corresponding locale for each language. For example, for the French version, `site-welcome.properties` is called `sitewelcome_fr.properties`.

Although the Alfresco interface is localized, the following components have not been localized, therefore, any strings originating from these components will be displayed in English.

- SharePoint
- Web Quick Start
- LibreOffice

The following files are not localized and the error messages remain in English to ease searching for fixes to issues.

- `content-service.properties`
- `dictionary-messages.properties`
- `module-messages.properties`
- `patch-service.properties`
- `repoadmin-interpreter-help.properties`
- `schema-update.properties`
- `system-messages.properties` (partially translated)
- `tenant-interpreter-help.properties`
- `version-service.properties`
- `webclient-config-admin-interpreter-help.properties`
- `workflow-interpreter-help.properties`
- `control.properties` (in `remote-api` directory)

What to do next:

[Go to Installing Alfresco flowchart](#)

[Go to Upgrading Alfresco flowchart](#)

[Validating the architecture](#)

Environment checklist

Use this check list to validate the architecture on which Alfresco will run and also for validating the environment prior to installing Alfresco.

Validating the architecture

There are a number of steps required to validate the architecture to ensure that it meets the prerequisites for an Alfresco installation.

1. Validate that your environment is on the [Supported Platforms](#) page.
The supported platforms are the combinations of operating systems, databases, and application servers that are tested and certified for Alfresco.
2. Validate and optimize the hardware (I/O subsystems and CPU) settings.
 - a. Optimize the following I/O, in this order of priority:
 - I/O to the relational database that Alfresco is configured to use.

- I/O to the disk subsystem on which the Solr indexes are stored.
- I/O to the disk subsystem on which the content is stored.

I/O is one of the main factors that influence Alfresco performance. In each case, the goal is to minimize the latency (response time) between Alfresco and the storage system, while also maximizing bandwidth. Low latency is particularly important for database I/O, and one rudimentary test of this is to ping the database server from the Alfresco server. Round trip times greater than 1ms indicate a suboptimal network topology or configuration that will adversely impact Alfresco performance. “Jitter” (highly variable round trip times) is also of concern, as that will increase the variability of Alfresco’s performance.

Alfresco recommends that the disk throughput is greater than 200 MB/sec. On Linux, use the `hdparm` tool to measure disk throughput. The following sample output is on an SATA disk:

```
hdparm -tT /dev/sda1
/dev/sda1:
Timing cached reads: 27998 MB in 2.00 seconds = 14018.28 MB/sec
Timing buffered disk reads: 536 MB in 3.01 seconds = 178.05 MB/sec
```

Other useful tools for detecting disk I/O issues include `dd`, `seeker`, and `iozone`.

- Ensure that your system has a clock speed of greater than 2.0 Ghz.

For production use, this clock speed will ensure reasonable response times to the end user. Alfresco Enterprise 3.x and later versions have been tested on 64-bit CPU architectures, primarily because it allows the JVM to use more memory (RAM) than the earlier 32-bit CPU architecture.

- CPU clock speed is of particular concern for the Oracle UltraSPARC architecture, as some current UltraSPARC based servers ship with CPUs that have clock speeds as low as 900 Mhz, well below what is required for adequate Alfresco performance. If you intend to use Oracle servers for hosting Alfresco, ensure that all CPUs have a clock speed of at least 2.0 Ghz.

This implies that:

- An X or M class Oracle server is required, with careful CPU selection to ensure 2.0 Ghz (or better) clock speed.
- T class servers should not be used, as they do not support CPUs faster than approximately 2 Ghz. Alfresco is unable to provide specific guidance on Oracle server classes, models, or configurations, so you should talk with your Oracle reseller to confirm that minimum CPU clock speed recommendations will be met.

3. Validate the database.

- Alfresco does not provide technical support for maintaining or tuning your relational database. Ensure that your project has access to a certified database administrator (DBA) to support your Alfresco installation.

Regular maintenance and tuning of the Alfresco database is necessary. Specifically, all of the database servers that Alfresco supports require at the very least that some form of index statistics maintenance be performed at frequent, regular intervals to maintain optimal Alfresco performance.

- Index maintenance can have a severe impact on Alfresco performance while in progress, hence it needs to be discussed with your project team and scheduled appropriately.

4. Validate the operating system (OS).

- a. Ensure that your chosen OS has been officially certified for use with Alfresco (refer to the supported stacks list for details).
- b. Alfresco recommends that a 64-bit OS is used.
5. Validate and tune the JVM.

Ensure that your chosen JDK-enabled Java Virtual Machine has been officially certified for use with Alfresco (refer to the Supported Stacks list for details).

For information on configuring and tuning the JVM, refer to [Tuning the JVM](#).

Validating the environment

The following environment-specific items must be validated prior to installing Alfresco.

 An Environment Validation tool is also available that can validate most of the following requirements. This tool is available from the Alfresco [Support Portal](#). All versions are also available on the Alfresco Nexus server at <https://artifacts.alfresco.com/nexus/index.html#nexus-search;quick~alfresco-environment-validation>.

1. Validate that the host name of the server can be resolved in DNS.

This is required if Alfresco is configured in a cluster.

 Using an incorrect host name or a host name that no longer resolves to its own IP address can give an internal error, such as ObjID already in use. You can get more information about this error by adding the following line into the log4j.properties file:

```
log4j.logger.org.springframework.remoting.rmi.RmiServiceExporter=debug
```

To resolve this error, you can either:

- Validate that the IP address and the host name of the server are correctly set in the /etc/hosts file. For example, if you set the IP address as 10.20.30.40 and the host name as ip-10-20-30-40, the content of the /etc/hosts file should contain the following entry:

```
10.20.30.40 ip-10-20-30-40
```

- Specify the correct IP address in the alfresco-global.properties file as shown:

```
alfresco.rmi.services.host=10.20.30.40
```

2. Validate that the user Alfresco will run as can open sufficient file descriptors (4096 or more).
3. Validate that the ports on which Alfresco listens are available:

 The ports listed in the following table are the defaults. If you are planning to reconfigure Alfresco to use different ports, or wish to enable additional protocols (such as HTTPS, SMTP, or IMAP), update this list with those port numbers.

Protocol	Port number	Notes
FTP	TCP 21	On Unix-like operating systems that offer so-called “privileged ports”, Alfresco will normally be unable to bind to this port, unless it is run as the root user (which is not recommended). In this case, even if this port is available, Alfresco will still fail to bind to it, however for FTP services, this is a non-fatal error. The Alfresco FTP functionality will be disabled in the repository.
SMTP	TCP 25	SMTP is not enabled by default.
SMB/NetBT:	UDP 137,138	
SMB/NetBT:	TCP 139,445	On Unix-like operating systems that offer so-called “privileged ports”, Alfresco will normally be unable to bind to this port, unless it is run as the root user (which is not recommended). In this case, even if this port is available, Alfresco will still fail to bind to it, however for CIFS services, this is a non-fatal error. The Alfresco CIFS functionality will be disabled in the repository.
IMAP	TCP 143	IMAP is not enabled by default.
Tomcat Administration	TCP 8005	
HTTP	TCP 8080	
RMI	TCP 50500	

4. Validate that the installed Oracle JVM is version 1.7 or 1.8.
5. Validate that the directory in which the JVM is installed does not contain spaces.
6. Validate that the directory in which Alfresco is installed does not contain spaces.
7. Validate that the directory Alfresco will use for the repository (typically called `alf_data`) is both readable and writeable by the operating system user that the Alfresco process will run as.
8. Validate that you can connect to the database as the Alfresco database user, from the Alfresco server.
Ensure that you install the database vendor's client tools on the Alfresco server.
9. Validate that the character encoding for the Alfresco database is UTF-8.
10. (MySQL only) Validate that the storage engine for the Alfresco database is InnoDB.
11. Validate that the following third-party software is installed and the correct versions:
 - a. ImageMagick v6.2 or newer
12. (RHEL and Solaris only) Validate that LibreOffice is able to run in headless mode.

What to do next:

[Go to Installing
Alfresco flowchart](#)

[Go to Upgrading
Alfresco flowchart](#)

Supported platforms

The supported platforms are the combinations of operating systems, databases, and application servers that are tested and certified for Alfresco.

For the latest list, refer to the **Supported Platforms** page at <http://www.alfresco.com/services/subscription/supported-platforms/>.

What you need to install Alfresco

There are a number of different installation files available to you, each of which you can choose depending on what is already installed on your system.

The setup wizards install all the components you need for running Alfresco and ensure that you have all the recommended software installed and that configurations are set. When you install Alfresco using the setup wizards, it runs within an instance of the Tomcat application server. The setup wizards provide a full Alfresco install, which you can use if no Alfresco component is installed on your production environment system. See [Installing Alfresco using setup wizards](#) on page 12 for more information on the options available.

If you wish to install Alfresco within an existing Tomcat or another application server, use the Alfresco WAR file. If you use the WAR file to install Alfresco, you must install the required additional components manually.

All files are available from the Alfresco Support Portal at <http://support.alfresco.com>. Click **Online Resources > Downloads**, and select the file you require.

The following information helps you to determine what files to download and install.

Installation file	File name	Description
Alfresco One setup wizard for Windows	alfresco-one-installer-201603-win-x64.exe (64 bit)	The Alfresco setup wizard for Windows is for 64-bit systems. It is not suitable for use on 32-bit environments.
Alfresco One Share setup wizard for Windows	alfresco-one-share-installer-5.1-win-x64.exe (64 bit)	The Share Installer installs Alfresco Share only, with its own Tomcat application server and the Share Services AMP. The Alfresco One Installer is recommended for most purposes. Use the Share Installer only if you have a specific requirement for it.
Alfresco One Platform setup wizard for Windows	alfresco-one-platform-installer-5.1-win-x64.exe (64 bit)	The Platform Installer installs the Alfresco repository and all the software and components that you require for running the Alfresco platform; for example, a Tomcat application server, PostgreSQL database, JDK, LibreOffice, Solr4 and other software such as ImageMagick. It does not install Alfresco Share. The Alfresco One Installer is recommended for most purposes. Use the Platform Installer only if you have a specific requirement for it.

Installation file	File name	Description
Alfresco One setup wizard for Linux	alfresco-one-installer-201603-linux-x64.bin (64 bit)	<p>The Alfresco setup wizard for Linux is for 64-bit systems. It is not suitable for use on 32-bit environments.</p> <p>The Linux executable file is a graphical installer, but you can also run this file to install Alfresco using text mode. Text mode is a keyboard-based installation method. Run the command with the --mode text option.</p>
Alfresco One Share setup wizard for Linux	alfresco-one-share-installer-5.1-linux-x64.bin (64 bit)	<p>The Share Installer installs Alfresco Share only, with its own Tomcat application server and the Share Services AMP. The Alfresco One Installer is recommended for most purposes. Use the Share Installer only if you have a specific requirement for it.</p>
Alfresco One Platform setup wizard for Linux	alfresco-one-platform-installer-5.1-linux-x64.bin (64 bit)	<p>The Platform Installer installs the Alfresco repository and all the software and components that you require for running the Alfresco platform; for example, a Tomcat application server, PostgreSQL database, JDK, LibreOffice, Solr4 and other software such as ImageMagick. It does not install Alfresco Share. The Alfresco One Installer is recommended for most purposes. Use the Platform Installer only if you have a specific requirement for it.</p>
Alfresco One Distribution zip	alfresco-one-distribution-201603.zip	<p>Alfresco WAR files for manual install into existing application servers or for upgrades to existing Alfresco installations. This distribution zip also contains the Module Management Tool (MMT) and the sample extension files, such as alfresco-global.properties.</p>
Alfresco One EAR zip	alfresco-one-ear-distribution-201603.zip	<p>Enterprise EAR file includes the sample extension files, such as alfresco-global.properties, and also contains the alfresco-enterprise.ear file and myfaces1_1-websphere-shared-lib.zip.</p>
Alfresco One File Transfer Receiver zip	alfresco-one-file-transfer-receiver-5.1.zip	<p>File Transfer Receiver installation file</p>
Web Quick Start zip	alfresco-wcmqs-5.1.zip	<p>Web Quick Start bundle containing the AMPs for Web Quick Start and the Alfresco Web Editor.</p>

Installing Alfresco on Tomcat

For more complex Alfresco installations or if you wish to use an existing Tomcat application server, you can use the Web Archive (WAR) bundle to install Alfresco on any platform. For this type of installation, you must ensure that the required software is installed on the machine.

Use this method of installing Alfresco if you already have installed a JDK, a supported database, an application server, and the additional Alfresco components.

For information about securing Tomcat, see [Tomcat security considerations](#).

Configuring Alfresco as a Windows service

Before you start, ensure that Alfresco and a supported JDK version are installed on your Windows system.

 For more information on editing, updating, installing a Windows service, see the Apache documentation for the version of Tomcat that you are using:

- For Tomcat 8: [Updating services](#)
- For Tomcat 7: [Updating services](#)

1. To install Alfresco as a service, run the following command from a command prompt:

```
service.bat install alfresco
```

2. Open the Services panel.

3. Locate the service named **alfresco**, and then select **Start the service**.

Alfresco starts running as a Windows service.

4. To uninstall the service, run the following commands from a command prompt:

```
cd c:\alfresco\tomcat\bin
service.bat uninstall alfresco
```

Installing the Tomcat application server

Install an instance of Tomcat 7 manually and modify it to use the correct directory structure and files for Alfresco.

These instructions recommend that you name the required directories as `shared/classes` and `shared/lib` because these are the path names used within full Alfresco installations. You can substitute alternative names for these directories. The installation directory for Tomcat is represented as `<TOMCAT_HOME>`.

1. Download and install Tomcat version 7 following the instructions from <http://tomcat.apache.org>.

2. Create the directories required for an Alfresco installation:

- a. Create the `shared/classes` directory.
- b. Create the `shared/lib` directory.

3. Open the `<TOMCAT_HOME>/conf/catalina.properties` file.

4. Change the value of the `shared.loader=` property to the following:

```
shared.loader=${catalina.base}/shared/classes,${catalina.base}/shared/
lib/*.jar
```

 If you have used alternative names for the directories, you must specify these names in the `shared.loader` property.

5. Copy the JDBC drivers for the database you are using to the `lib/` directory.

6. Edit the `<TOMCAT_HOME>/conf/server.xml` file.

7. Set attributes of HTTP connectors.

Tomcat uses ISO-8859-1 character encoding when decoding URLs that are received from a browser. This can cause problems when creating, uploading, and renaming files with international characters.

By default, Tomcat uses an 8 KB header buffer size, which might not be large enough for Kerberos and NTLM authentication protocols.

Locate the `Connector` sections, and then add the `URIEncoding="UTF-8"` and `maxHttpHeaderSize="32768"` properties.

```
<Connector port="8080" protocol="HTTP/1.1" URIEncoding="UTF-8"
connectionTimeout="20000" redirectPort="8443" maxHttpHeaderSize="32768" />
```

8. Save the `server.xml` file.

When using Internet Explorer versions 7 and 8, if you try to download a document from Alfresco Share running in Tomcat with https (SSL) enabled, you might see an error message. To resolve this issue, add the following line to the `context` element in the `<TOMCAT_HOME>/conf/context.xml` file:

```
<Valve className="org.apache.catalina.authenticator.SSLAuthenticator"
securePagesWithPragma="false" />
```

Installing the Alfresco WARs

Use this method of installing if you already have installed a JDK, a supported database, an application server, and the additional Alfresco components.

The Alfresco Enterprise Distribution file is a zip containing the required WAR files, in addition to the additional commands, and configuration files for a manual installation.

1. Browse to the Alfresco Support Portal at <http://support.alfresco.com>.
2. Download the following file:
`alfresco-one-distribution-201603.zip`
3. Specify a location for the download and extract the file.

You see the following directory structure:

```
alf_data
amps
amps_share
bin
licenses
modules
solr4
web-server
```

The Distribution zip also contains the following file:

```
README.txt
```

The `/alf_data` directory contains the following directory:

```
keystore
```

This directory contains the following files:

File name	Description
<code>browser.p12</code>	The pkcs12 keystore generated from <code>ssl.keystore</code> that contains the repository private key and certificate for use in browsers, such as Firefox.
<code>CreateSSLKeystores.txt</code>	Contains instructions to create an RSA public/private key pair for the repository with a certificate that has been signed by the Alfresco Certificate Authority (CA).
<code>generate_keystores.bat</code>	Windows batch file for generating secure keys for Solr communication.
<code>generate_keystores.sh</code>	Linux script file for generating secure keys for Solr communication.

File name	Description
keystore	Secret key keystore containing the secret key used to encrypt and decrypt node properties.
keystore-passwords.properties	Contains password protecting the keystore entries.
readme.txt	Text file containing information about other files in a directory.
ssl-keystore-passwords.properties	Contains passwords for SSL keystore.
ssl-truststore-passwords.properties	Contains passwords for SSL truststore.
ssl.keystore	Repository keystore containing the repository private/public key pair and certificate.
ssl.truststore	Repository truststore containing certificates that the repository trusts.

The `/amps` directory contains the following files:

File name	Description
alfresco-share-services.amp	Share Services AMP

The `/amps_share` directory is empty, but included for any Share AMP files that you install separately.

The `/bin` directory contains these files:

File name	Description
Win32NetBIOS.dll	These DLLs handle the connection between the native CIFS server and Alfresco.
Win32NetBIOSx64.dll	
Win32Utils.dll	
Win32Utilsx64.dll	
alfresco-mmt.jar	Alfresco Module Management Tool (MMT).
alfresco-spring-encryptor.jar	Alfresco Encrypted Properties Management tool
apply_amps.bat	Windows batch file for Tomcat application server installs, used to apply all AMP files in the <code><installLocation></code> directory.
apply_amps.sh	Linux script file for Tomcat application server installs, used to apply all AMP files in the <code><installLocation></code> directory.
clean_tomcat.bat	Windows batch file for cleaning out temporary application server files from previous installations.
clean_tomcat.sh	Linux script for cleaning out temporary application server files from previous installations.

The `/licenses` directory contains the following structure:

3rd-party

This directory contains the third-party license files.

The `/modules` directory contains the following directories:

```
platform
share
```

You can put simple JAR modules in these folders, and they are loaded when Alfresco starts up. See [Simple Module](#) for more information.

The `/solr4` directory contains the following files and folders:

File or folder name	Description
<code>/alfrescoModels</code>	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.
<code>/archive-SpacesStore</code>	Configuration directory for the archive core.
<code>/conf</code>	Contains configuration files.
<code>context.xml</code>	Configuration file specifies the Solr web application context template to use when installing Solr in separate tomcat server.
<code>/lib</code>	This directory contains extra libraries that Solr 4 loads on start up. These libraries are used to communicate with Alfresco by using CMIS, Alfresco data model or Alfresco Surf Web Scripts.
<code>log4j-solr.properties</code>	Configuration file for Solr 4-specific logging.
<code>solr.xml</code>	Configuration file which specifies the cores to be used by Solr 4.
<code>/templates</code>	
<code>/workspace-SpacesStore</code>	Configuration directory for the workspace core.

The `/web-server` directory has a standard Tomcat structure, including `/shared` and `/webapps` directories.

The `/conf` directory contains Catalina repository and Share xml files.

The `/lib` directory contains the PostgreSQL JDBC jar file.

The `/shared` directory includes the Alfresco configuration files:

File name	Description
<code>/classes/alfresco-global.properties.sample</code>	The sample global properties file, which is used for Alfresco configuration properties.
<code>/classes/encrypted.properties.sample</code>	A sample encrypted properties overlay file.
<code>/classes/alfresco</code>	Contains the Alfresco directory structure for the configuration override files, including the extension, messages, and web-extension directories.

The `/webapps` directory contains these files:

File name	Description
<code>alfresco.war</code>	The Alfresco WAR file
<code>ROOT.war</code>	Application for the server root
<code>share.war</code>	The Alfresco Share WAR file
<code>solr4.war</code>	The Solr 4 WAR file

4. Move the WAR files from /webapps to the appropriate location for your application server.

For example, for Tomcat, move the WAR files to the <TOMCAT_HOME>/webapps directory. If you already have a web application that is running in the server root, see [Installing Alfresco into an existing web application](#) on page 46 for instructions on how to merge the files into your application.

-  If you are using JBoss, you must customize the `web.xml` for all WAR files to include this code fragment:

```
<context-param>
  <param-name>
    org.jboss.jbossfaces.WAR_BUNDLES_JSF_IMPL
  </param-name>
  <param-value>true</param-value>
</context-param>
```

This ensures that the JSF deployer in JBoss uses its own bundled JSF version.

5. Remove all directories in <TOMCAT_HOME>/webapps.

If you do not remove these directories, then the WAR files are not deployed when the server starts.

6. Edit the `/shared/classes/alfresco-global.properties.sample` file with your configuration settings.
7. Save the file without the `.sample` extension.
8. Move the `alfresco-global.properties` file to `<classpathRoot>`.

For example, <TOMCAT_HOME>/shared/classes.

You are now ready to install any additional software that you require. See [Installing additional software for Alfresco](#) on page 51 and [Installing Alfresco integrations](#) on page 60 for more information.

-  If you deployed previous versions of Alfresco, you must remove any temporary files created by your application server. Use the `clean_tomcat.bat` or `clean_tomcat.sh` command.
-  If you are installing the S3 connector as part of your Alfresco installation, do not start Alfresco before applying the S3 AMP file.
-  After installation, you must generate and install your own certificates to secure the installation. For more information, see [Generating Secure Keys for Solr 4 Communication](#).

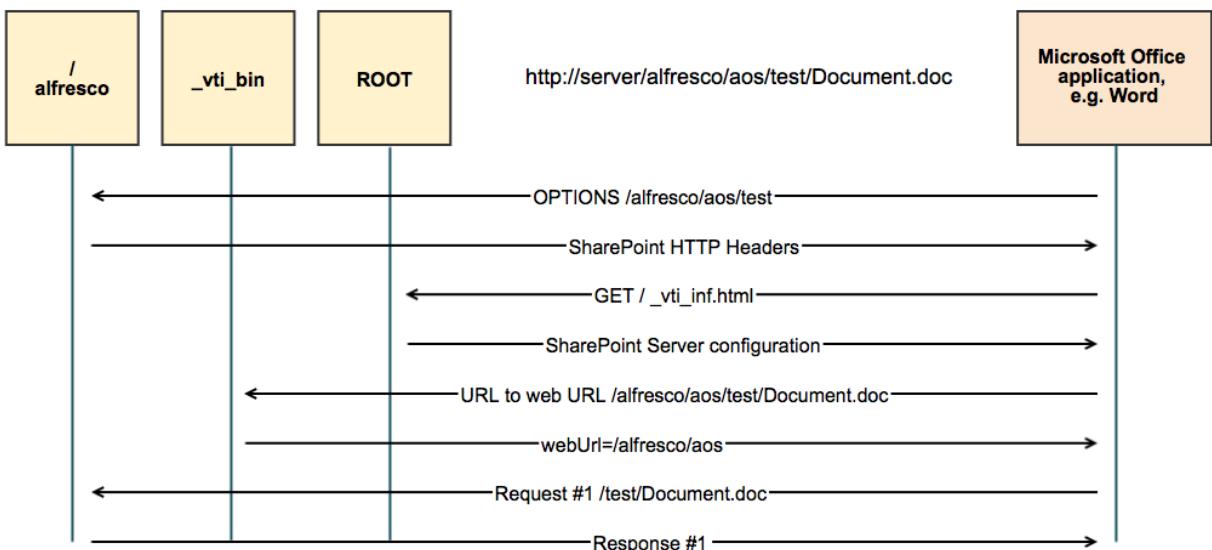
Installing Alfresco into an existing web application

The `ROOT.war` application is required to enable Alfresco Office Services (AOS). If you have a custom application that is running in the server root directory, it is important that you modify this application to enable AOS.

There are two types of requests that are sent to the server root directly by Microsoft Office and Windows:

1. A request for the `_vti_inf.html` file that contains configuration information
2. `OPTIONS` and `PROPFIND` requests

The following diagram shows the information flow between Microsoft Office and Alfresco, including interactions with the `/alfresco`, `_vti_bin` and `ROOT` applications:



1. Extract the `_vti_inf.html` file from the `<TOMCAT_HOME>/webapps/ROOT.war` archive file and add it to your web application.
 2. In your web application, modify the service that responds to requests to the server root, so that it sends `PROPFIND` and `OPTIONS` requests to the `/alfresco` application.
- If you have a `.jsp` page responding to the server root, you can add this code example to that page:

```

<%
if(request.getMethod().equals("PROPFIND") || 
request.getMethod().equals("OPTIONS"))
{
    ServletContext alfrescoContext = application.getServletContext("/alfresco");

    if( (alfrescoContext != null) && !
alfrescoContext.equals(getServletContext()) )
    {
        RequestDispatcher rd = alfrescoContext.getRequestDispatcher("/
AosResponder_ServerRoot");
        if(rd != null)
        {
            rd.forward(request, response);
            return;
        }
    }
}
%>
```

and add this import statement to the top of the `.jsp` page:

```
<%@page session="true" import="javax.servlet.ServletContext,
javax.servlet.RequestDispatcher" %>
```

If you have deployed alfresco to a different context path (something other than `/alfresco`), make sure that you edit the `application.getServletContext` value to represent this.

If you have a servlet responding to these requests, integrate the Java code from these JSP code examples into your application.

3. Depending on your application server, ensure that requests are dispatched by default between different application servers.

For Tomcat, add a file called `context.xml` to the `META-INF` directory of your web application. Here is an example of the `context.xml` file:

```
<?xml version="1.0" encoding="UTF-8"?>
<Context path="/" debug="100" privileged="true" reloadable="true"
  crossContext="true">
</Context>
```

Tailoring your Alfresco installation

When installing Alfresco, an important part of the configuration process is the removal of any unused applications. Use this information to determine any applications that you might want to remove from your Alfresco installation and how to remove them.

For example, if you want a Share-only tier, remove the Alfresco WAR file and any Solr configurations. Likewise, if you want an Alfresco-only tier, remove the Alfresco Share WAR file and any Solr configurations.

Alternatively, consider using the Alfresco One Share Installer or the Alfresco One Platform Installer instead of the Alfresco One Installer. See [Installing](#) on page 9 for more information.

Removing the `alfresco.war` file

The Alfresco WAR file is a bundle file containing the required WAR files, additional commands, configuration files, and licenses for a manual installation. Use this information to remove the `alfresco.war` file from your application.

If you want a Share-only tier in your application, you will need to delete the `alfresco.war` file from your application server. The `alfresco.war` file is stored in the `<TOMCAT-HOME>` directory.

1. Navigate to `<TOMCAT_HOME>/webapps` directory.
2. Delete the `alfresco.war` file.

You have successfully removed the `alfresco.war` file from your application server.

What to do next:

[Go to Parent topic: Remove any unwanted applications](#)

[Next: Remove `share.war` file](#)

Removing the `share.war` file

Use this information to remove the `share.war` file from your application.

If you want an Alfresco-only tier in your application, you will need to delete the `share.war` file from your application server. The `share.war` file is stored in the `<TOMCAT-HOME>` directory.

1. Navigate to `<TOMCAT_HOME>/webapps` directory.
2. Delete the `share.war` file.

You have successfully removed the `share.war` file from your application server.

What to do next:

[Go to Parent topic: Remove any unwanted applications](#)

[Next: Modify Alfresco applications](#)

Installing an Alfresco Module Package

An Alfresco Module Package (AMP) is a bundle of code, content model, content, and the directory structure that is used to distribute additional functionality for Alfresco. Use the Module Management Tool (MMT) to install and manage AMP files. You can install an AMP in an Alfresco WAR using the MMT, or by using the `apply_amps` tool.

The MMT is included in the Alfresco installers, and it is also available as a separate JAR file from the Alfresco Distribution zip (`alfresco-one-distribution-201603.zip`), in the zip's / `bin` directory. Place the `/bin` directory and its contents in the same location that is used by the Alfresco installer (`<installdir>/bin`).

1. Browse to the `/bin` directory:

- (Windows) `C:\Alfresco\bin`
- (Linux) `/opt/alfresco/bin`

2. Run the `apply_amps` application to apply all AMP files that are in the `amps` and `amps_share` directories:

- For Windows, navigate to the `bin` directory and double click `apply_amps`.
- For Linux, enter the command:

```
bin/apply_amps.sh
```

3. Alternatively, to install individual AMP files, use MMT:

```
java -jar alfresco-mmt.jar install <AMPFileLocation> <WARFileLocation>
[options]
```

Where:

Option	Description
<code><AMPFileLocation></code>	The location of the AMP file that you want to install.
<code><WARFileLocation></code>	The location of the WAR file for your Alfresco installation.
<code>-verbose</code>	Install command [options]. Enables detailed output containing what is being updated and to where it is being copied.
<code>-directory</code>	Install command [options]. Indicates that the AMP file location specified is a directory. All AMP files found in the directory and its sub directories are installed.
<code>-force</code>	Install command [options]. Forces installation of AMP regardless of currently installed module version.
<code>-preview</code>	Install command [options]. Previews installation of AMP without modifying WAR file. It reports the modifications that will occur on the WAR without making any physical changes, for example, the changes that will update existing files. It is good practice to use this option before installing the AMP.
<code>-nobackup</code>	Indicates that the WAR will not be backed up before the AMP is installed.

This command installs the files found in the AMP into the Alfresco WAR. If the module represented by the AMP is already installed and the installing AMP is of a higher release version, then the files for the older version are removed from the WAR and replaced with the newer files.

The following commands show examples of how to install the `example-amp.amp`, and assumes that the AMP file is in the same directory as the WAR file:

```
java -jar alfresco-mmt.jar install example-amp.amp alfresco.war -preview
```

Review the modification to check the changes that will update any existing files.

The following example will install the AMP file:

```
java -jar alfresco-mmt.jar install example-amp.amp alfresco.war -verbose
```

The modified Alfresco WAR can then be redeployed back into your application server.

On restarting the application server, the console will show that the custom class was initialized during startup.

4. Verify that the AMP is installed using the MMT `list` command. For example:

```
java -jar alfresco-mmt.jar list <WARfileLocation>
```

This command provides a detailed listing of all the modules currently installed in the WAR file specified.

When the repository is next started, the installed module configuration will be detected, and the repository will be bootstrapped to include the new module functionality and data.

It is not recommended that you overwrite an existing file in an AMP, however it is sometimes necessary. The MMT makes a backup copy of the updated file and stores it in the WAR. When an update of the module occurs and the old files are removed, this backup will be restored prior to the installation of the new files. Problems can occur if multiple installed modules modify the same existing file. In these cases, a manual restore might be necessary if recovery to an existing state is required.

Some application servers (notably Tomcat) do not always fully clean up their temporary working files, and this can interfere with successful installation of an AMP file. To remedy this situation, it is recommended that you delete (or move) the Tomcat `work` and `temp` directories while Tomcat is shut down.

Viewing module packages

Alfresco Module Packages are used to package Alfresco customizations and extensions for deployment. Use the **Module Browser** page to view all the AMPs that have been applied to Alfresco.

Click **Admin Tools** then **Module Browser**.

The **Module Browser** page shows a list of all the module packages that are either pre-configured in an out-of-the-box Alfresco installation or applied by the user, along with the description and version number.

Home My Files Shared Files Sites ▾ Tasks



Module Browser

▼ Tools

- Application
- Category Manager
- Module Browser
- Node Browser
- Tag Manager
- Model Manager
- Sites Manager

Repository

- Replication Jobs

Users and Groups

- Groups
- Users

Title

Alfresco / Google Docs

Installing additional software for Alfresco

Installing LibreOffice

In Alfresco, you can transform a document from one format to another, for example, a text file to a PDF file. To have access to these transformation facilities in Alfresco, you must install LibreOffice. This is optional, and can be done any time after Alfresco is installed.

1. Browse to the LibreOffice download site: [LibreOffice download site](#)
 2. Download the latest (stable) version of LibreOffice for your platform.
 3. When prompted, specify a download destination.
 4. Browse to the location of your downloaded file, and install the application.
 5. Change the installation directory to:
 - (Windows) `c:\Alfresco\LibreOffice`
 - (Linux) `/opt/alfresco/LibreOffice`
- If you are installing LibreOffice on Linux, you also need a number of libraries to be installed. See [Installing Linux libraries manually](#) on page 15 for more information.
6. Modify the `ooo.exe=` property in the `<classpathRoot>/alfresco-global.properties` file to point to the LibreOffice application `libreoffice.app`.
-  For Windows, set the path using the `\` separator or use the forward slash / Unix path separator. For example: `c:\\\\Alfresco\\\\LibreOffice\\\\libreoffice.app` or `c:/Alfresco/LibreOffice/libreoffice.app`.
7. If the Alfresco server is running, stop and restart the server.

Installing ImageMagick

To enable image manipulation in Alfresco, you must install and configure ImageMagick. Alfresco uses ImageMagick to manipulate images for previewing.

1. Verify that ImageMagick, Ghostscript, and Ghostscript fonts are already installed on your system.
Use the `ImageMagick convert` command to check that you have the right software installed on your machine. This command is usually located in `/usr/bin`: `install Image`.
 2. If the ImageMagick and Ghostscript software is not available on your system, download and install the appropriate package for your platform.
To download ImageMagick, browse to [ImageMagick download website](#).
To download Ghostscript, browse to [Ghostscript download website](#).
 In next steps you will make changes to the Alfresco application configuration files to enable the manually installed ImageMagick application. These steps can only be performed after Alfresco has been installed.
3. Browse to the `<classpathRoot>` directory. See [System paths](#) for more information.
 4. Open the `alfresco-global.properties` file.
 5. Modify the ImageMagick properties to point to the ImageMagick root directory:

Table 2: ImageMagick properties

Property	Description
<code>img.root</code>	On Windows, set this property to <code>img.root=C:\\ImageMagick</code> On Linux, set this property to <code>img.root=/ImageMagick</code>  Do not include a slash (/) at the end of the path. For example, <code>/ImageMagick/</code>
<code>img.dyn</code>	On Windows, set this property to <code>img.dyn=\${img.root}\\\\lib</code> On Linux, set this property to <code>img.dyn=\${img.root}/lib</code>

Property	Description
img.exe	On Windows, set this property to img.exe= \${img.root}\\\convert.exe On Linux, set this property to img.exe= \${img.root}/bin/convert
img.coders	On Windows, set this property to img.coders=\${img.root}\\\modules\\coders On Linux, set this property to img.coders= \${img.root}/modules/coders
img.config	On Windows, set this property to img.config= \${img.root}\\\config On Linux, set this property to img.config= \${img.root}/config

 Test that you are able to convert a PDF using the command convert filename.pdf[0] filename.png.

Installing Ghostscript

ImageMagick uses Ghostscript to render Postscript and PDF files, as well as formats where a translator to Postscript is available. ImageMagick will also use Ghostscript fonts to support the standard set of Postscript fonts.

-  Where a step indicates to update the alfresco-global.properties file, this makes changes to the Alfresco application configuration files to enable the manually installed GhostScript application. This step can only be performed after Alfresco has been installed.
-  The Ghostscript executable file is entirely platform-specific.
- For Windows:
 - Download Ghostscript (32 bit) from the [Ghostscript Downloads Page](#).
 - Browse to the location of your downloaded file and install the application.
 - Update the img.gslib property in the alfresco-global.properties file as shown:

```
img.gslib = <GhostScript_installation_dir>/lib
```
- For Linux:

From Source:

- Download Ghostscript from the [Ghostscript Downloads Page](#).
- Make sure that building toolchains specific to your OS version are installed (for example, gcc, make or any related packages).
- Run the following commands to install Ghostscript:

```
./configure
make
make install
```

This installs Ghostscript at /usr/local/.

- Add the following to the alfresco-global.properties file:

```
img.gslib = /usr/local/share/ghostscript/<version>/lib
```

From repositories/CD:

- Make sure that your OS is connected to the online repository/repositories or the installation media is accessible by the package manager.
- Open your distribution's terminal program.
- Based on your Linux distribution, type the following command in the terminal.

RHEL:

```
yum install ghostscript
```

SLES:

```
zypper install ghostscript
```

Ubuntu:

```
apt-get install ghostscript
```

- Wait as the files are downloaded and installed onto your system. A list of added files will show when the process is complete.
- For Solaris:

From Source:

- Download Ghostscript from the [Ghostscript Downloads Page](#).
- Make sure that building toolchains specific to your OS version are installed (for example, gcc, make or any related packages).
- Run the following commands to install Ghostscript:

```
./configure
make
make install
```

This installs Ghostscript at /usr/local/.

- Add the following to the alfresco-global.properties file:

```
img.gslib = /usr/local/share/ghostscript/<version>/lib
```

From repositories/CD:

- Make sure that your OS is connected to the online repository/repositories or the installation media is accessible by the package manager.
- Open your distribution's terminal program.
- Type the following command in the terminal:

```
pkgadd SUNWgscr
```

- Wait as the files are downloaded and installed onto your system. A list of added files will show when the process is complete.

Installing TinyMCE language packs

Translations in Alfresco use the language packs supplied in the default install. The supported language packs are: German (de), English (en), Spanish (es), French (fr), Italian (it), Japanese (ja), and Dutch (nl). The language used switches according to the browser locale. Ensure that your browser is set up to view the relevant locale, which ensures that the special characters display correctly in your installed instance.

The source-localized files are encoded in ASCII, and the special and accented characters are displayed using escape sequences. The source files have been renamed using the corresponding locale for each language. For example, site-welcome.properties is called sitewelcome_fr.properties for the French version.

If you wish to use a translation that is not supplied with Alfresco, then you must add the appropriate TinyMCE language pack for the translation to work correctly.

If you installed Alfresco using one of the setup wizards, the default language packs are already installed. If you have installed Alfresco manually, you must install the supported language pack manually.

1. Browse to the TinyMCE website: [TinyMCE](#).
2. Download the required TinyMCE language pack.



The next step makes configuration changes to the Share application to configure the additional language packs for TinyMCE. This step can only be performed after Alfresco has been installed.

3. Unpack the language file to:
`<TOMCAT_HOME>/webapps/share/modules/editors/tiny_mce/langs.`
4. Ensure that the browser cache is cleared or refresh the page.

Testing the Alfresco installation

Installation testing checks that Alfresco is successfully installed and it is working as expected after installation.

Some of the points that need to be checked prior to testing your Alfresco installation are:

- Verify the pre-requisites needed to install Alfresco.
- Verify that after successfully installing Alfresco, the application is working as per the specification document & meet user needs.
- Upon uninstalling Alfresco, check that all previously installed files and registry entries are removed as expected.

Test and familiarize after installing Alfresco

Here are some tips to familiarize yourself with Alfresco.



Alfresco recommends that you create a test site for testing purpose and put all your test data in that site.

- Can you login to Alfresco using your user name and password. See [Logging in](#).
- Can you create a collaboration site. See [Creating your first Collaboration site](#).
- Can you add new users to the collaboration site. See [Adding users to a site](#).
- Can you add pages to the collaboration site. See [Adding pages to a site](#).
- Can you add content to a site library. See [Adding content items](#).
- Can you copy or move content from its current location to another folder or any other site. See [Copying content](#) and [Moving content](#).
- Can you update/ edit content. See [Updating content](#).
- Can you manage permissions for a user or a group for accessing content. See [Managing content permissions](#).
- Can you add a new rule to a folder in the site library and check if it works. See [Adding a new rule](#).
- Can you edit the new rule. See [Editing a rule](#).
- Can you schedule events, such as meeting, for your team. See [Scheduling events](#).

-  After you have finished testing, remember to delete the test site or test data in order to clear your database. Alternatively, if you have made any configuration changes, it is recommended that you [Uninstall Alfresco](#) and then [Reinstall Alfresco](#) to get a clean database.

Test and familiarize after configuring Alfresco

Tips to help you test your Alfresco customizations:

-  Alfresco recommends that you create one or two test sites for testing purposes and put all your test data in those sites. After finishing the tests, you can delete the test sites in order to clear your database.
- If you have installed Records Management, test whether you can use it search records in the Alfresco repository. See [Using Records Management](#).
 - If you have installed and configured the Alfresco Kofax Release script, test whether you can use it to capture and publish content. See [Installing and configuring Alfresco Kofax Release script](#).

Test and familiarize after installing Alfresco in a clustered environment

Here are some of the tips to help you test your Alfresco customizations.

-  Alfresco recommends that you create a test site for testing purpose and put all your test data in that site.
- Check that the Alfresco application server is running.
 - Can you login to Alfresco using your user name and password. See [Logging in](#).
 - Check that various Alfresco components are communicating with each other.
 - For a clustered installation, check if one node is down, check if the request is forwarded to the next available node.
 - Check if clustering is working properly by running the [cluster validation tool](#) in the Admin Console.
 - Check if you are using a clustering-enabled license.
 - Change the cluster-related properties in the `alfresco-global.properties` file, and check if all the nodes are up and running.
-  After you have finished testing, remember to delete the test site or test data in order to clear your database. Alternatively, if you have made any configuration changes, it is recommended that you [Uninstall Alfresco](#) and then [Reinstall Alfresco](#) to get a clean database.

Troubleshooting the installation

Follow these tips if you see error messages when using the Alfresco One Share Installer.

-  The Alfresco One Installer is recommended for most purposes. See [Installing Alfresco on Linux using the Alfresco One Installer](#) on page 12 for more information. These tips help troubleshoot problems found in Alfresco Share, when you have used the Alfresco One Share Installer (see [Installing Alfresco on Linux using the Share Installer](#) on page 19 for more information). Use this installer only if you have a specific requirement for it.
1. Start Alfresco and log on to Alfresco Share (`http://localhost:port/share`) as the **admin** user. Enter the password that you specified in [Installing Alfresco on Linux using the Share Installer](#) on page 19 in the **Admin Password** window.
 2. Check for error messages as you start Share.

3. If you see the following message:

Alfresco is running without Share Services. See your System Administrator for more details.

check the Admin Console to determine which AMP files have been installed, and their versions. It might be that either you have not installed the Share Services AMP in your Alfresco repository (see [Installing Alfresco on Linux using the Share Installer](#) on page 19 or [Installing Alfresco on Windows using the Share Installer](#) on page 29), or the version of the AMP that you have installed is not correct.

4. If you see the following message:

Invalid MANIFEST.MF: Share Specification-Version is missing, are you using the valid MANIFEST.MF supplied with the Share.war?

check that you have not deleted or changed the MANIFEST.MF file. The MANIFEST.MF file shipped with the Share.war is required for validation, and Alfresco Share will not work correctly if this cannot be read.

If you are using a Maven WAR build, this will override the shipped MANIFEST.MF file. When you unpack your WAR file, you will need to specify `unpack-dependencies` explicitly, for example:

```
<plugin>
  <artifactId>maven-dependency-plugin</artifactId>
  <executions>
    <execution>
      <id>unpack</id>
      <phase>generate-sources</phase>
      <goals>
        <goal>unpack-dependencies</goal>
      </goals>
      <configuration>
        <includeTypes>war</includeTypes>
        <includeGroupIds>org.alfresco</includeGroupIds>
        <includeArtifactIds>share</includeArtifactIds>
        <includes>META-INF/MANIFEST.MF</includes>
      </configuration>
    </execution>
  </executions>
</plugin>

<plugin>
  <artifactId>maven-war-plugin</artifactId>
  <configuration>
    <archive>
      <addMavenDescriptor>false</addMavenDescriptor>
      <manifestFile>${project.build.directory} /dependency/META-INF/
MANIFEST.MF</manifestFile>
    </archive>
    <webResources>
      </webResources>
    </configuration>
  </plugin>
```

Uninstalling Alfresco

Use this information to uninstall Alfresco, or any Alfresco AMP files.

Uninstalling an AMP file

Use the Module Management Tool (MMT) to uninstall one or more AMP files.

The MMT program, `alfresco-mmt.jar`, is available in the `bin` directory of the Alfresco installation. MMT uninstalls an AMP file by removing content from the `alfresco.war` and

share.war files. For more information on the tool, see [Using the Module Management Tool \(MMT\)](#).

MMT is a command line tool. The syntax for uninstalling an AMP file using MMT is:

```
java -jar bin/alfresco-mmt.jar uninstall <ModuleId> <WARFileLocation>
```

-  The `apply_amps` command does not uninstall AMP files (even if you remove the AMP files manually from the `amps` and `amps_share` directories). Use `apply_amps` to install AMP files only.

For each integration, there is always at least one AMP file to remove from the `alfresco.war` and `share.war` files. AMP files that are applied to `alfresco.war` usually reside in the `amps` directory, and AMP files that are applied to `share.war` usually reside in the `amps_share` directory.

1. Open a command prompt and change into the root directory of your Alfresco installation.
2. Check for the presence of the module you wish to delete by typing in the following command:

```
java -jar bin/alfresco-mmt.jar list tomcat/webapps/alfresco.war
```

for `alfresco.war` AMP files, and

```
java -jar bin/alfresco-mmt.jar list tomcat/webapps/share.war
```

for `share.war` AMP files.

This displays a list of installed modules. Make a note of the module ID of the module you wish to uninstall, for example, `org.alfresco.integrations.google.docs` in the `amps` directory, and `org.alfresco.integrations.share.google.docs` in the `amps_share` directory.

3. Uninstall the module by entering the following command:

```
java -jar bin/alfresco-mmt.jar  
uninstall org.alfresco.integrations.google.docs tomcat/webapps/  
alfresco.war
```

and

```
java -jar bin/alfresco-mmt.jar  
uninstall org.alfresco.integrations.share.google.docs tomcat/webapps/  
share.war
```

4. You can check that the AMP files have been removed by rerunning the command:

```
java -jar bin/alfresco-mmt.jar list tomcat/webapps/alfresco.war
```

and

```
java -jar bin/alfresco-mmt.jar list tomcat/webapps/share.war
```

5. Delete the `tomcat/webapps/alfresco` and `tomcat/webapps/share` folders in the Alfresco installation directory.
Deleting these directories forces Tomcat to read the edited WAR files when Alfresco is restarted.
6. Restart Alfresco to see your changes.

Uninstalling Alfresco on Linux

Use this information to uninstall Alfresco on Linux.

The uninstalling steps below are based on the following assumptions:

- Alfresco is installed using one of the Alfresco setup wizards.
 - Alfresco is installed at `/opt/alfresco-`.
 - Alfresco service is created.
1. Navigate to the directory where Alfresco is installed.
For example:
 - For Alfresco One, this is `/opt/alfresco-one`
 - For Alfresco Community, this is `/opt/alfresco-community`
 - For Alfresco One Platform, this is `/opt/alfresco-one-platform`
 - For Alfresco One Share, this is `/opt/alfresco-one-share`
 2. Launch the uninstall binary file.
You will see the **Question** window.
 3. Click **Yes** to continue with uninstalling Alfresco.
If you do not want to uninstall Alfresco, click **No**.
The **Setup** window displays the progress bar for uninstalling Alfresco.
 4. After the uninstall process is complete, click **OK** to close the window.

The uninstall process is complete. The Alfresco installation directory has been successfully removed from your system.

What to do next:

[Go to Installing
Alfresco flowchart](#)

[Go to Upgrading
Alfresco flowchart](#)

Next:

[Installing Alfresco using setup wizards](#)

Uninstalling Alfresco on Windows

Use this information to uninstall Alfresco on Windows.

1. Stop the Alfresco server, as specified in [Stopping the Alfresco server](#) on page 451.
2. From the **Start Menu > Control Panel > Uninstall a program**, double-click the Alfresco installation that you want to remove. For example, **Alfresco One**, **Alfresco Community**, **Alfresco One Platform** or **Alfresco One Share**
The Alfresco wizard prompts you to confirm the uninstallation of Alfresco and all its modules.
3. Click **Yes**.

The Alfresco uninstall window appears and the installation directory and its contents are removed.

What to do next:

[Go to Installing Alfresco flowchart](#)

[Go to Upgrading Alfresco flowchart](#)

Next:

[Installing Alfresco using setup wizards](#)

Installing Alfresco integrations

Use this information to install any components or modules that integrate Alfresco to other applications.

Module or Integration	Additional information	Link
Alfresco Analytics	Paid add-on module that requires additional software. Install with AMPs.	Installing and configuring Alfresco Analytics
Alfresco EMC Centera Connector	Paid add-on module. Install with an AMP file in Alfresco and install the EMC Centera SDK. Requires additional software.	Installing and configuring the Alfresco EMC Centera Connector on page 60
Alfresco Kofax Release script	Install with an AMP in Alfresco and binary files in Kofax.	Installing and configuring Alfresco Kofax Release script on page 70
Alfresco Office Services	AMP installed as part of the standard Alfresco install.	Installing and configuring Alfresco Office Services on page 79
Alfresco Outlook Integration	Paid add-on module. Install with AMPs in Alfresco and a zip file in Microsoft Outlook.	External link: Installing and configuring Alfresco Outlook Integration
Alfresco Media Management	Paid add-on module that requires additional software. Install with AMPs.	External link: Installing and configuring Alfresco Media Management
Alfresco Records Management	Paid add-on module. Install with AMPs.	External link: Installing Alfresco Records Management
Alfresco S3 Connector	Paid add-on module. Install with an AMP file.	Installing and configuring Alfresco S3 Connector on page 87
Alfresco Transformation Server	Paid add-on module. Install with AMP files in Alfresco and and an MSI package on the standalone Transformation Server.	Installing and configuring the Alfresco Transformation Server on page 90
Alfresco Web Quick Start	Can be installed as part of the standard Alfresco install (Advanced option).	Installing and configuring Alfresco Web Quick Start on page 97
Google Docs Integration	AMP installed as part of the standard Alfresco install.	Installing and configuring Google Docs integration on page 107

Installing and configuring the Alfresco EMC Centera Connector

Use this information to install and configure the Alfresco EMC Centera Connector module. It also lists the prerequisites for setting up the EMC Centera environment on Windows and Linux platforms.

The Alfresco EMC Centera Connector module addresses the Centera store directly through its native API.

The module uses a series of properties to control the way that you access the store. A feature of this module allows you to set retention policies, such as, preventing content from being deleted for a period of time (for example, retaining invoices for seven years).

The Alfresco EMC Centera Connector module can be applied to Alfresco Enterprise 4.2.0 or later.

Software prerequisites for the Alfresco EMC Centera Connector

To use the Alfresco EMC Centera Connector module, ensure that you have the prerequisite software installed on your machine.

The software is available from the EMC Community Network. Register and log in to the Community Network before you try to access the required software.

Download the following prerequisite software:

- (Windows installation only) Microsoft Visual C++ 2005 Service Pack 1 Redistributable Package
To download, go to <http://www.microsoft.com/en-us/download/default.aspx>.
- EMC Centera® SDK 3.3
To download, go to <https://developer-content.emc.com/downloads/centera/download.htm> (log in to the site required).
- EMC Centera SDK 3.3 and Community Tools
To download, go to <https://community.emc.com/docs/DOC-2393>.
- Server details and .pea files
To download, go to <https://community.emc.com/docs/DOC-1038>.

Setting up the Alfresco EMC Centera Connector environment on Windows

Create the environment on Windows for checking the EMC Centera connection.

1. Download and install the Microsoft Visual C++ 2005 Service Pack 1 Redistributable Package.
2. Download and extract EMC Centera® SDK to a suitable directory, for example, c:\centera.
 - Centera_SDK_Windows_2000-5.0-Win32Dev8.zip for 32-bit systems
 - Centera_SDK_Windows_2000-5.0-Win64Dev8.zip for 64-bit systems

On 32-bit systems, the subdirectory structure of the c:\centera directory includes the following directories:

```
docs
include
lib
lib32
sdk_samples
```

On 64-bit systems, the subdirectory structure of the c:\centera directory includes the following directories:

```
docs
include
lib
lib64
sdk_samples
```

3. Download the Centera .pea file.

For example, c2armtesting.pea.

4. Move the c2armtesting.pea file to the Centera C:\centera directory.
5. Download and extract EMC Centera® SDK and Community Tools to any directory.

The structure of the C:\centera directory is similar to the following example (for 32-bit systems):

```
10.01.2014 17:55 <DIR> .
10.01.2014 17:55 <DIR> ..
11.12.2013 16:25 2 470 c2armtesting.pea
10.01.2014 17:41 docs
10.01.2014 17:41 include
10.01.2014 17:41 lib
10.01.2014 17:41 lib32
10.01.2014 17:41 sdk_samples
1 File(s) 2 470 bytes
7 Dir(s) 49 088 593 920 bytes free
```

The structure of the C:\centera directory is similar to the following example (for 64-bit systems):

```
10.01.2014 17:55 <DIR> .
10.01.2014 17:55 <DIR> ..
11.12.2013 16:25 2 470 c2armtesting.pea
10.01.2014 17:37 docs
10.01.2014 17:37 include
10.01.2014 17:37 lib
10.01.2014 17:37 lib64
10.01.2014 17:37 sdk_samples
1 File(s) 2 470 bytes
7 Dir(s) 49 088 593 920 bytes free
```

The structure of the C:\centera\lib32 directory is similar to the following example:

```
10.01.2014 17:41 <DIR> .
10.01.2014 17:41 <DIR> ..
29.08.2012 17:33 774 144 FPCore.dll
29.08.2012 17:33 610 304 FPLibrary.dll
29.08.2012 17:33 610 948 FPLibrary.lib
29.08.2012 17:33 323 584 fpos32.dll
29.08.2012 17:33 2 011 136 fpparser.dll
29.08.2012 17:33 184 320 FPStreams.dll
29.08.2012 17:33 438 272 FPUtils.dll
29.08.2012 17:33 184 320 FPXML.dll
10.01.2014 17:41 <DIR> lib
29.08.2012 17:33 262 144 pai_module.dll
9 File(s) 5 399 172 bytes
3 Dir(s) 49 088 593 920 bytes free
```

The structure of the C:\centera\lib64 directory is similar to the following example:

```
10.01.2014 17:37 <DIR> .
10.01.2014 17:37 <DIR> ..
29.08.2012 17:34 983 552 FPCore.dll
29.08.2012 17:34 690 688 FPLibrary.dll
29.08.2012 17:34 616 178 FPLibrary.lib
29.08.2012 17:34 412 160 fpos64.dll
29.08.2012 17:34 2 919 424 fpparser.dll
29.08.2012 17:34 165 888 FPStreams.dll
29.08.2012 17:34 483 840 FPUtils.dll
29.08.2012 17:34 168 960 FPXML.dll
10.01.2014 17:37 <DIR> lib
29.08.2012 17:34 63 488 pai_module.dll
9 File(s) 6 504 178 bytes
3 Dir(s) 49 088 593 920 bytes free
```

Setting up the Alfresco EMC Centera Connector environment on Linux

Create the environment on Linux for checking the EMC Centera connection.

1. Download and extract EMC Centera® SDK (Centera_SDK_Linux-gcc3.3.tgz), for example, to /opt.

A subdirectory structure of the /opt/Centera_SDK directory includes the following directories:

```
total 20
drwxr-xr-x. 4 root root 4096 Jan 10 21:32 docs
drwxr-xr-x. 2 root root 4096 Jan 10 21:32 include
drwxr-xr-x. 2 root root 4096 Aug 30 2012 install
drwxr-xr-x. 2 root root 4096 Jan 10 21:32 lib
drwxr-xr-x. 13 root root 4096 Sep 14 2006 sdk_samples
```

2. Install the EMC Centera® SDK using the following commands:

```
cd /opt/Centera_SDK/install
./install
```

The default installation directory is /usr/local/Centera_SDK.

3. Download the Centera .pea file.

For example, c2armtesting.pea.

4. Move the c2armtesting.pea file to the Centera /usr/local/Centera_SDK directory.
5. Download and extract EMC Centera® SDK and Community Tools to any directory.

The structure of the /usr/local/Centera_SDK directory is similar to the following example:

```
total 12
-rw-r--r--. 1 root root 2470 Dec 11 16:25 c2armtesting.pea
drwxr-xr-x. 2 root root 4096 Dec 19 22:51 include
drwxr-xr-x. 4 root root 4096 Dec 19 22:51 lib
```

The structure of the /usr/local/Centera_SDK/lib/32 directory is similar to the following example:

```
total 6316
lrwxrwxrwx. 1 root root      52 Dec 19 22:51 libFPCore32.so -> /usr/local/
Centera_SDK/lib/32/libFPCore32.so.3.3.719
-rwxr-xr-x. 1 root root 1063484 Dec 19 22:51 libFPCore32.so.3.3.719
lrwxrwxrwx. 1 root root      44 Dec 19 22:51 libFPCore.so -> /usr/local/
Centera_SDK/lib/32/libFPCore32.so
lrwxrwxrwx. 1 root root      55 Dec 19 22:51 libFPLibrary32.so -> /usr/local/
Centera_SDK/lib/32/libFPLibrary32.so.3.3.719
-rwxr-xr-x. 1 root root 643603 Dec 19 22:51 libFPLibrary32.so.3.3.719
lrwxrwxrwx. 1 root root      47 Dec 19 22:51 libFPLibrary.so -> /usr/local/
Centera_SDK/lib/32/libFPLibrary32.so
lrwxrwxrwx. 1 root root      53 Dec 19 22:51 libFPParser32.so -> /usr/local/
Centera_SDK/lib/32/libFPParser32.so.3.3.50
-rwxr-xr-x. 1 root root 3800245 Dec 19 22:51 libFPParser32.so.3.3.50
lrwxrwxrwx. 1 root root      46 Dec 19 22:51 libFPParser.so -> /usr/local/
Centera_SDK/lib/32/libFPParser32.so
lrwxrwxrwx. 1 root root      55 Dec 19 22:51 libFPStreams32.so -> /usr/local/
Centera_SDK/lib/32/libFPStreams32.so.3.3.719
-rwxr-xr-x. 1 root root 121784 Dec 19 22:51 libFPStreams32.so.3.3.719
lrwxrwxrwx. 1 root root      47 Dec 19 22:51 libFPStreams.so -> /usr/local/
Centera_SDK/lib/32/libFPStreams32.so
lrwxrwxrwx. 1 root root      53 Dec 19 22:51 libFPUtils32.so -> /usr/local/
Centera_SDK/lib/32/libFPUtils32.so.3.3.719
-rwxr-xr-x. 1 root root 648376 Dec 19 22:51 libFPUtils32.so.3.3.719
lrwxrwxrwx. 1 root root      45 Dec 19 22:51 libFPUtils.so -> /usr/local/
Centera_SDK/lib/32/libFPUtils32.so
lrwxrwxrwx. 1 root root      51 Dec 19 22:51 libFPXML32.so -> /usr/local/
Centera_SDK/lib/32/libFPXML32.so.3.3.719
-rwxr-xr-x. 1 root root 129647 Dec 19 22:51 libFPXML32.so.3.3.719
lrwxrwxrwx. 1 root root      43 Dec 19 22:51 libFPXML.so -> /usr/local/
Centera_SDK/lib/32/libFPXML32.so
lrwxrwxrwx. 1 root root      56 Dec 19 22:51 libPAI_module32.so -> /usr/local/
Centera_SDK/lib/32/libPAI_module32.so.3.3.100
```

```
-rwxr-xr-x. 1 root root 49036 Dec 19 22:51 libPAI_module32.so.3.3.100
lrwxrwxrwx. 1 root root 48 Dec 19 22:51 libPAI_module.so -> /usr/local/
Centera_SDK/lib/32/libPAI_module32.so
```

The structure of the `/usr/local/Centera_SDK/lib/64` directory is similar to the following example:

```
total 6736
lrwxrwxrwx. 1 root root 52 Dec 19 22:51 libFPCore64.so -> /usr/local/
Centera_SDK/lib/64/libFPCore64.so.3.3.719
-rwxr-xr-x. 1 root root 1098829 Dec 19 22:51 libFPCore64.so.3.3.719
lrwxrwxrwx. 1 root root 44 Dec 19 22:51 libFPCore.so -> /usr/local/
Centera_SDK/lib/64/libFPCore64.so
lrwxrwxrwx. 1 root root 55 Dec 19 22:51 libFPLibrary64.so -> /usr/local/
Centera_SDK/lib/64/libFPLibrary64.so.3.3.719
-rwxr-xr-x. 1 root root 671881 Dec 19 22:51 libFPLibrary64.so.3.3.719
lrwxrwxrwx. 1 root root 47 Dec 19 22:51 libFPLibrary.so -> /usr/local/
Centera_SDK/lib/64/libFPLibrary64.so
lrwxrwxrwx. 1 root root 53 Dec 19 22:51 libFPParser64.so -> /usr/local/
Centera_SDK/lib/64/libFPParser64.so.3.3.50
-rwxr-xr-x. 1 root root 4061679 Dec 19 22:51 libFPParser64.so.3.3.50
lrwxrwxrwx. 1 root root 46 Dec 19 22:51 libFPParser.so -> /usr/local/
Centera_SDK/lib/64/libFPParser64.so
lrwxrwxrwx. 1 root root 55 Dec 19 22:51 libFPStreams64.so -> /usr/local/
Centera_SDK/lib/64/libFPStreams64.so.3.3.719
-rwxr-xr-x. 1 root root 134962 Dec 19 22:51 libFPStreams64.so.3.3.719
lrwxrwxrwx. 1 root root 47 Dec 19 22:51 libFPStreams.so -> /usr/local/
Centera_SDK/lib/64/libFPStreams64.so
lrwxrwxrwx. 1 root root 53 Dec 19 22:51 libFPUtils64.so -> /usr/local/
Centera_SDK/lib/64/libFPUtils64.so.3.3.719
-rwxr-xr-x. 1 root root 713762 Dec 19 22:51 libFPUtils64.so.3.3.719
lrwxrwxrwx. 1 root root 45 Dec 19 22:51 libFPUtils.so -> /usr/local/
Centera_SDK/lib/64/libFPUtils64.so
lrwxrwxrwx. 1 root root 51 Dec 19 22:51 libFPXML64.so -> /usr/local/
Centera_SDK/lib/64/libFPXML64.so.3.3.719
-rwxr-xr-x. 1 root root 151395 Dec 19 22:51 libFPXML64.so.3.3.719
lrwxrwxrwx. 1 root root 43 Dec 19 22:51 libFPXML.so -> /usr/local/
Centera_SDK/lib/64/libFPXML64.so
lrwxrwxrwx. 1 root root 56 Dec 19 22:51 libPAI_module64.so -> /usr/local/
Centera_SDK/lib/64/libPAI_module64.so.3.3.100
-rwxr-xr-x. 1 root root 52961 Dec 19 22:51 libPAI_module64.so.3.3.100
lrwxrwxrwx. 1 root root 48 Dec 19 22:51 libPAI_module.so -> /usr/local/
Centera_SDK/lib/64/libPAI_module64.so
```

Configuring the EMC Centera connection

You can configure the Alfresco EMC Centera Connector module to alter the behavior of the connection.

1. Open the `<classpathRoot>/alfresco-global.properties` file.
2. Add the `centera.url` property.

For example:

```
centera.url=168.159.214.24?c:/centera/c2armtesting.pea
```

The `centera.url` property specifies the details of the Centera server. For example, in this case, it specifies the IP address 168.159.214.24.

The property also includes the location of the Centera `c2armtesting.pea` file.

For example, `C:/centera/c2armtesting.pea` or `/usr/local/Centera_SDK/c2armtesting.pea`.

3. Set any additional properties to alter the way that the Alfresco EMC Centera Connector behaves.

There are various additional properties that can be set to control the Alfresco EMC Centera Connector module. For example, the retention period for storing content is controlled using the `xam.archive.retentionPeriodDays=1` property.

 The sample `alfresco-global.properties` file supplied in the Alfresco EMC Centera Connector AMP provides example settings and values.

4. Save the `alfresco-global.properties` file.
5. Ensure that Java can find the Centera libraries.

On Windows, set the `Path` environment variable.

- a. Open the **Control Panel\All Control Panel Items\System**.
- b. Select **Advanced System Settings > Advanced > Environment Variables**.
- c. In the **System Variables** section, modify the existing `Path` environment variable by adding the path to the Centera libs.

For example:

```
Path=c:\centera\lib64
```

On Linux, set the `PATH` and `LD_LIBRARY_PATH` environment variables.

For example:

```
export PATH=$PATH:/usr/local/Centera_SDK/lib/64
export LD_LIBRARY_PATH=/usr/local/Centera_SDK/lib/64
```

Alfresco EMC Centera Connector module properties

The following properties can be set for the Alfresco EMC Centera Connector module.

Set these properties in the `alfresco-global.properties` file.

centera.url=168.159.214.24?c:/centera/c2armtesting.pea

Specifies the full Centera connection string.

xam.archive.storeName=xamArchive

Specifies the name of the XAM store that will be used by the `xam:archive` behavior.

xam.archive.retentionPeriodDays=0

Specifies the number of days to retain a XAM document. Use `0` to ignore; `>0` days to retain.

Alfresco can be configured to allow deletes in conflict to the Centera enforce retention periods. A retention period is the time that a C-Clip and its underlying blobs must be stored on an EMC Centera before an application is allowed to delete them. According to configuration, `retentionPeriod` is set to 1 day. If you switch the server date to 1-2 days ahead on the Alfresco side, this will not result in the expiry of retention periods in the Centera cluster. For this reason, delete is not permitted. You should not change the date/time but wait until this period finished. Change the `xam.archive.retentionPeriodDays` to be not be greater than `system.content.orphanProtectDays`. This will prevent the cleaner from deleting non-expired Centera binary content.

xam.archive.addLock=true

Specifies whether to add the `cm:lockable` aspect automatically. Set to `true` to apply a lock when the aspect is added; set to `false` to not apply a lock

xam.archive.cronExpression=0 0 21 * * ?

Specifies a cron expression for the XAM archive cleaner job.

xam.archive.bindingPropertiesPattern=vnd\\.com\\.alfresco\\..*

Specifies the pattern for all binding properties. Any property (full property name at time of writing) that does not match will be written as non-binding. For example, `vnd\\.com\\alfresco\\..*` will match all properties prefixed with `vnd.com.alfresco`. Refer to <http://download.oracle.com/javase/tutorial/essential/regex/>, also <http://download.oracle.com/javase/6/docs/api/>.

xam.archive.app.db=\${db.url}

The XAM well-known properties, which will be automatically populated.

xam.archive.globalPropertiesPrefix=vnd.com.alfresco.

xam.archive.globalPropertiesToWrite=xam.archive.app.vendor, xam.archive.app.name, xam.archive.app.version, xam.archive.app.db

The list of global properties to add to the XSet (comma-separated). For example, `${xam.archive.globalPropertiesPrefix}xam.archive.app.vendor`. This can be a list of any value that can be set in the `alfresco-global.properties` file but you should import any required properties using variable replacement to get consistent naming.

xam.archive.contentFieldName=\${xam.archive.globalPropertiesPrefix}content

Specifies the name of the property against which to store content.

xam.archive.nodePropertiesPrefix=xam.archive.node.

xam.archive.nodePropertiesToWrite=sys:ref, sys:path, cm:name, cm:created, cm:creator, cm:owners

The list of node properties to add to the XSet (comma-separated, namespace-prefixes). For example, `${xam.archive.globalPropertiesPrefix}${xam.archive.nodePropertiesPrefix}cm:name`. Properties that are not present on the node are ignored. Implicit properties are generated and can be listed, for example, `sys:ref, sys:path`.

xam.archive.forceBackgroundStoreMove

Specifies whether to move content to the XAM store immediately or as a background job. The aspect `xam.archivemodel:archivePending` is added to the document, pending the move to the XAM store. Set to false to move the content binaries to XAM as soon as the retention date is set. Set to true to move the content when the scheduled job runs. The default value for this property is false.

Testing the EMC Centera connection

The JCASScript tool is provided with the EMC Centera® SDK and Community Tools.

Use the JCASScript tool to connect to the XAM server using the `centera.url` property that you specified in the `alfresco-global.properties` file.

1. Start the JCASScript tool using the following command:

```
java -jar JCASScript.jar
```

2. Enter the following command to connect to the XAM server:

```
poolOpen 168.159.214.24?/usr/local/Centera_SDK/c2armtesting.pea
```

An example of the output is as follows:

```
CASScript>poolOpen 168.159.214.24?/usr/local/Centera_SDK/c2armtesting.pea
Attempting to connect to: 168.159.214.24?/usr/local/Centera_SDK/c2armtesting.pea
Connected to: 168.159.214.24?/usr/local/Centera_SDK/c2armtesting.pea

CASPool Properties:
  Connection String:          168.159.214.24?/usr/local/
  Centera_SDK/c2armtesting.pea
  Cluster Time:               2014.01.10 06:25:31 GMT
  Buffer Size:                16384
```

Prefetch Buffer Size:	32768
Connection Timeout:	120000
Multi-Cluster Failover Enabled:	True
Collision Avoidance Enabled:	False

Installing the Alfresco EMC Centera Connector module

These steps describe how to install the Alfresco EMC Centera Connector module to an instance of Alfresco.

The Alfresco EMC Centera Connector is packaged as an Alfresco Module Package (AMP) file.

1. Browse to the Alfresco Support Portal.

<http://support.alfresco.com>

2. Download the `alfresco-centera-connector-1.0.5-22.amp` file.
3. Use the Module Management Tool (MMT) to install the AMP.

```
java -jar <installLocation>\bin\alfresco-mmt.jar install
<installLocation>\amps\alfresco-centera-connector-1.0.5-22.amp
<installLocation>\tomcat\webapps\alfresco.war
```

If your Alfresco installation is running within the Tomcat application server, you can use the `<installLocation>\bin\apply_amps` command to apply all AMP files that are located in the `<installLocation>\amps` directory.

4. Restart the Alfresco server.

Working with the Alfresco EMC Centera Connector module

Test that the Alfresco EMC Centera Connector module is working correctly with Alfresco.

1. Enable `DEBUG` logging for the Alfresco EMC Centera Connector components.

For example:

```
log4j.logger.org.alfresco.enterprise.repo.content.centera=DEBUG
log4j.logger.org.alfresco.enterprise.repo.centera=DEBUG
```

2. Add the `xam:archived` aspect to the `share-config-custom.xml` file.

For example:

```
<alfresco-config>

    <config evaluator="node-type" condition="cm:content">

        <forms>
            <form>
                <!-- 2 column template -->

                <edit-form />

                <field-visibility>

                    <!-- aspect: cm:storeSelector -->

                    <show id="cm:storeName" />

                    <!-- aspect: xam:archive -->
                    <show id="xam:dateArchived" for-mode="view" />
                    <show id="xam:retainUntil" for-mode="view" />
                    <show id="cm:content" for-mode="view" />
                </field-visibility>

                <appearance>
                    <!-- Store Selector -->
                    <field id="cm:storeName" label="Store Name"
                        description="Content Store Name" />
                </appearance>
            </form>
        </forms>
    </config>
</alfresco-config>
```

```

        <set id="xam-archive" appearance="bordered-panel"
label="XAM Archived" />
            <field id="xam:dateArchived" label="XAM Date Archived"
set="xam-archive" />
                <field id="xam:retainUntil" label="XAM Retain Until Date"
set="xam-archive" />
                    </appearance>
                </form>
            </forms>
        </config>

<config evaluator="string-compare" condition="DocumentLibrary">
    <aspects>
        <visible>
            <aspect name="xam:archive" label="XAM Archive" />
        </visible>
    </aspects>
</config>
</alfresco-config>
```

3. View the metadata for the document.

The new store is shown as **xamArchive** and the **retainedUntil** date is set.

4. Copy the ClipID, and then open the C-Clip using the JCASScript tool.

For example:

```
CASScript>clipopen EQM2GC012MC77e72B24N2MMFU59G418ACSAIE70BAS340TN3E1JJL

Clip Properties:

Name:          untitled
Creation Date: 2013.11.27 01:35:09 GMT
Size:          13474
Number of Tags: 1
Number of Blobs: 1
Retention Class:
Retention Seconds: 86396
Modified:       False
EBR Enabled :  False
Retention Hold: False
```

- a. Check that the retention period was set.

```
CASScript>clipattribs

Number of attributes: 17

Name: creation.poolid           Value: 861673fa-1dd2-11b2-b535-
b66ede9133c1-7
Name: retention.period          Value: 86396
Name: sdk.version                Value: 3.3.718
Name: modification.poolid       Value: 861673fa-1dd2-11b2-b535-
b66ede9133c1-7
Name: type                       Value: Standard
Name: name                        Value: untitled
Name: creation.date              Value: 2013.11.27 13:35:09 GMT
Name: modification.date          Value: 2013.11.27 13:35:12 GMT
Name: creation.profile           Value: armtesting
Name: modification.profile       Value: armtesting
Name: numfiles                   Value: 1
Name: totalsize                  Value: 13474
Name: refid                      Value: E5S2HABU8PRRBAS340TN3E1JJL
Name: clusterid                 Value: 25c57a54-1dd2-11b2-b87c-
ce625a7031f2
Name: prev.clip                  Value:
Name: clip.naming.scheme         Value: MD5
Name: numtags                     Value: 1
```

- b. Check that the node and application properties have been copied over.

Select the first tag of the opened C-Clip. For example:

```
CASScript>tagfirst
CASTag Properties:

Name: com.alfresco.content
Has Blob: True
Blob Size: 13474
Number of Attributes: 10
Has Parent: False
Has Next Sibling: False
Has Child: False
```

Display all the attributes. For example:

```
CASScript>tagattribs
Number of attributes: 10

Name: modified-date Value: 1385553402696
Name: com.alfresco.xam.archive.node.sys:ref Value:
workspace://SpacesStore/51bba786-184b-4d7b-8b2a-da90875e5b16
Name: com.alfresco.xam.archive.app.name Value: Main
Repository
Name: com.alfresco.xam.archive.node.cm:created Value:
2013-11-27T15:56:27.011+04:00
Name: com.alfresco.xam.archive.app.version Value: 4.2.0
(28)
Name: com.alfresco.xam.archive.app.db
Value: jdbc:mysql://localhost:3306/alfresco?
useUnicode=yes&characterEncoding=UTF-8
Name: com.alfresco.xam.archive.node.sys:path Value: /
app:company_home/st:sites/cm:test/cm:documentLibrary/cm: abc.txt
Name: com.alfresco.xam.archive.node.cm:creator Value: admin
Name: com.alfresco.xam.archive.node.cm:name Value: abc.txt
Name: com.alfresco.xam.archive.app.vendor Value: Alfresco
Software
```

- c. Type `tagClose` to close current tag.
 - d. Type `clipClose` to close current C-Clip.
 - e. Type `poolClose` to close current connection to EMC Centera pool.
5. Test the folder hierarchy.
- a. Create a folder containing several files and folders,
 - b. Apply the `xam:archived` aspect to the top-level folder.
 - c. Check that the aspect has been applied to the entire hierarchy.
 - d. Choose one of the files in the hierarchy and check through for a single file from step 1.

Setting up the CenteraContentStore as the main store

To set up the CenteraContentStore to be the main store, it is recommended that you also configure the primary store as a CachingContentStore.

See [Configuring CachingContentStore](#) for more information.

This setup relates to new content and cannot be applied retrospectively, unless all content is moved from the file system to Centera.

1. Create `xam-custom-context.xml` file in the `<extension>` directory.
For example, `<installLocation>/tomcat/shared/classes/alfresco/extension`.
2. Copy the `org_alfresco_module_centera_centeraContentStore` bean from `<installLocation>/tomcat/webapps/alfresco/WEB-INF/classes/alfresco/module/org_alfresco_module_xamconnector/module-context.xml` file.

For example:

```
?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="org_alfresco_module_centera_centeraContentStore"
class="org.alfresco.enterprise.repo.content.centera.CenteraContentStore"
init-method="init">

        <property name="readOnly" value="false" />
        <property name="centeraConnection"
ref="org_alfresco_module_centera_centeraConnection"/>
            <property name="contentFieldName"
value="${xam.archive.contentFieldName}" />

    </bean>

</beans>
```

3. Paste the bean in to the newly created `xam-custom-context.xml` file.
4. Change the bean id to `fileContentStore`.

For example:

```
<?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="fileContentStore"
class="org.alfresco.enterprise.repo.content.centera.CenteraContentStore"
init-method="init">

        <property name="readOnly" value="false" />
        <property name="centeraConnection"
ref="org_alfresco_module_centera_centeraConnection"/>
            <property name="contentFieldName"
value="${xam.archive.contentFieldName}" />

    </bean>

</beans>
```

5. Add the following property to `alfresco-global.properties` file.

```
xam.archive.contentFieldName=com.alfresco.content
```

6. Start Alfresco server.

Installing and configuring Alfresco Kofax Release script

Use this information to install, configure, and use the Alfresco Kofax Release script.

Integrating Kofax and Alfresco provides complete content management support including the capture, management, and publishing of content. Kofax Capture captures content from various sources, typically through scanning and OCR. The captured information is then released to Alfresco to be managed in an ad-hoc manner or using pre-defined business processes.

The Kofax architecture provides a plug-in architecture for deploying a Kofax Release script that is responsible for mapping and transferring the information captured by Kofax to the destination application or database.

The Alfresco Kofax Release script comprises a Release script plug-in that is installed within the Kofax Capture application and a set of Alfresco web scripts installed on the Alfresco server.

The Alfresco Kofax Release script provides the following capabilities:

- Alfresco server connection (connection URL, user name, password)
- Destination folder in which to store the captured documents (folders can be automatically created based on index field values)
- Mapping of Kofax Capture indexing information and files to Alfresco properties
 - Support for Alfresco types, sub-types, and aspects, and their associated properties
 - Mapping of Kofax Image (TIFF), Text (OCR), or PDF files to Alfresco content properties
- Automatic versioning, overwrite, and error handling for existing documents

System requirements and prerequisites

There are a number of system requirements for the Alfresco Kofax Release script.

The Alfresco Kofax Release script has the following prerequisites:

- Alfresco Version 3.1 or higher
- Kofax 9 Service Pack 3, or Kofax 10

You need to have a working knowledge of Kofax Capture and Alfresco.

Installation and advanced configuration requires experience with Alfresco Module Packages (AMPs) and defining Alfresco models. For more information on Kofax, refer to the Kofax Capture documentation.

Installing Kofax Release script

Installing the Alfresco Kofax Release script is a two-part process.

The installation process involves the following steps:

1. Installation of the Alfresco Kofax Release script Alfresco Module Package (AMP) file using the Alfresco Module Management Tool.
2. Installation of the Alfresco Kofax Release script binaries in your Kofax Capture installation.

Installing the Alfresco Kofax Release script AMP

The following describes how to install the Alfresco Kofax Release script AMP file (`alfresco-kofax-integration-2.0.0-13.amp`) on your Alfresco server.

1. Shut down your Alfresco server.
2. Browse to the Alfresco Support Portal.
<http://support.alfresco.com>
3. Download the `alfresco-kofax-integration-2.0.0-13.zip` file.
4. Extract the zip file into a relevant directory.
5. Move or copy the `alfresco-kofax-integration-2.0.0-13.amp` file to the `amps` directory in your Alfresco installation.
 - (Windows) `c:\Alfresco\amps`
 - (Linux) `/opt/alfresco/amps`
6. From the command line, browse to the Alfresco `bin` directory.

Installing

- (Windows) c:\Alfresco\bin
 - (Linux) /opt/alfresco/bin
7. Install the Alfresco Kofax AMP using the Module Management Tool.
- For more information on MMT, see [Installing an Alfresco Module Package](#) on page 48.
- For Windows:
- `java -jar alfresco-mmt.jar install c:\Alfresco\amps\alfresco-kofax-integration-2.0.0-13.amp c:\Alfresco\tomcat\webapps\alfresco.war`
- For Linux:
- `java -jar alfresco-mmt.jar install /opt/alfresco/amps/alfresco-kofax-integration-2.0.0-13.amp /opt/alfresco/tomcat/webapps/alfresco.war`
-  Alternatively for Tomcat, you can run the `apply_amps.bat` command in the root Alfresco directory to install the `alfresco-kofax-integration-2.0.0-13.amp`. This batch file applies all the AMPs that are located in the `amps` directory.
8. Remove your existing expanded Alfresco web application directory to allow updates to be picked up when the server restarts.
- (Windows) c:\Alfresco\tomcat\webapps\alfresco
 - (Linux) /opt/alfresco/tomcat/webapps/alfresco

Installing the Alfresco Kofax Capture Release script binaries

The following steps describe how to install the binaries required to set up and configure the Kofax Release script in your Kofax Capture installation.

-  You must have Windows administrator privileges to install Kofax Capture Release script binaries. If you do not have administrator rights, you might encounter errors and the script might fail to install.
1. Unzip the `alfresco-kofax-integration-2.0.0-13.zip` file to your Kofax Capture bin directory.
For example, (Windows) c:\Program Files\Kofax\Capture\bin
 2. Start the Kofax Capture Administration Module.
 3. In the Kofax Administration module, click **Tools > Release Script Manager**.
 4. From the **Release Script Manager** dialog box, click **Add** and then browse to the directory of the unzipped files.
 5. Select `Alfresco.Kofax.Release.inf`, and click **Open**.
 6. Click **OK** to register the release script.
 7. Close the open dialog boxes to complete the process.

Configuring the Alfresco Kofax Release script

Use these instructions to set up the Alfresco Kofax Release script. These instructions assume you are familiar with Kofax Capture and have created a Kofax Capture batch class. For information on setting up batch classes in Kofax Capture, refer to the Kofax Capture documentation.

In Kofax Capture, release scripts are associated with document classes. The script is configured to define where and how the documents will be released, including:

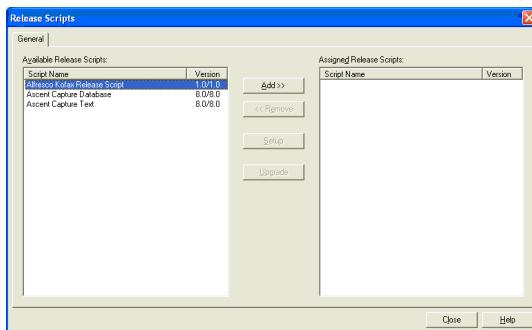
- URL to connect to your Alfresco server

- Alfresco user name and password used to create the documents in Alfresco
- Location in the Alfresco repository where documents will be released
- Options for handing existing documents, such as Overwrite, Version, Release to Default Folder, or Report an Error
- Alfresco document type
- Mapping between the Alfresco properties (including those based on type and configured aspects), and the Kofax indexing fields to be populated by the release script

Associating the Alfresco Kofax Release script with a document class

Once you have set up a batch class with an associated document class in Kofax Capture, you can associate a Release script with the batches document class. As part of this process, you are prompted to enter the connection details for your Alfresco server.

1. Start the Kofax Capture Administration Module.
2. Select the **Batch class** tab from the **Definitions** panel, and right-click the applicable document class. (Expand the Batch class item to select associated document classes.)
3. From the **Context** menu, select **Release Scripts**.



The **Release Scripts** dialog box displays, listing all available release scripts. Available release scripts are those that are registered with Kofax Capture.

4. From the **Release Scripts** dialog box, select the Alfresco Kofax Release Script, and click **Add**.



The **Login** dialog box displays.

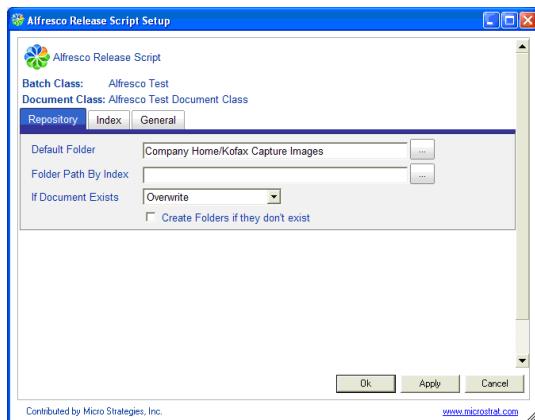
5. Enter your Alfresco server URL, user name, and password.
6. Click **Login**.

Alfresco Kofax Release script configuration tabs

The Kofax Release script is configured using three main tabs. The following information describes each of the configuration tabs and the options available.

Repository tab

The **Repository** tab is used to configure where documents are stored in the Alfresco repository and how existing documents are handled.



The Repository tab has the following options:

Default Folder

Defines the root Alfresco space in which documents will be created.

- The user that connects to Alfresco must have permission to create documents in this space.

Folder Path by Index

Allows the folder path to be dynamically generated based on indexing values. Substitute Alfresco property name(s) to be used as part of the folder path.

For example, the following will store all documents with the same `Invoice Date` property in folders according to the invoice date:

```
Company Home/Invoices/[Invoice Date]
```

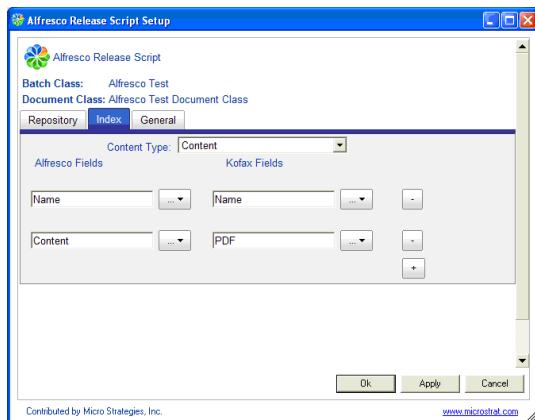
If Document Exists

A document already exists if a document of the same name already exists in the folder in which the document is being released. The following defines how the Release script will handle existing documents.

- **Overwrite:** Replaces the document with the one being currently released.
- **Version:** Creates a new version of the document.
- **Release To Default Folder:** If the folder path specified in the **Folder Path By Index** field has an existing document with the same name, the document will be put into the location specified in the **Default Folder** field.
- **Throw Error:** The release fails with the error `Duplicate child name not allowed`.
- **Create Folders if they don't exist:** If selected, this will automatically create folders that do not exist as defined by the previous **Folder Path by Index** settings. If this is not selected, and the folder path(s) do not exist, an error will occur and the document will fail.

Index tab

The **Index** tab defines the Alfresco document type used for released documents, and the mappings between Kofax index fields and Alfresco properties.



Each row defines the mapping between an Alfresco property and a Kofax indexing field. The **Content Type** and **Alfresco Fields** values available can be controlled through configuration.

Content type

The Alfresco content type that will be used for documents created by the Release script. It can be a custom content type or content.

Alfresco Fields

Use the list to pick Alfresco properties based on the available types and aspects that will be populated with Kofax Capture index data.

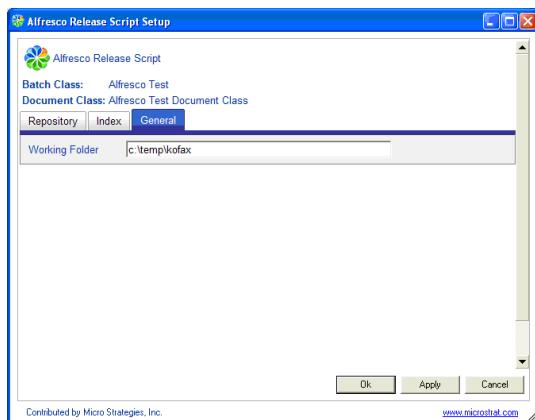
Kofax fields

Use the list to pick the Kofax Capture field to map to the Alfresco property. The **Text Constant** field can provide a fixed text value for the field.

- ! You must define an Alfresco **Name** field and an Alfresco **Content** field, as shown in the previous figure. The **Content** field is used to store the image file, such as Image (TIF), PDF, or Text (OCR).

General tab

The **General** tab defines the working folder used by Kofax Capture for temporary file storage during the release process.



Working Folder

Set this to a folder where the user running the script has write access on the local Kofax Capture machine.

Publishing a batch class

After you select all your batch class settings, you must publish your batch class before you can use it.

The publishing process checks the integrity of the settings in your batch class and makes the batch class available for use. If problems are found with any of the settings, error and warning messages will display, along with the recommended actions for fixing the problems.

If you edit your batch class, you must publish your batch class again before your changes can be used. Your changes will not be applied to batches created before the new publication date.

1. Start the Kofax Capture Administration module to display the main screen.
2. Select the **Batch class** tab from the **Definitions** panel, and right-click the applicable batch class.
3. From the **Context** menu, select **Publish**.
4. From the **Publish** window, select your batch class and click **Publish**.

Kofax Capture will check all of your batch class settings and display the results in the **Results** box.

If no problems are detected, the message `Publishing successful` displays. If a problem is detected, a warning or error message displays along with recommended actions to resolve the problem. Perform the recommended actions, and then try to publish the batch class again.

5. Run some sample batches through the system to test the operation of the release script.

After successfully publishing, you can create batches based on your batch class. As your batches flow through your Kofax Capture system, they will be routed from module to module. The modules that are used to process a batch, and the order that processing occurs, are specified as part of the batch class definition for the batch.

Refer to the Kofax Capture Help for more information about batch classes.

Releasing batches

The Kofax Capture Release module will process batches based on the settings of the associated batch classes. This module is responsible for releasing documents, as well as index data using the attributes defined during release setup.

The Kofax Capture Release module usually runs as an unattended module on a Windows workstation, periodically polling the module for available batches. It can be configured to run during off-hours to avoid any impact to the throughput of Kofax Capture and/or the network system.

1. Start the Kofax Capture Release module by selecting **Start > Programs > Kofax Capture > Release**.
All batches queued for release will be processed after initiation of the module.
Once your batch is released, it will be removed from Kofax Capture. If any documents or pages are rejected, the batch will be routed to the Kofax Capture Quality Control module.
2. To exit the Kofax Capture Release module, select **Batch > Exit** from the module menu bar.

Refer to the Kofax Capture Help for more information about releasing batches.

Advanced configuration: custom types, aspects, and properties

By default, the Release Setup web script (`\service\kofax\releasesetup`) displays all types, aspects, and their associated properties available in your Alfresco repository.

The Release script can be configured to limit this list to only show only those values that are applicable to your use case. A web script configuration file is used to define the items to be displayed.

The Release script configuration file uses a structure similar to that used by the model definitions themselves. Add the types and/or aspects and the relevant properties to the `releasescript.get.config.xml` file to define the options you want available. See the sample configuration provided for examples.

-  For information on defining your own model for types and aspects, refer to the Alfresco Wiki page **Data Dictionary Guide**.
1. Locate the `releasesetup.get.config.xml.sample` file. For Tomcat this will be located at:
`tomcat\WEBINF\classes\alfresco\templates\webscripts\com\microstrat\kofax\releasesetup.get.config.xml.sample`
 -  This is the default location used by the Tomcat application server. The location of the file can vary depending on the application server used by your Alfresco installation.
 2. Rename `releasesetup.get.config.xml.sample` to `releasesetup.get.config.xml`.
 3. Reload your web script using the Web Script Index page as follows:
 - a. Go to `http://YOURHOST:8080/alfresco/service/index`.
 - b. Click **Refresh Web Scripts**.
 4. Open the Release Script **Index** tab.

This will now only allow selection of types, aspects, and properties as defined in the configuration file.

If an aspect exists with properties and these properties are to be mapped from Kofax to Alfresco, then all properties for this aspect must be populated in the batch process. If certain properties are omitted from the mapping within the release script set up, then when documents are released, the unmapped properties are overwritten with empty strings.

For example, you have an aspect with properties assigned to the default content model and have a document with this aspect assigned. When using Kofax integration, when the document `exists` `version` option is set, all aspect properties must be mapped and populated in the batch process, otherwise all unmapped properties are overwritten with empty strings (blanked out). This is because in the document `exists` case, the `version` option uses checkout/check in functionality, which means that the aspect as a whole is repopulated with empty strings if they are unmapped.

The workarounds are:

- Map all properties in the batch process
- Split out your aspects so that unmapped properties are part of different aspects

Removing the Alfresco Kofax Release script

The following steps describe how to remove the Alfresco Kofax Release script from your Kofax installation.

1. Start the Kofax Capture Administration module.
2. Remove the Alfresco Kofax Release script from any document classes using the script:
 - a. Right-click the applicable document class. (Expand the batch class item to select associated document classes.)
 - b. From the **Context** menu, select **Release Scripts**.
 - c. From the **Release Scripts** dialog box, select the Alfresco Kofax Release Script from the list of **Assigned Release Scripts**, and click **Remove**.
3. Repeat step 2 for all document classes using the Alfresco Kofax Release script.
4. In the Kofax Administration module, click **Tools > Release Script Manager**.
5. Select **Alfresco Kofax Release Script**, and click **Remove**.

6. To remove the installation files, manually delete the following files from your Kofax Capture bin directory.

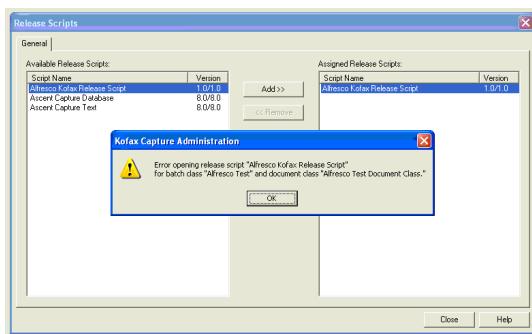
- Alfresco.Kofax.Release.Core.dll
- Alfresco.Kofax.Release.Core.Logging.xml
- Alfresco.Kofax.Release.Core.xml
- Alfresco.Kofax.Release.inf
- Alfresco.Kofax.Release.WebScripts.dll
- Antlr.runtime.dll
- Common.Logging.dll
- Jayrock.Json.dll
- log4net.dll
- Spring.Core.dll

Troubleshooting the Kofax Release script

Use this information to troubleshoot the Kofax Release script.

Error adding the Alfresco Kofax Release script to a document class

If you see an error message “Error opening release script “Alfresco Kofax Release Script” when adding the script to a document class, it is an indication that you might not have copied the binaries to your Kofax Capture bin directory.



Ensure that the following files are in the bin directory:

- Alfresco.Kofax.Release.Core.dll
- Alfresco.Kofax.Release.Core.Logging.xml
- Alfresco.Kofax.Release.Core.xml
- Alfresco.Kofax.Release.inf
- Alfresco.Kofax.Release.WebScripts.dll
- Antlr.runtime.dll
- Common.Logging.dll
- Jayrock.Json.dll
- log4net.dll
- Spring.Core.dll

Release Error: [Release Script Returned -1. Your release script may need to be re-installed.]

This is a generic Kofax error. The most likely cause is that an invalid working folder has been specified when setting up the release.

Ensure that you have entered a valid folder path in the **Working Folder** field on the **General** tab.

Other causes of this error include missing dependencies in the installation. Check that you have installed all the required files the `bin` directory.

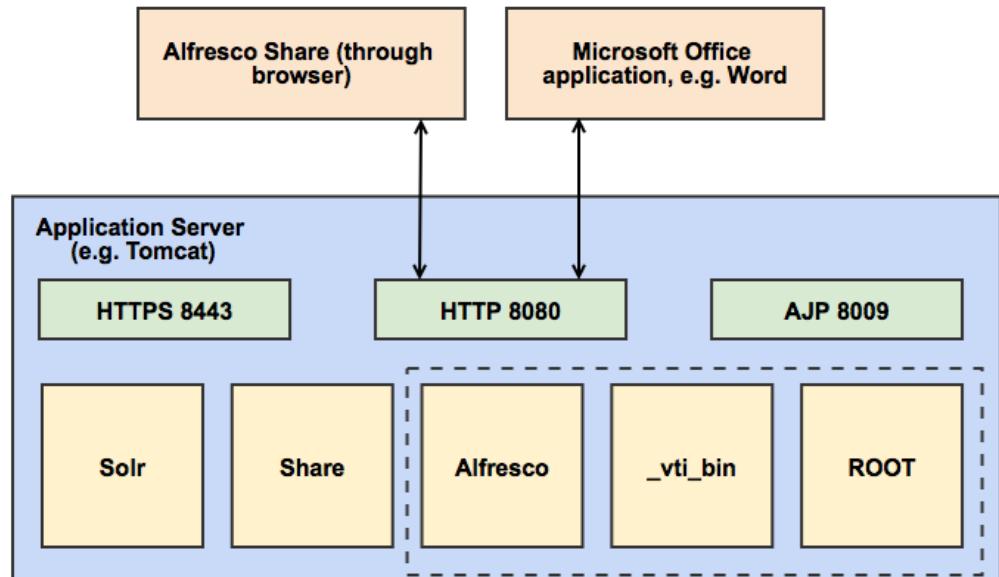
Installing and configuring Alfresco Office Services

When you install Alfresco, a fully-compatible SharePoint repository is also installed, that allows Microsoft Office Suite applications (for example, Word, PowerPoint, and Excel) to interact with Alfresco as if it was SharePoint. This feature allows you to edit Office documents in Alfresco Share and to modify Office files without checking them in and out. Alfresco locks the file while it is being modified and releases the lock when the file is saved and closed.

If you are installing the Alfresco repository manually, you'll need to install the Alfresco Office Services AMP file (`alfresco-aos-module-1.1-65.amp`), which deploys the `_vti_bin.war` file. See [Installing an Alfresco Module Package](#) on page 48 for installing an AMP file.

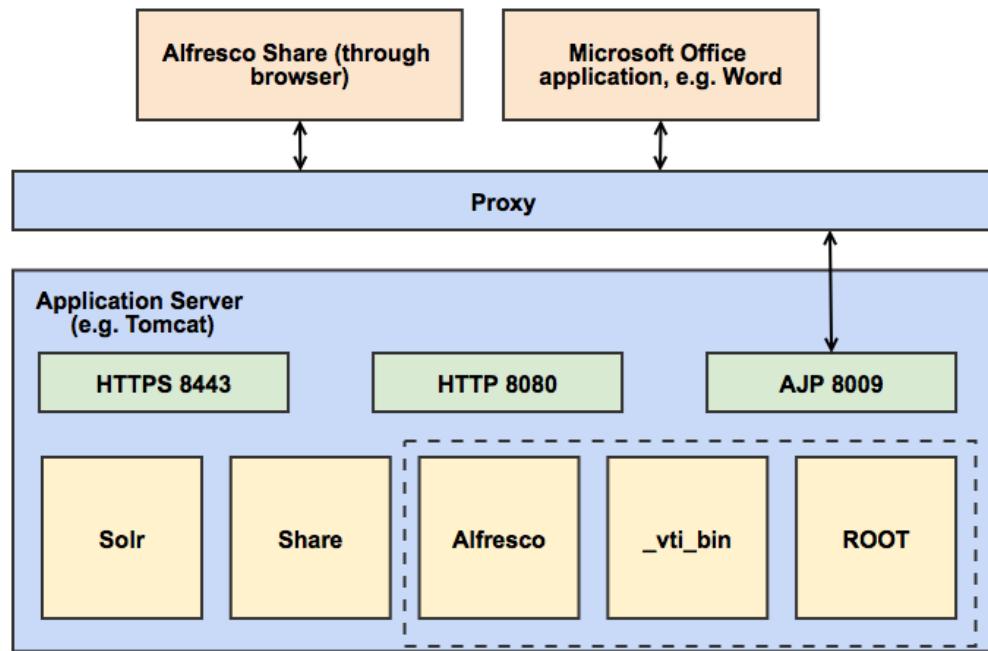
If you are deploying to a different application server, you'll need to set the context path during deployment. See [Deploying Alfresco with a different context path](#) on page 335 for more information.

It is important to note the URL required to access Alfresco from a Microsoft Office application. For more information, see [Using Alfresco from Microsoft Office](#). The following diagram shows the architecture of AOS in relation to an Alfresco installation:



Communication is over HTTP with either the Alfresco repository (through Alfresco Share) or directly from a Microsoft Office application.

If you are using a proxy server to handle SSL communication, the proxy handles the communication with Share and Microsoft Office through an AJP port (if you are using Tomcat). For more information about setting up a proxy server, see [Secure Sockets Layer \(SSL\) and the Alfresco repository](#) on page 345. The architecture is as follows:



AOS replaces and enhances the Microsoft SharePoint Protocol Support that was available in previous versions of Alfresco.

Prerequisites for using Alfresco Office Services

There are a number of software requirements for using AOS.

Alfresco Office Services is part of the standard Alfresco installation, and software and hardware requirements are the same as those for Alfresco. See [Supported platforms](#) for more information.

Microsoft Office requirements

- Microsoft Office 2010 (32 or 64-bit)
 - Microsoft Office 2013 (32 or 64-bit)
 - Microsoft Office for Mac 2011
- It is strongly advised that you activate SSL when using Alfresco Office Services. For more information, see [Secure Sockets Layer \(SSL\) and the Alfresco repository](#) on page 345.
- If you are using a non-SSL connection, you must edit your Mac or Windows registry as specified in [Setting up Alfresco Office Services using a non-SSL connection](#) on page 82.

Considerations when using Alfresco Office Services

- AOS relies on SSL to allow communication with the Alfresco repository:
 - It is strongly advised that you activate SSL when using Alfresco Office Services. For more information, see [Secure Sockets Layer \(SSL\) and the Alfresco repository](#) on page 345.
 - If you are using a non-SSL connection, you must edit your Mac or Windows client registry files as specified in [Setting up Alfresco Office Services using a non-SSL connection](#) on page 82 to allow for communication between Microsoft Office and Alfresco.

- If you are using a proxy server to handle SSL, make sure that the proxy is not filtering requests to Alfresco. For more information on proxy SSL configurations, see [Configuring SSL for a production environment](#) on page 346.
- There is limited support for AOS with Microsoft Office for Mac. It is a known problem that there is no property mapping function in Microsoft Office for Mac.
- AOS is installed by default during the standard Alfresco installation:
 - If you are installing the Alfresco repository manually, you'll need to install the Alfresco Office Services AMP file. See the guidance in [Installing an Alfresco Module Package](#) on page 48 for installing an AMP file.
 - If you have a custom application that is running at the server root directory, it is important that you merge the `_vti_inf.html` and `index.jsp` files into this application to enable AOS. For more information, see [Installing Alfresco into an existing web application](#) on page 46.
- AOS interacts very closely with Microsoft Office, and there are some implications as a result:
 - Alfresco simulates a SharePoint Site in the `/alfresco/aos` directory and uses the child folder to represent the SharePoint document library. As a result, Office does not check out documents in the repository root; that is, if your document is located in `/alfresco/aos`. Make sure that you add a child folder in the `/alfresco/aos` directory and place documents there. For example:
`http://localhost:8080/alfresco/aos/documents/doc1.docx`

Alfresco and Office handle property mapping and time values differently:

- Alfresco and Microsoft use different mechanisms to calculate Daylight Saving Time (DST). In Alfresco, DST is applied to dates; for example, a time in August is displayed in DST, but a time in November is displayed without DST. Microsoft applies DST to all dates depending on the current date. For example, if today is in August, the time values of all dates are displayed in DST, even a time in November. This means that if you are looking at a date six months away, there is a one-hour difference between the time value displayed by Alfresco One and the time displayed in Microsoft Office. This mechanism is used across Microsoft products; for example, the same behaviour is visible in the last modified timestamp in Windows Explorer.
- Date values are represented by Microsoft Office and Alfresco as `DateTime` values with the time zeroed out (for example, `03.09.2014 00:00:00`). After applying time zone conversion to this value, the date might change to the previous or next day. For example, if you are storing `03.09.2014 00:00:00` in UTC+2 and then reading the value in UTC-1.
- If mapped properties are embedded into an OOXML file (e.g. a `.docx` file), time values are displayed in the user's timezone. Properties embedded into OLE files (e.g. `.doc` files) are displayed in Coordinated Universal Time (UTC).
- There are known issues with decimal numeric values (float and double) in non-English versions of certain Office products and if Office runs with a non-English regional setting.

Upgrading Alfresco Office Services from a previous version of Alfresco

AOS replaces the Microsoft SharePoint Protocol Support that was available in previous versions of Alfresco. Use this information to upgrade from Microsoft SharePoint Protocol Support, or from a previous version of AOS.

1. Upgrade Alfresco, as described in [Upgrading Alfresco](#) on page 115.

 If you are not using the Alfresco setup wizard, it is very important that you install the `alfresco-aos-module-1.1-65.amp`, which deploys the `_vti_bin.war` file. See [Installing an Alfresco Module Package](#) on page 48 for installing an AMP file.

2. Launch Alfresco Share.

Test that you can edit your Microsoft Office documents by using the **Edit in Microsoft Office** action on any Office document in Alfresco Share.

3. Alternatively, open a Microsoft Office application (for example, Word) and select the **File** tab and **Open**. Enter the Alfresco server address in the **File name** field in the format: `http://servername:portnumber/alfresco-aos` and browse to a folder to edit an Office document.

Any version history from the previous version of Alfresco will not be available in Microsoft Office, but is available in Alfresco Share.

Configuring Alfresco Office Services

This information will help you to configure AOS for your environment.

Setting up Alfresco Office Services using a non-SSL connection

We strongly recommend that you activate SSL when using Alfresco Office Services. For more information, see [Secure Sockets Layer \(SSL\) and the Alfresco repository](#) on page 345. This information is provided in the case where it is not possible to set up SSL in your environment.

1. If you are using Microsoft Office for Mac 2011 without SSL, run this command on the client machine:

```
defaults -currentHost write com.microsoft.registrationDB
    hkey_current_user\hkey_local_machine\software\microsoft\office\
    \14.0\common\internet\basicauthlevel -int 2
```

2. Follow the remaining steps if you are using Microsoft Office 2010 or 2013. Set the following registry keys:

Each registry key takes the following values:

Options	Description
0	Basic authentication disabled
1	Basic authentication enabled for SSL shares only
2 or greater	Basic authentication enabled for SSL shares and for non-SSL shares

- a. Enable basic authentication on the client machine. Change or create the following registry key and set its value to 2:

```
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WebClient
    \Parameters\BasicAuthLevel" (REG_DWORD)
```

- b. Update the registry on the client machine, using the appropriate registry key for your version of Office. Change or create the following registry key and set its value to 2:

For Office 2010:

```
"HKEY_CURRENT_USER\Software\Microsoft\Office\14.0\Common\Internet
    \BasicAuthLevel" (REG_DWORD)
```

For Office 2013:

```
"HKEY_CURRENT_USER\Software\Microsoft\Office\15.0\Common\Internet
    \BasicAuthLevel" (REG_DWORD)
```

- c. Restart the web client Windows service for the changes to take effect.

3. If the Office client requests credentials for each operation, this might relate to domain policy and client settings. See the Microsoft article, [Prompt for Credentials When Accessing FQDN Sites From a Windows Vista or Windows 7 Computer](#), if you experience one of the following problems:
 - Windows 7 prompts for authentication when accessing documents
 - You are prompted to enter your credentials when you access an FQDN site from a computer that is running Windows Vista or Windows 7 and has no proxy configured

 If you are not being prompted for your credentials, try clicking the small down arrow next to the **Open** button (instead of clicking the **Open** button itself). The client will then prompt you for your credentials and you can start working with documents from Alfresco sites.

Setting up a global filepath to access Alfresco

In Windows Explorer, you can set up a Group Policy to manage Favorites on client machines, or share a .lnk file in your Links folder. This can be useful if you want to preconfigure the folder that users will need to access the Alfresco repository from Microsoft Office (<http://servername:port/alfresco/aos>).

On a Windows 7 machine, the contents of **Favorites** in Windows Explorer is assembled from the .lnk files in C:\Users\username\Links. You can create a .lnk file in your Links folder and distribute this to the Links folder of other users, or preferably, you can use a Group Policy to manage Favorites on user machines. Follow these steps to use a Group Policy:

1. In the Group Policy Management Console, navigate to User Configuration \Preferences\Window Settings\Shortcuts.
2. Create a new shortcut (Group Policy Object) to a folder (not a link to a URL) with the following UNC target path:

`\servername@SSL\DaVWWRoot\alfresco\aos`

Alternatively, you can specify @port instead of @SSL, but not both. If you use SSL, it must use the default port of 443.

For more information, see [Configure a Shortcut Item](#).

Configuring mapping properties in Alfresco Office Services

Property mapping in AOS allows you to inject custom properties and metadata into Office documents. Property mapping is deactivated by default. Follow these instructions to activate property mapping.

Property mapping is different from the standard metadata extraction mechanism and should be carefully configured to ensure that different properties are set up. Properties stored in the Alfresco repository are injected into Office documents when these files are read through AOS, and equally properties are extracted from Office files written through AOS and then updated in the Alfresco repository.

-  When added, these injected properties form part of the document. If the document is removed from the organization, for example, anyone outside the organization reading the document can view all the properties that have been mapped into the document.
-  • Single value properties only can be mapped in Office documents. Multi-value properties are ignored.
- Accepted data type properties are `text`, `mltext`, `int`, `long`, `float`, `double`, `date`, `datetime` and `boolean`. Other data type properties are ignored.
- The following constraints are supported: `MINMAX` for numeric data types, `LENGTH` for `text`, or `LIST` for `text`. Properties that have other constraints are ignored; for example, `REGEX` for `text`.

- Property mapping is only available for OOXML files (.docx, .xlsx, .pptx) and OLE files (.doc, .xls, .ppt). OLE files do not support read-only properties and are ignored. Protected properties are available in OOXML files only.

If any ignored properties are declared as mandatory, then users will not be able to save documents.

It is possible to define a list of types for new documents. Whenever a user creates a new document with the **Save As** dialog, Microsoft Office displays this list to choose from. If the type contains mandatory properties, Office enforces values for these properties before the file can be saved. Files created outside of Office (for example, in Windows Explorer) are created with a type of cm:content.

Alfresco provides basic configuration of four patterns, includedTypesPatterns, excludedTypesPatterns, includedAspectsPatterns and excludedAspectsPatterns in the aosBaseDataModelMappingConfiguration abstract bean.

1. Rename or copy the <classpathRoot>/alfresco/extension/custom-aos-metadata-mapping-context.xml.sample file to <classpathRoot>/alfresco/extension/custom-aos-metadata-mapping-context.xml.

This sample configuration file activates metadata mapping for the basic cm:content type and all its sub-types, except for some system types. All type properties and all applied aspects (except for some system aspects) are mapped into the documents.

2. In custom-aos-metadata-mapping-context.xml, check your file type based on the includedTypesPatterns and excludedTypesPatterns properties.

Both properties contain a list of regular expressions that are applied to the fully qualified QName. A file is valid for property mapping if its type is accepted by one of the regular expressions in the includedTypesPatterns list and does not exist in the excludedTypesPatterns list. For more information on regular expressions, see [Class pattern](#).

3. In custom-aos-metadata-mapping-context.xml, check the file aspects based on the includedAspectsPatterns and excludedAspectsPatterns properties.

If a file is valid for property mapping, aspects applied to this file are filtered further depending on the two properties, includedAspectsPatterns and excludedAspectsPatterns. Property mapping occurs only if the file type is included in the includedTypesPatterns list (even if there are aspects that are included in the includedAspectsPatterns property).

4. In custom-aos-metadata-mapping-context.xml, check the includedInstantiableTypesPatterns and excludedInstantiableTypesPatterns properties.

These properties define the list of types that are available to users for document creation in the **Save As** dialog. If the includedInstantiableTypesPatterns is empty or not set, new documents are always created with the default type. If no system type matches the types configured in includedInstantiableTypesPatterns, the base type cm:content is used by default. If exactly one type matches the configuration, this type is automatically used for all documents created with the **Save As** dialog in Microsoft Office.

Troubleshooting Alfresco Office Services

Issue with Online Editing

There is a known issue where Online Editing is not available using the 64-bit version of Internet Explorer. See [Plan browser support \(SharePoint Server 2010\)](#) for more information.

Error message: "The address is not valid" when connecting to `http://server:port/alfresco-aos`

If you have installed Alfresco manually or upgraded from a previous version of Alfresco, you might not have installed the Alfresco Office Services AMP file. If that is the case, you will receive an error message "The address is not valid" when you try to connect and authenticate with the address: `http://server:port/alfresco/aos`. You will also see the following error message in the server log:

Blocked a directory listing request from MS-Office. This indicates a broken MS-Office deployment. Please check that the ROOT and the _vti_bin webapps are deployed properly and reachable from the outside!

To fix this problem, ensure that you have installed the Alfresco Office Services AMP file, which deploys the `_vti_bin.war` file that is required for AOS to work correctly.

Error or blank screen when accessing web server

If you have not enabled SSL, you might encounter the following error message:

`Access to this web server is disabled by default because it is controlled by basic authentication and does not use Secure Socket Layer (SSL).`

If you are running an old version of Office, you might see nothing after entering the URL in the Office file dialog.

To rectify this problem, we strongly recommend that you activate SSL when using Alfresco Office Services. For more information, see [Secure Sockets Layer \(SSL\) and the Alfresco repository](#) on page 345. Alternatively, if you need to use a non-SSL connection, you must edit the Mac or Windows registry for every client machine as specified in [Setting up Alfresco Office Services using a non-SSL connection](#) on page 82.

Missing version history and check in/ check out options in Office

If you cannot see certain fields in the Document Panel in your Microsoft Office applications; for example, version history, check out and check in history, or you cannot see a directory listing for a file, it might be that your `ROOT` and `_vti_bin` files have not been deployed properly, or you have not applied the Alfresco Office Services AMP file, if you have installed Alfresco manually.

To check whether this is the case, try to open the `ROOT` and `_vti_bin` files from a browser. In these examples, replace `server:port` with your server and port details.

If you type: `http://server:port/`, you will see a message `Welcome to Alfresco!`.

If you type: `http://server:port/_vti_inf.html`, you will see a blank page. Select **Show page source** in the browser to see `_vti_bin ScriptUrl` information.

If you type: `http://server:port/_vti_bin/`, you will see a message `Welcome to Alfresco!` This is the `/_vti_bin` application. This application does not provide a web interface in the browser..

If these files and messages are not available from the browser, then AOS has not been deployed properly.

See [Installing the Alfresco WARs](#) on page 43 for information on where the deployed `ROOT` and `_vti_bin` WAR files need to be located. If `_vti_bin.war` does not exist, you'll need to reinstall the Alfresco Office Services AMP.

Extra files created when mounting AOS using WebDAV and Mac Finder

Do not mount the AOS repository root (`alfresco/aos` or any sub folder) as a WebDAV folder with Mac Finder. Otherwise you might see extraneous files in Alfresco Share; for example, files prefixed with the characters `.-`

Microsoft path length limitation

Microsoft Office has a general path length limitation of 250 characters. This affects any external application interacting with Office, not just AOS. Office can handle more than 250 characters in

many cases, but Microsoft does not provide official support in these circumstances. These are problems that you might encounter if you use long paths:

- Office reports that a document cannot be registered and OLE linking is deactivated (due to the path length limitation in OLE)
- The browser plug-in does not open a document

Avoid deep folder structures and path lengths over 250 characters, or if you must use long path lengths, test extensively with Office before deploying to a production environment.

File dialog in Microsoft Office shows file listing instead of graphical view

For untrusted servers, Microsoft Office blocks the graphical web view of files and instead shows the files as a list.

To solve this problem, either:

- On each client machine, in Internet Options or Internet Accounts, add the server in the list of trusted sites
- On your local intranet, modify the rules used to identify servers to include your server

Check in failure

There is a known problem if property mapping is activated and, in a single MS Office session, you create a new file with the **Save As** option, then check it out and check it back in. The check in will fail in this situation.

To avoid this problem, upgrade to Alfresco 5.0.1, or exclude read-only mandatory properties from the property mapping.

The problem is caused when some mandatory fields are not filled out, but are declared as read-only. This is typically caused by system properties (for example, Creator or Modifier) that come with some system aspects. You can avoid this by overwriting the includedAspectsPatterns configuration to include specific custom aspects only.

Property mapping failure with Office 2013 and Windows

If you are using Office 2013 and are working with an OLE file (for example, .doc, .xls, or .ppt files), and the **Protected View** is activated for the document, then property mapping can fail even after switching into **Editing Mode**.

To resolve this problem, you need to prevent the **Protected View** in Office by adding the Alfresco repository server to the list of trusted sites.

Values of date fields in OLE documents not stored

Values of Date and DateTime fields are not set in OLE documents (for example, .doc, .xls, or .ppt files) if the time zone of the client machine is greater than UTC+1. If these Date or DateTime fields are declared as mandatory, then you will not be able to save document changes.

To resolve this problem, you need to either set Date and DateTime as optional fields, or ensure that the time zone is not greater than UTC+1.

Problems deploying AOS on JBoss

If you use the JBoss application server, you must customize the web.xml file in the Alfresco ROOT.war, _vti_bin.war and share.war files to include this code fragment:

```
<context-param>
    <param-name>
        org.jboss.jbosfaces.WAR_BUNDLES_JSF_IMPL
    </param-name>
    <param-value>true</param-value>
```

```
</context-param>
```

This ensures that the JSF deployer in JBoss uses its own bundled JSF version.

Installing and configuring Alfresco Outlook Integration

Alfresco Outlook Integration is an extension to Alfresco and Microsoft Outlook, that allows you to save and file your emails to Alfresco in Microsoft Outlook, in a centralized and structured way.

You can drag and drop emails in and out of the repository, and add metadata automatically when an email is filed. Other features include leveraging Alfresco's in-built workflow processing and filtered search capabilities.

Advanced metadata support includes:

- Full support for custom models
- A configurable and dynamic metadata dialog
- The ability to map metadata configuration to a path, folder type, or aspect
- The ability to assign the same metadata to a set of emails in Microsoft Outlook, or a set of files in your file system

You can apply a sorted view to the Alfresco repository (in Microsoft Outlook), and page through a folder or site if it contains a large number of files.

For more information about installing and configuring Alfresco Outlook Integration, see [Installing and configuring Alfresco Outlook Integration](#).

Installing and configuring Alfresco S3 Connector

Use this information to install and configure Alfresco S3 Connector as an alternative content store.

 The Alfresco S3 Connector module can be applied to Alfresco Enterprise 4.1.1 or later. It requires an Alfresco instance running on Amazon's Elastic Compute Cloud (EC2), connected to Amazon's Simple Storage Service (SSS). Other devices or services that advertise as being S3 compatible have not been tested and are therefore not supported.

Using an Alfresco Module Package, the connector supplies a new content store which replaces the default file system-based implementation for the standard and deleted content stores. The content store implementation is responsible for reading and writing content streams using the S3 API, however, in order to improve performance a local Caching Content Store is used which uses the local disk to cache recently-used content items.

 By default the module configures the caching content store to use a maximum of 50 GB of disk, with no limit on individual file sizes.

The Alfresco S3 Connector uses a single S3 bucket and all content is stored in that bucket within one of the following directories:

```
<bucket-root>/contentstore for the main content store  
<bucket-root>/contentstore.deleted for the 'deleted' content store
```

 Always install the S3 connector cleanly. Upgrades from a local content store to S3 are not supported, and will corrupt the Alfresco repository.

Installing the Alfresco S3 Connector

These steps describe how to install the Alfresco S3 Connector to an instance of Alfresco.

The Alfresco S3 Connector functionality is packaged as an Alfresco Module Package (AMP) file.

 Ensure that you do not start Alfresco before installing the S3 AMP.

1. Go to the [Alfresco Support Portal](#).
2. Download the `alfresco-s3-connector-1.2.0-1.amp` file.
3. Use the Module Management Tool (MMT) to install the AMP into the repository WAR (`alfresco.war`).
For more information, see [Using the Module Management Tool \(MMT\)](#) and [Installing an Alfresco Module Package](#).
4. Restart the Alfresco server.

Configuring the Alfresco S3 Connector

The Alfresco S3 Connector is configured using properties set in the global properties file.

1. Open the `<classpathRoot>/alfresco-global.properties` file.

2. Add the `s3.accessKey`, for example:

```
s3.accessKey=AKIAIOSFODNN7EXAMPLE
```

The access key is required to identify the Amazon Web Services account and can be obtained from the Amazon Web Services site [AWS Credentials](#).

3. Add the `s3.secretKey` property, for example:

```
s3.secretKey=wJalrXUtnFEMI/K7MDENG/bPxRfCYEXAMPLEKEY
```

The secret key is required to identify the Amazon Web Services account and can be obtained from the Amazon Web Services site [AWS Credentials](#).

4. Add the `s3.bucketName` property, for example:

```
s3.bucketName=myawsbucket
```

The bucket name must be unique among all Amazon Web Services users globally. If the bucket does not already exist, it will be created, but the name must not have already been taken by another user. If the bucket has an error, it will be reported in the `alfresco.log` file. See [S3 bucket restrictions](#) for more information on bucket naming.

5. Add the `s3.bucketLocation` as specified in the [Amazon Simple Storage Service \(S3\) table](#).

The value is taken from the *Location constraint* column. For example, for EU (Frankfurt):

```
s3.bucketLocation=eu-central-1
```

 If you use a region other than the US Standard endpoint to create a bucket, `s3.bucketLocation` is a mandatory field. Use the [Amazon Simple Storage Service \(S3\) table](#) for guidance on the correct value.

6. If you need to use a single bucket for multiple purposes, set the content store as a subdirectory of the bucket, using these properties:

```
dir.contentstore=/SubPath/contentstore  
dir.contentstore.deleted=/SubPath/contentstore.deleted
```

7. Set optional configuration properties; for example, where the cached content is stored, and how much cache size you need:

The cached content location (and default value) is `dir.cachedcontent=${dir.root}/cachedcontent`. See [CachingContentStore properties](#) on page 574 for more information on the caching content store.

 The size of the local caching content store can be configured as necessary to limit its use to a maximum overall size or by files with a maximum file size. For example:

```
#Maximum disk usage for the cache in MB  
system.content.caching.maxUsageMB=51200  
#Maximum size of files which can be stored in the cache in MB (zero  
implies no limit)
```

```
system.content.caching.maxFileSizeMB=0
```

8. To configure an advanced S3 setup; for example, using a proxy server, see the [JetS3t](#) information for a full list of configuration parameters.
9. Save the `alfresco-global.properties` file.
You are now ready to start Alfresco.

Configuring the Alfresco S3 Connector with WebSphere Application Server

If you are using WebSphere Application Server with the Alfresco S3 Connector, you must configure additional settings in the WAS Console.

Configure an isolated shared library for S3 to use to prevent errors when starting Alfresco with WebSphere Application Server.

1. Create a folder for your shared library.
For example, `WAS_installation_directory/s3_shared_lib`.
2. Copy the following files from `alfresco.war\WEB-INF\lib\` to the new shared library folder:
 - `httpclient-<version>.jar`
 - `httpcore-<version>.jar`
 - `jets3t-<version>.jar`
 - `commons-codec-<version>.jar`
3. Navigate to the WAS console: `localhost:9060/ibm/console` and select **Environment > Shared libraries**.
4. Select the Alfresco application and **Create new shared library**.
5. Define the new shared library name, for example, `s3_shared_lib`, and for the classpath, choose the folder you created in step 1.
6. Check **Use an isolated class loader for this shared library**.
7. Click **Apply** and **Save**.
8. Navigate to **Applications > Application Types > WebSphere enterprise applications > Alfresco > References > Shared library references**, and check **Alfresco module**.
Make sure you do not check **Alfresco application**.
9. Click **Reference shared libraries** and add the new shared library name (`s3_shared_lib`) to the **Selected** column.
10. Click **OK** to save all changes.

Configuring the Alfresco S3 Connector with AWS GovCloud

The Alfresco S3 Connector can be configured to use the AWS GovCloud region.

AWS GovCloud is an isolated Amazon Web Services region, that allows US governmental agencies and customers to store content and workload in the cloud while still conforming to regulatory and compliance requirements. Alfresco supports both Federal Information Processing Standard (FIPS) compliant and non-FIPS compliant Amazon S3 AWS services. For more information about AWS FIPS compliance, see [AWS GovCloud \(US\) Endpoints](#).

1. Open the `<classpathRoot>/alfresco-global.properties` file.
Add one of these entries to your configuration, depending on the FIPS compliance status of your organization.
2. If you are a non-FIPS compliant organization, add the following entry to your `<classpathRoot>/alfresco-global.properties` file:
`s3service.s3-endpoint=s3-us-gov-west-1.amazonaws.com`

3. If you are a FIPS compliant organization, add the following entry to your <classpathRoot>/alfresco-global.properties file:

```
s3service.s3-endpoint=s3-fips-us-gov-west-1.amazonaws.com
```
4. Save the alfresco-global.properties file.

Installing and configuring the Alfresco Transformation Server

Use this information to install and configure the Alfresco Transformation Server.

Transformation Server overview

The Alfresco Transformation Server is a stable, fast, and scalable solution for high-quality transformations of Microsoft Office documents. It is an enterprise-scale and enterprise quality alternative for LibreOffice.

The server features an open architecture, and it offers the following features:

High quality

The Alfresco Transformation Server uses genuine Microsoft Office software to transform MS Word, Excel, and PowerPoint documents to PDF. This guarantees the handling of all Office files and pixel-perfect transformations, and it corrects previous layout issues in the Share preview feature.

Scalable

The Alfresco Transformation Server communicates with Alfresco using an HTTP REST API, which means that you can scale up by adding multiple instances of the server and connecting them through a standard HTTP Network Load Balancer.

Stable

If Microsoft Office can open and transform your document, then so can the Alfresco Transformation Server. Robust error handling will take care of corrupt and encrypted documents. A Web Console shows you a detailed report if there is a problem during transformation, allowing you to correct documents.

Fast

The Alfresco Transformation Server is two to three times faster when transforming multi-megabyte Office documents when compared with LibreOffice on the same hardware.

Extensible format support

The Alfresco Transformation Server supports the transformation of MS Office formats. Upcoming versions will support image and video transformations. Contact Alfresco Support if you need support for other formats.

Transformation Server prerequisites

The Alfresco Transformation Server consists of two software modules:

- Standalone Transformation Server
- Alfresco Transformation Client

The Alfresco Transformation Server is sold as a separate product, which can only be enabled with a separate license key.

The Standalone Transformation Server runs on Windows and takes care of the file transformations.

The Alfresco Transformation Client runs as a part of the Alfresco ECM server and takes care of the communication between Alfresco and the Transformation Server.

Disc I/O bandwidth

Microsoft Office transformations are I/O-heavy, and so on some solutions, I/O contention can be a performance bottleneck. When multiple Word conversions occur in parallel, performance can suffer heavily from poor random read and write speeds.

Using an Amazon EC2 instance c3.2xlarge, the I/O metrics are as follows:

- seq. read speed: 131 MB/s
- seq. write speed: 83 MB/s
- random qd32 read speed: 10,4 MB/s
- random qd32 write speed: 3,8 MB/s

Tests on this environment indicate that I/O was the performance bottleneck, not the transformation server software. Effectively this measures disk speed, not transformation performance. Switching to SSD instance resulted in `.doc` files transforming 1.2 times faster and `.docx` files transforming 1.7 times faster.

Standalone Transformation Server prerequisites

The Alfresco Transformation Server requires prerequisite software components to be installed and available on the same machine.

See [Alfresco Supported Platforms](#) for details of the correct prerequisite software.

The following points are important to note before you install the Transformation Server:

- Install only the English versions of MS Windows Server 2012 and Office 2013 because other languages result in unpredictable behavior.
 -  Although the server must be configured in English, this has no impact on the transformation language used for documents.
- Make sure that the Windows print spooler service is running
- Java 7 is not supported
- Java 6 x64 is not supported

There are a number of recommendations for calculating sizing. You will need:

- Four high clocked cores per server, with between 4 GB and 6 GB RAM. If you find that you need more power, it is better to add another server instance with a similar specification than to upgrade the hardware. The reason for this is that Microsoft Office is not very scalable.
- Between 10 GB and 15 GB of free space. Storage is not that important, but if you have lots of large files, you should make sure that creating temporary copies of those files will not slow the system down.
- Gigabit Ethernet
- At least one CPU for each concurrent transformation that is expected to be processed by the server

Installing the Alfresco Transformation Server

Use this information to install all the components required for the Alfresco Transformation Server.

The following artifact is shipped for the Alfresco Transformation Server:

`alfresco-transformationserver-1.5.1-23.zip`

The Transformation Server zip file contains the following files:

- `alfresco-5.0-transformationserver-repo-1.5.1.amp`
- `alfresco-5.0-transformationserver-share-1.5.1.amp`

- alfresco-5.0-transformationserver-server.msi
- TransformationServer-amps-1.5.1-releaseNotes.html
- TransformationServer-server-1.5.1-releaseNotes.html

Installing the Transformation Server consists of two parts:

1. Installing the MSI installation package on the standalone Transformation Server.
2. Installing the relevant AMP files and updating the license on the Alfresco server.

 When upgrading the Transformation Server, the previous installation must be uninstalled first. If your old version of the Transformation Server is earlier than 1.3.1, you need to use the Control Panel's **Uninstall a program** option to remove the old version, and then manually remove the Transformation Server directory. By default, the Transformation Server directory is `C:\Program Files (x86)\Transformation Server\`). If your old version of the Transformation Server is 1.3.1 or later, the new Transformation Server msi prompts you to uninstall the previous version. Once the uninstall is complete, you can run the msi again to install the new version. There is no need to manually remove anything.

Installing the standalone Transformation Server

Use this information to install the standalone Transformation Server.

Before you start the installation, verify that you have:

- Installed and activated the correct software (see [Standalone Transformation Server prerequisites](#) on page 91)
 - Logged on to the Windows Server as a user with administrator rights
1. Double click the MSI installer package `alfresco-5.0-transformationserver-server-1.4.1.msi`.
The Welcome screen opens.
 2. Click **Next**.
The license information screen displays.
 3. Click **Next**.
 4. Select an installation folder or accept the default folder, and then click **Next**.
 5. Select the TCP/IP ports used by the Transformation Server.
The default values are 8080 (HTTP) and 8443 (HTTPS) but you can also use the standard ports 80 and 443 (or any other port) if this fits better into your network infrastructure.
 6. Click **Next** to start the installation.
You see a progress bar and a command line window during the installation. The installer will show a confirmation when the installation is finished.
 7. Click **Next** to finish the installation.
 8. Verify that the installation has completed successfully.
 - a. Check the Windows Services in the management console.
 - b. Locate the new service called **Transformation Service**, and check that it is **Started**.
-  Each time a file is transformed in Alfresco, the .NET program starts and Microsoft Office tries to check for a Certificate Revocation List (CRL).
- Depending on the access that the Transformation Server has to the Internet when transforming a file, this check can delay the operation for up to two minutes, and will therefore, delay transformation of the file.

To prevent this, use the Windows server firewall to block internet access for all office binaries.

Installing the Transformation Server on Alfresco

This information will help you to install the Transformation Server AMP and to update the required license.

Before you start, make sure that you verify the following prerequisites:

- Check that your Alfresco server is correctly configured and tested
 - Make sure that you have the correct Transformation Server ZIP file for the version of Alfresco that you are running
 - Make sure that you have an updated license file (a `*.lic` file)
1. Stop the Alfresco server.
 2. Open a terminal (Linux) or command line window (Windows).
 3. Unzip the `alfresco-transformationserver-1.5.1-23.zip` file.
 4. Move the `repo` AMP file to the `<ALFRESCO_HOME>/amps` folder, and move the `share` AMP file to the `<ALFRESCO_HOME>/amps_share` folder.
 5. Install the AMP files using the Module Management Tool (MMT).
 6. Copy your updated license file into the Alfresco installation folder.
Delete all files with extension `*.installed` in this directory.
 7. Start the Alfresco server.
 8. Monitor the Alfresco log.

You will see successful log entries about the license installation and the installation of the Alfresco Module Package (depending on the configuration of your log level).

Configuring the Alfresco Transformation Server

Configuring the Alfresco Transformation Server consists of two parts: configuring the Standalone Transformation Server using the Web Console, and configuring the Alfresco Transformation Client using a properties file or JMX.

Configuring the Standalone Transformation Server

This information describes how to configure the Standalone Transformation Server. You need only to change the password of the transformation service.

1. Open your browser and navigate to the following URL:
`http://<transformation-host>:<port>/transformation-server/settings`
or `https://` if you are using SSL
2. Enter your login name and a password.
By default, the login name is set to `alfresco`, and the password is set to `alfresco`. The login name `alfresco` cannot be changed.
3. Enter a new password, and then click **Change** to save the password.

If you close and reopen your browser, reenter your login and new password.

Configuring the Transformation Client

You can configure the Transformation Client by defining several parameters; for example, by using HTTP or HTTPS.

There are three ways that you can configure the Alfresco Transformation Client:

- Using the `alfresco-global.properties` file
- Using a JMX client, if you have installed the Oracle Java SE Development Kit (JDK)
- Using the `default-configuration.properties` file

Configuration using the global properties file

You configure the Transformation Client by adding the relevant properties to the Alfresco global properties file.

1. Open the `alfresco-global.properties` file.
2. Add the required properties for configuration settings on the Transformation Client.
3. Save the `alfresco-global.properties` file, and then restart your Alfresco server.

The following table shows an overview of the available properties:

Property	Default value	Description
<code>transformserver.aliveCheckTimeout</code>	2	Sets the timeout for the connection tester in seconds. If the transformation server does not answer in this time interval, it is considered to be off line.
<code>transformserver.test.cronExpression</code>	0/10 * * * * ?	Sets the cron expression that defines how often the connection tester will check. The default value is every 10 seconds.
<code>transformserver.disableSSLCertificateValidation</code>	false	Set this property to true to allow self-signed certificates (that is, it is not issued by an official Cert Authority).
<code>transformserver.username</code>	alfresco	The user name used to connect to the Transformation Server.  Do not change this default.
<code>transformserver.password</code>	alfresco	The password used to connect to the Transformation Server.  Always change the password from the default.
<code>transformserver.qualityPreference</code>	QUALITY	There are two values for this property: <ul style="list-style-type: none"> • QUALITY: optimizes the preview for quality. • SIZE: optimizes the preview for size. This is interesting if you have a lot of big Office documents, for example, PPT > 100 MB.
<code>transformserver.transformationTimeout</code>	300	Sets the time in seconds to wait for the transformation to complete before assuming that it has hung and therefore stop the transformation. If you are transforming very large or complex files, this time can be increased.

Property	Default value	Description
transformserver.url		The URL of your Transformation Server (or the network load balancer if you are using more than one Transformation Server). Use https:// if you want to use encrypted communication between the Alfresco server and the Transformation Server.

In a normal setup, you will always overwrite the `transformserver.password` and `transformserver.url` properties. If you want to use SSL encryption with the default certificate of the transformation server, make sure that you set `transformserver.disableSSLCertificateValidation=true`.

Configuration using JMX

The Transformation Client configuration parameters are exposed as JMX MBeans, which means that you can view and set the parameters using a JMX client.

See [Runtime administration with a JMX client](#) for instructions on how to connect a JMX client to your Alfresco server.

Configuration using the default configuration properties file

You can configure timeout values in the Transformation Client by adding the relevant properties to the transformation server configuration file, for example; C:\Program Files (x86)\TransformationServer\tomcat\webapps\transformation-server\WEB-INF\classes\default-configuration.properties.

Use the code sample to set these timeouts:

```
# transformer timeout in seconds
transformer.timeout.default=300
transformer.timeout.word = ${transformer.timeout.default}
transformer.timeout.excel = ${transformer.timeout.default}
transformer.timeout.powerpoint = ${transformer.timeout.default}
```

Using the Transformation Server

Whenever you upload your Office files in Share, you will now be using the Alfresco Transformation Server, and you can see results in the Share preview.

Administrators can view information about the server and transformation errors using the Web Console.

Using the Transformation Server Web Console

Use the Transformation Server Web Console to view information about the server and transformation errors. The server lets you view the status of the server, a historical view of all the transformations completed, and the number of successful and failed transformations.

Only Administrators can access and use the Transformation Server Web Console.

1. To open the Transformation Server Web Console, open a browser, and then navigate to the following URL:

`http://<transformation-host>:</port>:/transformation-server/`

Use `https://` if you use SSL.

The **Server Status** view is the default view when you open the Transformation Server Web Console. The **Server Status** view shows an overview of the health and the memory

use of the Transformation Server. Ensure that you have the flash plug-in to see the **Active Threads** and **Memory Usage** graphics.

2. Click **History** view.

Alternatively, you can go directly to the **History** view by opening a browser, and then navigating to the following URL:

```
http://<transformation-host>:<port>/transformation-server/
transformations
```

The **History** view shows the details of the document transformations. It provides a number of search functions that allow administrators to find transformation problems for specific documents.

3. You can query the transformation history using the following parameters:

- Date-time From and To
- File name
- Status
- User name

4. To investigate errors, set the **Outcome** field to **Error**. Hover over the warning sign to view an indication of the problem with the file.

5. Click the **Statistics** view.

Alternatively, you can go directly to the **Statistics** view by opening a browser, and then navigating to the following URL:

```
http://<transformation-host>:<port>/transformation-server/stats
```

The **Statistics** view indicates the number of transformations, and the success or failed ratio.

6. Click the reset link to reset the counter.

Integrating with monitoring tools

You can integrate the Alfresco Transformation Server with monitoring tools, for example Nagios or Hyperic, by using HTTP REST calls.

The tool should call the Transformation Server URL with a set of parameters, and then monitor the response.

Two calls are available:

1. Connection tester call

This call is also used by the Alfresco Transformation Client to test availability. It checks the transformation service is up and responding.

- a. URL: `http://<transformation-host>:<port>/transformation-server/ / service/transform/v1/version`
- b. HTTP Method: GET
- c. Make sure that you include basic authentication credentials to your call.

2. Transformation execution call

This call posts an Office file to the Transformation Service to check whether the transformation engine is still functioning. This can be used for more in-depth monitoring.

- a. URL: `http://<transformation-host>:<port>/transformation-server/ / service/transform/v1/available`
- b. HTTP Method: POST
- c. Make sure that you include basic authentication credentials to your call.

Installing and configuring Alfresco Web Quick Start

Web Quick Start

Web Quick Start is packaged in four parts:

- An Alfresco Module Package (AMP) that extends the repository to support a generic website model
- An AMP that extends Alfresco Share for editing content for the website, managing the structure of the website, and publishing content using workflow.
- A JAR file that contains a Java API for accessing the website data held in the repository.
- A web application that, when deployed to a servlet container such as Tomcat, delivers a fictional financial news website. The web application is a Spring MVC application constructed using Alfresco Surf, and communicating with the Alfresco repository using the Java API. As well as dynamically building the website from data held in the repository, Web Quick Start also provides examples of user generated content whereby content is sent from the web application back to the repository.

About Web Quick Start

With Quick Start, developers can rapidly build customized and dynamic web applications with powerful content management features for the business users without having to start from scratch.

Using standard development tools developers can quickly deploy the comprehensive content management capabilities of Alfresco to build new and innovative web applications. Developed using the Spring framework with Alfresco Surf, the Web Quick Start allows developers to easily extend Alfresco to add new features to support the demands of the business.

Installing Alfresco and Web Quick Start

When you run the setup wizard, you can choose to install a number of Alfresco components. Web Quick Start is provided as a component but it not selected by default.

Manually installing Web Quick Start

This procedure describes how to copy the AMP files into their appropriate AMP directories and uses the `apply_amps.bat` or `.sh` file to apply them. Alternatively, use the Module Management Tool (MMT) to apply the AMP file.

1. Download the Web Quick Start zip bundle file:
`alfresco-wcmqs-5.1.zip`
2. Unzip the file into a temporary location. The artifacts supplied with Web Quick Start are:
 - `alfresco-wcmqs-5.1.amp` (AMP file for Alfresco)
 - `alfresco-wcmqs-share-5.1.amp` (AMP file for Share)
 - `awe.war` (Web Editor)
 - `wcmqs.war` (Spring-based Web Quick Start application)
 - `awe-config-custom.xml`
3. Locate your Alfresco installation directory.
4. Copy the AMP files into the relevant amps directories for Alfresco and Share:
 - a. Copy the `alfresco-wcmqs-5.1.amp` file to the `amps` directory.
 - b. Copy the `alfresco-wcmqs-share-5.1.amp` file to the `amps-share` directory.

5. Apply the AMP files using the `apply_amps` command for the Tomcat application server, or, alternatively, use the Module Management Tool (MMT).
6. Copy the website WAR (`wcmqs.war`) into the `webapps` directory of your existing Alfresco installation.

For example, on Windows with a Tomcat application server, this is `C:\Alfresco\tomcat\webapps`.
7. Copy the Alfresco Web Editor file (`awe.war`) into the `webapps` directory to replace the existing `awe.war` file.
8. Delete the existing `alfresco` and `share` directories.
9. Restart the Alfresco server.

Creating the Web Quick Start site

1. Open Share.
2. Click **Create Site**.

This creates a new collaboration site.
3. Type a name for the site, for example, **Web Quick Start**.
4. Type a URL name for the site, for example **wcmqs**.
5. Click **OK**. The new site displays in your My Sites dashlet.
6. Open the new site.
7. Click **Customize Dashboard**.
8. Click **Add Dashlets**.
9. Drag the Web Quick Start dashlet to your dashboard layout.
10. Click **OK**.

The Web Quick Start dashlet displays in the site dashboard.

Importing Web Quick Start demo data

1. Click **Import Website Data**.

Choose the sample content to import: **Government or Finance**.
Both samples are identical in functionality but contain different images and section headings. The samples provide an example of how developers can package and import their own sample site data.
The system imports the data for the demo website.
2. Refresh the browser running Share.

The Web Quick Start dashlet now displays a link to the **Web Quick Start Help**.

By default, Web Quick Start is configured to be accessed at `localhost` on port 8080. If these settings are relevant for your installation and the `wcmqs.war` is running in the same container as Alfresco, you will now be able to access the Web Quick Start editorial website on <http://localhost:8080/wcmqs>.

To change the server host name, port, or web application context from the default values, refer to [Configuring Web Quick Start](#).

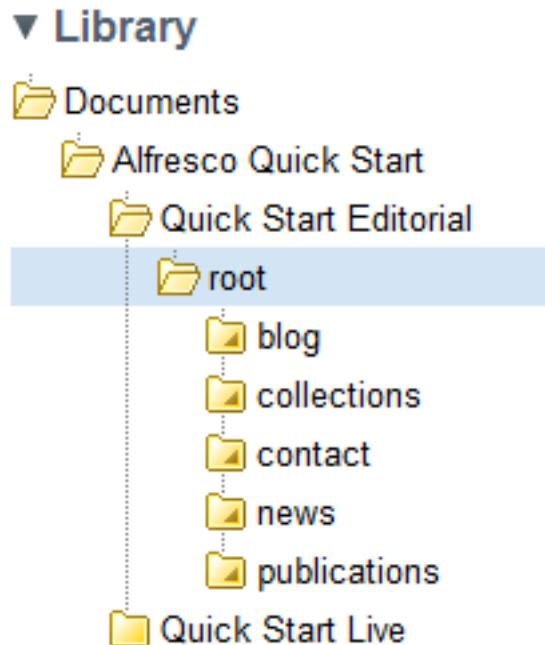
Configuring Web Quick Start

After you have imported the Web Quick Start website data, when you have refreshed Share, or the next time you log on, you can access the Web Quick Start site for configuration.

1. Open the Web Quick Start site.

2. Navigate to the Document Library.

The default site structure will have the following structure:



The site structure contains two folders: **Quick Start Editorial** and **Quick Start Live**. These folders represent a separation between the work in progress content, and the finished, reviewed, editorially complete content that is then published to the “Live” environment.

If your web container is running on port 8080 and the web application is running in the same container as Alfresco, the setup is complete and you should be able to access the web site on <http://localhost:8080/wcmqs>.

Configuring the web application host name, port, and context

The Web Quick Start installation assumes that the web application has been deployed to localhost on port 8080, using the context of wcmqs. This means that the editorial website can be accessed at <http://localhost:8080/wcmqs>. The “live” website can be accessed as default on <http://127.0.0.1:8080/wcmqs>.

If you are not running the web application on port 8080 or if the web application is deployed to a different container or host, you can configure the site to the required location.

1. In the Web Quick Start site, navigate to the **Document Library**.
2. Click **Edit Metadata** on either the **Quick Start Editorial** folder, or the **Quick Start Live** folder.
3. Configure the **Host Name**, **Port**, and **Web App Context** fields to point to the location your web application (`wcmqs.war`).
4. Click **Submit**.

Disabling AWE on the Live environment

This procedure configures the web application to view the “Live” site structure.

1. Edit the metadata properties on the **Quick Start Live** folder.
2. In the **Site Configuration** field, enter the `isEditorial=true` flag.

The screenshot shows a configuration dialog box with the following fields:

- Name:** Quick Start Editorial
- Title:** Alfresco WCM Quick Start
- Description:** This is a demonstration Alfresco WCM backed Internet Web Site
- Host Name:** localhost
- Host Port:** 8080
- Web App Context:** wcmqs
- Site Configuration:** isEditorial=true
- Publish Target:** Quick Start Live (selected)

At the bottom are **Submit** and **Cancel** buttons.

3. Click **Submit**.

The default configuration sets the host address to 127.0.0.1, so if you are running Web Quick Start locally, you can view the editorial environment on <http://localhost:8080/wcmqs> and the live on <http://127.0.0.1:8080/wcmqs>.

Configuring the API

You configure the API in the `wcmqs-api.properties` file. The file is located in the `clientapi` JAR file in the `alfresco` folder. You can override this location by adding a file with the same name on the classpath before the `clientapi` JAR. For example, if you're using the WQS API from within a JEE webapp then add a `wcmqs-api.properties` file to the `WEB-INF/classes/alfresco/` folder.

You can specify the following properties:

`wcmqs.api.alfresco`

The base URL for the Alfresco repository. The default value is `http://localhost:8080/alfresco`.

wcmqs.api.user

The username to authenticate the WQS API to Alfresco. It is recommended that this is changed.

wcmqs.api.password

The password to authenticate the WQS API to Alfresco. It is recommended that this is changed.

wcmqs.api.alfresco.cmis

The URL that the API will use to reach the CMIS interface. The default value is `%{wcmqs.api.alfresco}/service/cmis`.

wcmqs.api.alfresco.webscript

The base URL the API uses invoke webscripts in the Alfresco repository. The default value is `%`.

To override individual properties, place them in a file named `wqsapi-custom.properties` located on the classpath in a `/alfresco/extension/` (under `/shared/classes/` in a Tomcat installation, for example). You can specify these additional properties in this file:

wcmqs.api.repositoryPollMilliseconds

The time the API will wait between checks for the repository being available. This mechanism ensures that the webapp can be started before the repository. The webapp will connect when the repository becomes available. The default value is 2000 milliseconds.

wcmqs.api.sectionCacheSeconds

The time the API caches section objects before reloading them from the repository. The default value is 60 seconds.

wcmqs.api.websiteCacheSeconds

The time the API caches website objects before reloading them from the repository. The default value is 300 seconds.

Alfresco Web Editor

The Alfresco Web Editor is a Alfresco Surf-based web application that provides in-context editing capabilities for Alfresco repository content. The editor provides a mechanism for non-technical users to make edits to Alfresco content directly within a web page.

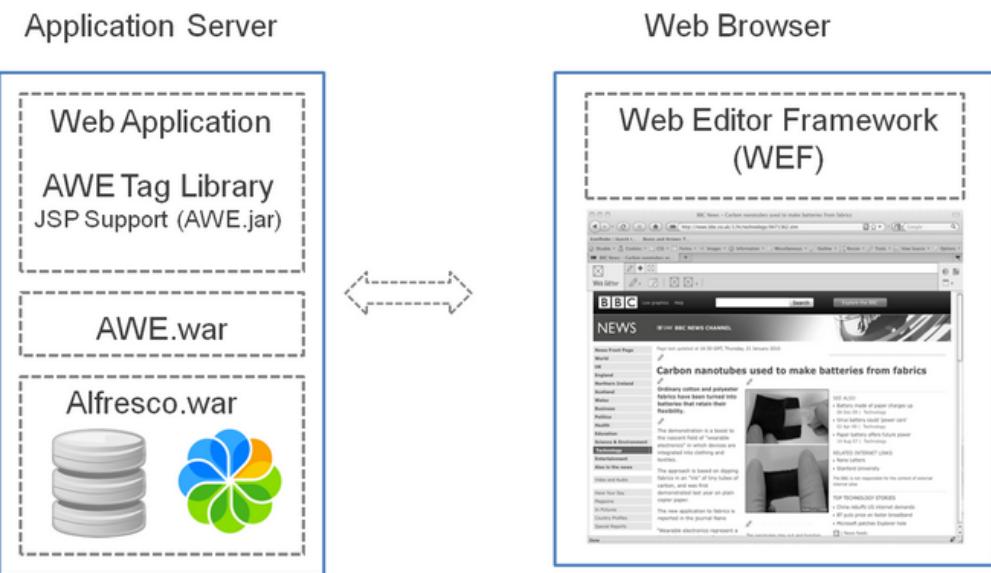
The Alfresco Web Editor uses the Forms Service default template.

The Alfresco Web Editor is packaged as a stand-alone WAR file so that it can be deployed to web applications that are in the sample instance, or remote, to the Alfresco server. When it is deployed, an Alfresco banner displays in your deployed web pages showing the Alfresco Web Editor tab and it identifies the editable content. By default, it assumes that you have JavaScript enabled but it can also run without JavaScript.

Alfresco Web Editor deployment

The simplest way to deploy the Alfresco Web Editor (AWE) is to use the pre-built WAR (`awe.war`) file and to deploy it in the same application server instance of your web application.

The following diagram shows an example Alfresco Web Editor deployment in the same application server as the Alfresco repository.



The Alfresco Web Editor is a Alfresco Surf-based application, therefore it is also possible to deploy it in a different application server instance from the Alfresco repository.

By default the AWE assumes your Alfresco repository is at <http://localhost:8080/alfresco/>. If your repository is not located here, you can use custom configuration to tell the AWE where to find your repository. To change the default repository location, add the following XML in the AWE configuration file with your values for **MY SERVER** and **MY PORT**:

```
<alfresco-config>
  <plug-ins>
    <element-readers>
      <element-reader element-name="remote"
class="org.springframework.extensions.config.RemoteConfigElementReader" />
    </element-readers>
  </plug-ins>

  <config evaluator="string-compare" condition="Remote">
    <remote>
      <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://MY SERVER:MY PORT/alfresco/s</endpoint-url>
        <identity>user</identity>
      </endpoint>
    </remote>
  </config>
</alfresco-config>
```

The AWE configuration file is placed on the classpath named `shared/classes/alfresco/web-extension/awe-config-custom.xml`.

The deployment comprises the following components:

AWE.war

The Alfresco Web Editor WAR file.

Web Application

Your own web application.

AWE tag library

Provides the ability to mark areas of the page as editable. The areas marked can represent any property or content from the Alfresco repository.

Web Editor Framework (WEF)

The client-side JavaScript framework on which the Web Editor is built. It is built using YUI and can be extended easily. New tabs and buttons can be packaged and dropped into the framework. This provides the core Alfresco product features, and also provides the ability to build additional custom plugins.

When the Alfresco Web Editor is enabled, the WEF renders the tool bar and basic in-context editing buttons and functionality. If the WEF is deployed as standalone, the default blank tool bar is rendered.

Deploying the Alfresco Web Editor

The Alfresco Web Editor distribution consists of a single zip file named `alfresco-webeditor-5.1.zip`.

1. Shut down your Alfresco server.
2. Browse to the Alfresco download area.
3. Download the `alfresco-webeditor-5.1.zip` file.
4. Deploy the `awe.war` file into the same application server instance as the Alfresco repository.
5. Copy the `alfresco-webeditor-taglib.jar` file to the `WEB-INF/lib` folder of your application.
6. To include the tag library in your application, add the following tag library declaration to your JSP page:

```
<%@ taglib uri="http://www.alfresco.org/tags/awe" prefix="awe" %>
```

Once the tag library is declared, you can use the `startTemplate`, `endTemplate` and `markContent` tags within your application.

7. Restart your Alfresco server.

Deploying the Alfresco Web Editor to a Alfresco Surf application

The Alfresco Web Editor distribution also includes all the files required to provide the functionality within an existing Alfresco Surf application.

1. Copy the following files to your application `WEB-INF/lib` directory:
 - a. `yui-2.7.0.jar`
 - b. `spring-webeditor-1.0.0.CI-SNAPSHOT.jar`
 - c. `alfresco-forms-client.jar`
 - d. `alfresco-webeditor-plugin.jar`

The `yui` and `spring-webeditor` JAR files represent the Web Editor Framework (WEF) upon which the Web Editor is built. The remaining `alfresco-form-client` and `alfresco-webeditor-plugin` JAR files provide the Web Editor functionality.

2. If you plan to use the Web Editor within the application (rather than the application being a host for the Web Editor services) you also must copy the following additional files into the `WEB-INF/lib` directory:
 - a. `spring-webeditor-client-jsp-1.0.0.CI-SNAPSHOT.jar`
 - b. `alfresco-webeditor-taglib.jar`

3. If you use the additional files, define a servlet filter in your application's `web.xml` file. If you do not provide the filter, the tags will be ignored. The following filter configuration is required:

```
<filter>
    <filter-name>Alfresco Web Editor Filter</filter-name>
    <description>Enables support for the Alfresco Web Editor</description>
    <filter-class>org.alfresco.web.awe.filter.WebEditorFilter</filter-class>
    <init-param>
        <param-name>contextPath</param-name>
        <param-value>/your-context-path</param-value>
    </init-param>
</filter>

<filter-mapping>
    <filter-name>Alfresco Web Editor Filter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

4. Set the `contextPath` parameter.

If you do not provided a value for this parameter, a default `contextPath` of `/awe` is presumed.

No further configuration is required as all the necessary Spring context files and Alfresco configuration files are contained within the JAR files. However, there is no default hook point for custom form configuration but this can be located anywhere within your application.

Configuring Alfresco Web Editor

The following Web Editor components must be configured:

- tag library, that is, the `markContent` tag used to define editable content
- servlet filter
- form configuration

Configuring the tag library

there are a number of steps needed to configure the tag library.

The tag library comprises the following tags:

- `startTemplate`
- `markContent`
- `endTemplate`

1. The `startTemplate` tag bootstraps the WEF using a script element that executes a web script. Place this tag in the `head` section of your page.

The `startTemplate` tag has only one optional attribute.

toolbarLocation

Controls the initial location of the tool bar. The valid values are: `top`, `left`, and `right`. The default is `top`.

The following shows an example of how to use the `startTemplate` tag:

```
<awe:startTemplate toolbarLocation="top" />
```

2. Use the `markContent` tag to indicate an editable area of the page.

The tag renders an edit icon that, when clicked, displays a form for editing the corresponding Alfresco content and properties, or both.

The `markContent` tag has two mandatory attributes and two optional attributes.

id

The mandatory identifier attribute specifies the NodeRef of the Alfresco node to be edited.

title

The mandatory title attribute defines a descriptive title for the editable area being marked. The title used is used in the quick edit menu of editable items, as the title of the form edit popup/dialog and the alt text and tool tip text of the edit icon.

formId

This is an optional attribute that specifies which form will be used when the marked area is edited.

nestedMarker

This is an optional attribute, which defines whether the editable area is nested within another HTML tag that represents the content being edited. If it is set to true, the whole parent element is highlighted when the area is selected in the quick edit menu. If set to "false" only the edit icon is highlighted.

An example use of the `markContent` tag is shown:

```
<awe:markContent id="<%subTextNodeRef%>" formId="description"
    title="Edit Description" nestedMarker="true" />
```

3. The `endTemplate` tag initializes the Web Editor with details of all the marked content areas on the page. It also renders a script element that executes the WEF resources web script, which starts the process of downloading all the assets required to render and display the tool bar and all configured plugins. Place this tag just before the closing body element.

The `endTemplate` tag does not have any attributes.

The following shows an example of how to use the `endTemplate` tag:

```
<awe:endTemplate />
```

Configuring the servlet filter

The `startTemplate`, `markContent`, and `endTemplate` tags will only render their output if they detect the presence of the Web Editor servlet filter. The tags can remain in the JSP page in production and have no effect until the servlet filter configuration is added to the `web.xml` file.

1. Add the following servlet filter configuration to the web application's `web.xml` file:

```
<filter>
    <filter-name>Alfresco Web Editor Filter</filter-name>
    <description>Enables support for the Alfresco Web Editor</description>
    <filter-class>org.alfresco.web.awe.filter.WebEditorFilter</filter-
class>
</filter>

<filter-mapping>
    <filter-name>Alfresco Web Editor Filter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

This enables the tags.

2. Set the following two optional parameters:

```
<init-param>
    <param-name>contextPath</param-name>
    <param-value>/quickstart</param-value>
</init-param>

<init-param>
    <param-name>debug</param-name>
```

```
<param-value>true</param-value>
</init-param>
```

These parameters control the `contextPath` that is used when URLs to the Web Editor are generated and the debug mode.

Configuring Web Editor forms

The Alfresco Web Editor (AWE) uses a form to edit the node referenced by a `markContent` tag. By default, the form displayed will contain the `cm:title`, `cm:description`, and `cm:content` fields. An alternative form can be used by providing the `markContent` tag with a `formId` attribute.

Out of the box, only two other forms are configured: a form with an identifier of `title`, and one with an identifier of `description`. As the identifiers indicate, the forms display a single property: `cm:title` and `cm:description`, respectively. The node type is presumed to be `cm:content`.

If you have custom types or wish to specify other properties, you can use the forms configuration techniques.

When starting up, the AWE looks for a configuration file on the classpath named `shared/classes/alfresco/web-extension/awe-config-custom.xml`. Place any custom form definitions in this file.

Sample web application using Alfresco Web Editor

A sample customer WAR file is available in the Alfresco Web Editor distribution. It demonstrates how a customer might use Alfresco Web Editor in a very simple JSP-based web application. This sample must not be used in a production environment and is not supported.

A sample customer tag library is provided, which includes two tags. These tags are included as a demonstration sample and should never be used in a production environment.

content

Allows content to be pulled from an Alfresco repository and sends output to a JSP page. The `content` tag requires one mandatory attribute called `nodeRef`

property

Allows properties to be pulled from an Alfresco repository and sends output to a JSP page. The `property` tag requires two mandatory attributes: `nodeRef` and `property`.

The following example show the use of these tags:

```
<customer:content nodeRef="<%=\$mainTextNodeRef%" />
<customer:property nodeRef="<%=\$subTextNodeRef%" property="cm:description" />
```

The sample customer application consists of several, simple JSP pages that display the content and properties of two nodes from the repository. Update the `/includes/noderefs.jsp` page to provide the NodeRefs of two nodes in your repository.

By default, the sample pulls content from the Alfresco repository located at `http://localhost:8080/alfresco`, using a user name and password of `admin`. These values can be supplied using `context-param` values in the `web.xml` file, for example:

```
<context-param>
  <param-name>org.customer.alfresco.host</param-name>
  <param-value>localhost</param-value>
</context-param>

<context-param>
  <param-name>org.customer.alfresco.port</param-name>
  <param-value>8080</param-value>
</context-param>

<context-param>
  <param-name>org.customer.alfresco.context</param-name>
  <param-value>alfresco</param-value>
```

```

</context-param>

<context-param>
  <param-name>org.customer.alfresco.username</param-name>
  <param-value>admin</param-value>
</context-param>

<context-param>
  <param-name>org.customer.alfresco.password</param-name>
  <param-value>admin</param-value>
</context-param>

```

Installing and configuring Google Docs integration

Google Docs integration allows you to use Google Docs to edit document content stored in Alfresco, as an alternative to the online and offline editing capabilities in Alfresco Share.

When you use the setup wizards to install Alfresco, the Google Docs integration feature is applied and enabled for supported content in an Alfresco installation.

If you install Alfresco manually, you need to apply the Google Docs AMP files separately to enable the feature.

With Google Docs integration, you'll see new actions for creating documents, spreadsheets, and presentations. Also, you'll see an action called **Edit in Google Docs** on all supported document types.

When configuring Google Docs integration with Alfresco, you don't need to identify a 'system' Google account.

Installing Google Docs integration manually

Google Docs integration is installed by default when you install Alfresco using the setup wizards. If you are installing Alfresco manually, use these steps to install Google Docs integration.

1. Browse to the [Support Portal](#), and download the following files:

<code>alfresco-googledocs-repo-3.0.3-4ent.amp</code>	This AMP contains the Google Docs functionality that is applied to the core Alfresco repository. The AMP should be applied to the <code>tomcat/webapps/alfresco</code> directory.
<code>alfresco-googledocs-share-3.0.3-4ent.amp</code>	This AMP file contains the additional Google Docs functionality that is applied to an existing Alfresco Share user interface. The AMP should be applied to the <code>tomcat/webapps/share</code> directory.

2. Change into the root of the Alfresco installation directory. Directories specified in the following procedures are relative to this directory.
3. Move the `alfresco-googledocs-repo-3.0.3-4ent.amp` file to the `amps` directory.
4. Move the `alfresco-googledocs-share-3.0.3-4ent.amp` file to the `amps_share` directory.
5. Stop the Alfresco server.
6. Delete the `tomcat/webapps/alfresco` and `tomcat/webapps/share` folders in the Alfresco installation directory.
7. Use the Module Management Tool (MMT) to install the AMP files.

```

java -jar <installLocation>\bin\alfresco-mmt.jar install
<installLocation>\amps\alfresco-googledocs-repo-3.0.3-4ent.amp
<installLocation>\tomcat\webapps\alfresco.war

```

```
java -jar <installLocation>\bin\alfresco-mmt.jar install
<installLocation>\amps_share\alfresco-googledocs-share-3.0.3-4ent.amp
<installLocation>\tomcat\webapps\share.war
```

Alternatively, if your Alfresco installation is running in the Tomcat application server, you can use the `<installLocation>\bin\apply_amps` command to apply all AMP files that are located in both the `amps` and `amps_share` directories.

Install both Google Docs AMP files at the same time by using the `apply_amps` command:

- Linux: `bin/apply_amps.sh`
- Windows: `bin\apply_amps.bat`

The `apply_amps` command checks the version of Alfresco so that you install the relevant AMP package to the correct Alfresco version.

8. Start the Alfresco server.

Google Docs configuration properties

The following properties can be configured for Google Docs integration in the `alfresco-global.properties` file.

`googledocs.enabled`

Enables the Google Docs functionality. By default, this property is set to true. If you set this option to false, the **Edit in Google Docs** action will not be available. Documents that are currently being edited will still be available using the **Resume editing in Google Docs** action until they are saved or discarded.

`googledocs.idleThresholdSeconds`

Sets the idle time threshold in seconds. Additional Google users that you invite to collaborate on the document will be considered to be 'idle' after this period. The period is measured from the time when the user last made a change to the document. When saving documents back to Alfresco, or discarding changes, you must confirm that you want to disconnect any non-idle users before the action completes.

You can also set these properties in the Admin Console. See [Google Docs™ Console](#)

Configuring Google Docs

Google Docs Console in Admin Tools provides the settings for enabling and controlling Google Docs integration.

1. Open the Admin Console.
2. In the **Consoles** section, click **Google Docs Console**.
3. Set the properties:

Google Docs property	Example setting	What is it?
<code>googledocs.enabled</code>	true	Enables the Google Docs functionality. If you set this option to false, the Edit in Google Docs action will not be available. Documents that are currently being edited will still be available using the Resume editing in Google Docs action until they are saved or discarded.

Google Docs property	Example setting	What is it?
<code>googledocs.idleThresholdSeconds</code>	600	Sets the idle time threshold in seconds. Additional Google users that you invite to collaborate on the document will be considered to be 'idle' after this period. The period is measured from the time when the user last made a change to the document. When saving documents back to Alfresco, or discarding changes, you must confirm that you want to disconnect any non-idle users before the action completes.

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

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

Google Docs supported document types

Google Docs restricts the formats of files or documents that can be uploaded or created.

The following table shows the file format restrictions for Alfresco content that integrates with Google Docs.

File type	Description
DOC	A Microsoft Word 97-2003 document.
XLS	A Microsoft Excel 97-2003 Workbook.
PPT	A Microsoft PowerPoint 97-2003 Presentation.
DOCX	An XML-based Microsoft Word document.
XLSX	An XML-based Microsoft Excel Workbook.
PPTX	An XML-based Microsoft PowerPoint presentation.

 You can edit the `DOC`, `XLS`, and `PPT` formats in Google Docs but when you save the content back to Alfresco, you must confirm that these formats will be converted to the equivalent Microsoft Office 2007 (OOXML) formats.

Google places further restrictions on the size and complexity of documents that can be edited in Google Docs. The **Edit in Google Docs** action is not available for documents or spreadsheets larger than 2 MB and presentations larger than 50 MB. Google also prevents editing of other documents that exceed their published limits. See the published [Google size limits](#).

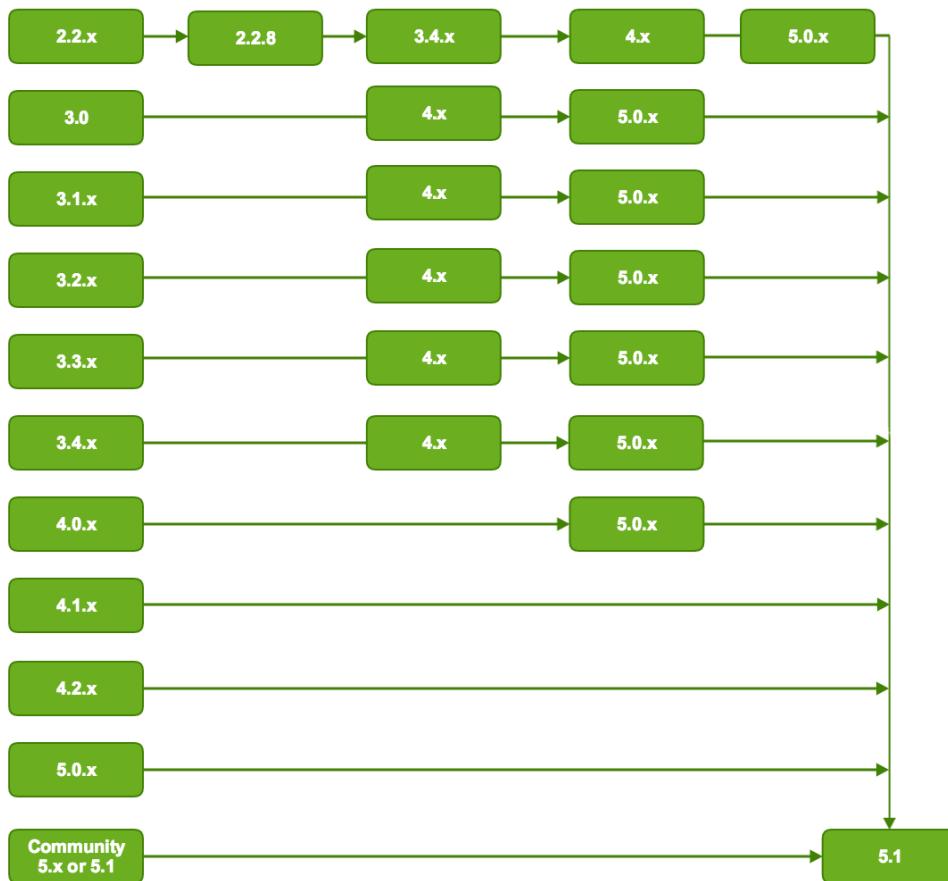
Upgrading

Use this recommended procedure for performing an upgrade.

Alfresco upgrade paths

When you upgrade Alfresco, it is recommended that you follow a structured upgrade path between versions.

The following diagram shows the upgrade paths for major versions:



The upgrade path recommendations are:

- Direct upgrades to Alfresco 5.1 are supported from only 4.1.x and later, with the latest Service Pack applied.
- Upgrades from Alfresco 4.x using Lucene require migration to Solr 1.4 before being able to upgrade to Alfresco 5.0 with Solr1.4. Once Alfresco 5.0 and Solr1.4 are running, then migration to Alfresco 5.0 with Solr4 will complete the upgrade. See [Upgrading search subsystems](#) for more information.
- Upgrades from Alfresco 3.x require the latest service pack of version 4.x before upgrading to version 5.0, and then upgrading to version 5.1.
- Upgrades from Alfresco 2.2.x require the service pack 2.2.8 to be applied first, followed by the latest service pack of version 3.4.x, and then the latest service pack of version 4.x before being able to upgrade to version 5.0.

-  If you are upgrading from an earlier release that is not shown on this diagram, contact Alfresco Support for assistance.

QuickStart upgrade guide

Use this information to upgrade Alfresco on a single instance and in a distributed and clustered environment.

Follow this checklist when upgrading or clustering an installation of Alfresco. For detailed step-by-step instructions for upgrading Alfresco, see [Upgrading Alfresco general procedure](#).

When upgrading Alfresco, in order to configure distribution and clustering optimally, contact [Alfresco Consulting](#) or your Alfresco certified partner.

Upgrading Alfresco on a single instance

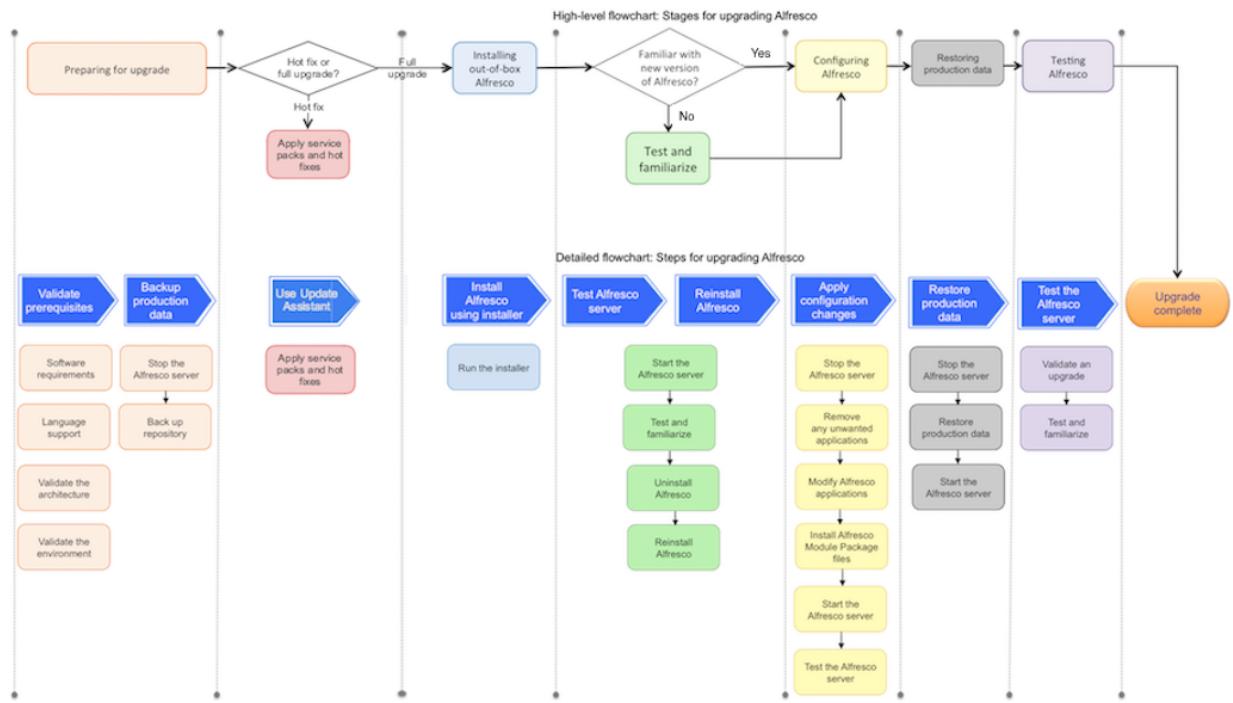
Use this information to upgrade a single instance of Alfresco.

The main stages involved in upgrading and configuring Alfresco are shown in the diagram. These include preparing your system for upgrade, installing Alfresco as an out-of-box application, configuring it based on your requirements, restoring production data, and finally, testing and getting familiar to Alfresco.

Each of these main stages consist of sub-steps, as shown in the diagram, which displays the sub-steps that need to be performed in order to complete each main stage.

-  Note that the steps shown in the diagrams have a colour code. For example, Restoring production data stage consists of three sub-steps, namely, Stop the Alfresco server, Restore production data, and Start the Alfresco server.
-  Alfresco recommends that you upgrade Alfresco in a test environment before you upgrade it in your production environment. This allows you to address any problems during the upgrade process more effectively. You can also verify that applications and scripts work properly before upgrading your production environment. In addition, you can assess the time that it takes to upgrade the database, to finalize your upgrade plan.

To get started quickly with upgrading a single instance of Alfresco, follow the process shown.



1. [Software requirements](#) on page 35
2. [Language support](#) on page 35
3. [Validating the architecture](#) on page 36
4. [Validating the environment](#) on page 38
5. [Stopping the Alfresco server](#) on page 451
6. [Backing up and restoring the repository](#) on page 612
7. [Applying Service Packs and Hot Fixes](#) on page 117
8. [Installing Alfresco using setup wizards](#) on page 12
9. [Starting the Alfresco server](#) on page 451
10. [Test and familiarize after upgrading Alfresco](#)
11. [Uninstalling Alfresco on Linux](#) on page 59
12. [Installing Alfresco using setup wizards](#) on page 12
13. [Stopping the Alfresco server](#) on page 451
14. [Tailoring your Alfresco installation](#) on page 48
15. [Customizing Alfresco applications](#) on page 141
16. [Installing an Alfresco Module Package](#) on page 48
17. [Starting the Alfresco server](#) on page 451
18. [Test the Alfresco server after customizing an upgrade](#)
19. [Stopping the Alfresco server](#) on page 451
20. [Restoring production data](#) on page 126
21. [Starting the Alfresco server](#) on page 451
22. [Validating an upgrade](#) on page 125
23. [Test and familiarize after upgrading and configuring Alfresco](#)

Upgrading Alfresco in a distributed environment

Use this information to understand how to upgrade Alfresco in a distributed environment.

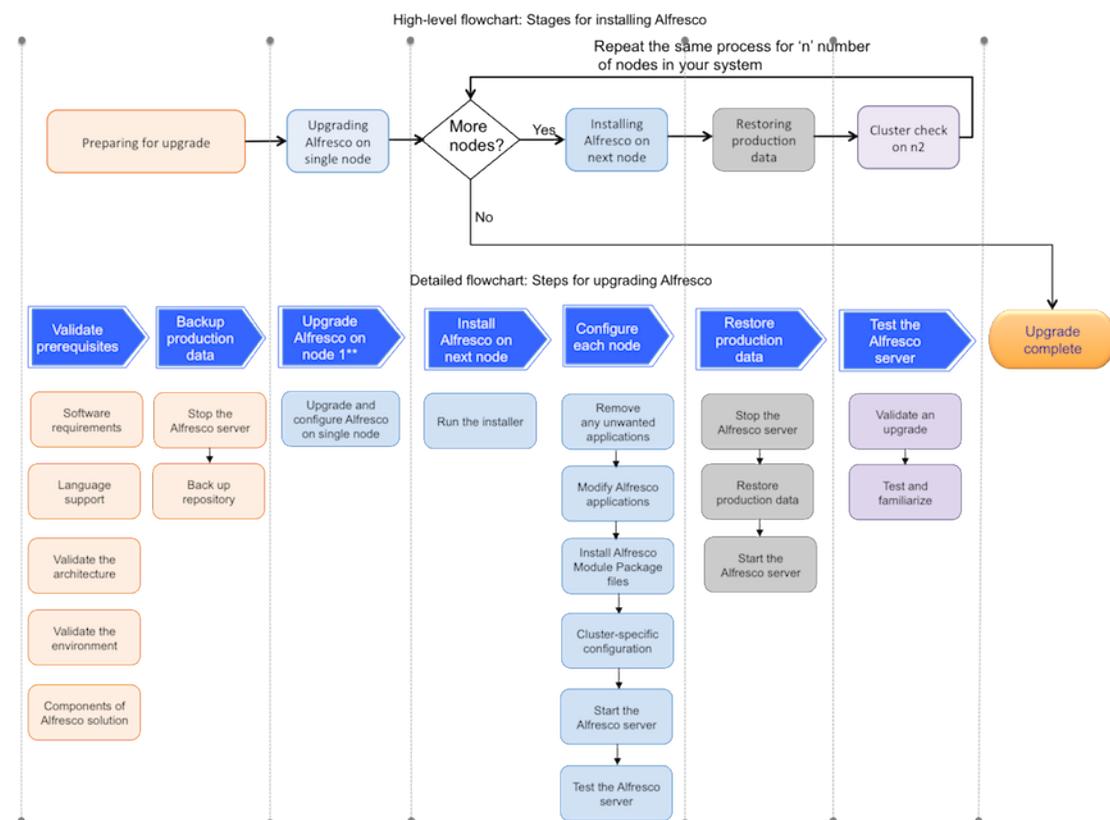
The main stages involved in installing Alfresco in a cluster are shown in the diagram. You must upgrade and configure your data on a single node first and then on the second node, and so on.

The main steps involved in the upgrading process include preparing your system for upgrade, [upgrading Alfresco on a single node](#), installing Alfresco on node 2, restoring production data, and finally, testing and getting familiar to Alfresco. Repeat the last three steps on all the other nodes in your system in series.

Each of these main stages consist of sub-steps, as shown in the diagram, which displays the sub-steps that need to be performed in order to complete each main stage.

-  Note that the steps shown in the diagrams have a colour code. For example, Restoring production data stage consists of three sub-steps, namely, Stop the Alfresco server, Restore production data, and Start the Alfresco server.
-  Make sure you do not install and configure all the nodes in parallel. Follow in the installation process in series for all the nodes in your system.
-  Alfresco recommends that you upgrade Alfresco in a test environment before you upgrade it in your production environment. This allows you to address any problems during the upgrade process more effectively. You can also verify that applications and scripts work properly before upgrading your production environment. In addition, you can assess the time that it takes to upgrade the database, to finalize your upgrade plan.

To get started quickly with upgrading Alfresco in a distributed environment, follow this process:



1. [Software requirements](#) on page 35
2. [Language support](#) on page 35
3. [Validating the architecture](#) on page 36
4. [Validating the environment](#) on page 38
5. [Components of an Alfresco solution](#) on page 466
6. [Stopping the Alfresco server](#) on page 451
7. [Backing up and restoring the repository](#) on page 612
8. [Upgrading Alfresco on a single instance](#) on page 111
9. [Installing Alfresco using setup wizards](#) on page 12
10. [Tailoring your Alfresco installation](#) on page 48
11. [Customizing Alfresco applications](#) on page 141
12. [Installing an Alfresco Module Package](#) on page 48
13. [Cluster-specific configuration](#) on page 12
14. [Starting the Alfresco server](#) on page 451
15. [Test and familiarize after upgrading Alfresco in a cluster](#)
16. [Stopping the Alfresco server](#) on page 451
17. [Restoring production data](#) on page 126
18. [Starting the Alfresco server](#) on page 451
19. [Validating an upgrade](#) on page 125
20. [Test and familiarize after upgrading and configuring Alfresco in a cluster](#)

Upgrade prerequisites checklist

This checklist describes the requirements and prerequisites necessary to begin planning for upgrading an existing Alfresco version to Alfresco One 5.0.

Before starting an upgrade:

- Validate your requirements.
 - Validate your platform is still on the supported stacks for the new version of Alfresco. See [Supported stacks](#).
 - Validate the [software requirements](#).
 - Validate the [language support](#).
 - Validate the [architecture](#).
 - Validate the [environment](#).
- [Backup your production data](#).
 - You must perform a test upgrade using a backup copy of the repository before attempting to upgrade your production environment. Therefore, it is important that your backups are up-to-date.
 - Ensure that you have backed up your production environment, for example, back up your database and content store (`alf_data` directory).
- If you are upgrading to Alfresco One 5.0, migrating from your existing search subsystem to the Solr 4 search subsystem ensures that you have access to the full search capabilities. For more information, see the [Solr 4 migration documentation](#).

- If you have any customizations (for example, AMPs) in your existing Alfresco installation, recompile all Java code against the new version of Alfresco and regression test against the new version of Alfresco.
- When you upgrade Alfresco with Oracle, the Alfresco user needs more privileges than connect and resource. At minimum, the Alfresco user should have permission to delete objects. A safer option is to give a sysdba role for the upgrade process only. After the upgrade, this role should be removed.

Database considerations

Large repositories require some additional consideration during an upgrade, such as optimization of the database and adding optional indices to the database for metadata queries.

Two important aspects to consider when upgrading a large repository are:

1. Transactional metadata query is a feature that requires the creation of new indices. For large repositories, this process may take a long time. See [Transactional metadata query](#) for more details.
2. After restoring the production data of large repositories and creating the indices, refer to [Database validation - Maintenance and Tuning](#) to ensure optimal performance.

After applying the patches, check that the logs show no warnings or issues with the database. If the indices could not be created, some queries may run very slow.

Upgrading Alfresco

In-place upgrade of the Alfresco binaries and configuration is not recommended. Creating a new installation ensures that if anything goes wrong during the upgrade, the original (not upgraded) system is still intact and available for immediate restart.

These steps assume that you have an existing Alfresco installation (`alfresco-v.1`) with the following settings:

File Name	Properties
<code>alfresco-global.properties</code>	<code>dir.root=/alfresco-v.1/alf_data</code> <code>db.url=url<v.1></code>
<code>solrcore.properties</code>	<code>data.dir.root=/alfresco-v.1/solr/</code> <code>myindexes</code>

1. Install the new version of Alfresco.
 - a. Shut down your existing Alfresco instance.
 - b. Back up your existing Alfresco repository (`alfresco-v.1`) and the database. See [Backing up the Alfresco repository](#).
 - Back up any configuration overrides from the `<extension>` directory.
 - c. Use the setup wizard/installer to install the new version (`alfresco-v.2`) of Alfresco into a different directory from the existing installation. See [Installing Alfresco using setup wizard](#).

For example, the new Alfresco installation will have the following settings:

```
In alfresco-global.properties:
dir.root=/alfresco-v.2/alf_data
db.url=url<v.2>

In solrcore.properties:
data.dir.root:/alfresco-v.2/solr/myindexes
```

2. Validate the new Alfresco One 5.1 installation to check that it is working correctly.

- a. Configure the new installation with a new repository and database (not the existing one).

- b. [Start Alfresco](#) and validate that the system works correctly.

For more information, see [Validating the upgrade](#).

3. Apply all customizations to the new Alfresco One 5.1 installation.

- a. [Stop](#) the Alfresco server.

- b. [Remove](#) any unwanted applications.

- c. [Modify](#) Alfresco applications.

- d. Install the required AMP files. See [installing an Alfresco Module Package](#).

- e. Do not copy the files. Copy only the override settings so that you will not overwrite the new extension files in the upgraded version.

- f. [Start](#) the Alfresco server.

Monitor the startup log messages for information on the status of the upgrade. If any issue(s) occur in the logs during startup, you need to rollback the whole repository to fix the issue(s) and then try again.

- g. Fully [test](#) the working and configuration of your customizations.

- h. [Stop](#) the Alfresco server.

4. Restore production data.

- a. Remove all the files and directories under the `contentstore` directory of the new installation. Also, delete the database.

- b. Delete the files in the two `Solr alfrescoModels` directories, and the indexes in the two directories (`solr/workspace/` and `solr/archive/`) of the new installation.

- c. Restore the backup of the indexes, `contentstore` directory, files, and database from your previous Alfresco installation into the new installation. See [restoring production data](#).

- d. [Start](#) the Alfresco server.

If any issue(s) occur in the logs during startup, you need to rollback the whole repository to fix the issue(s) and then try again.

5. If you are happy with the upgraded system, remove the old Alfresco installation and repository.

6. [Optional] Perform this additional step only if you have configured multi-tenancy and are upgrading Alfresco.

If upgrading to the latest Alfresco version, your existing MT sample extension files are no longer relevant and must be deleted. It is also recommended that you backup your existing MT files.

- a. Take a backup of the following three existing MT extension files and delete them from the existing MT extension directory:

- `alfresco/extension/mt/mt-context.xml` to `alfresco/extension/mt/mt-context.xml`
- `alfresco/extension/mt/mt-admin-context.xml` to `alfresco/extension/mt/mt-admin-context.xml`
- `alfresco/extension/mt/mt-contentstore-context.xml` to `alfresco/extension/mt/mt-contentstore-context.xml`

7. [Optional] Perform this step if you are working in a clustered environment:

- a. Shut down all nodes in the cluster.

- b. Perform steps 1 to 5 on each additional node in turn, ensuring that each node starts fully before restarting the next one.

You need to copy the database once only as it is upgraded by the first node that is upgraded. The other nodes detect it has been upgraded and skip the database upgrade step.

Applying Service Packs and Hot Fixes

The Update Assistant gives you the flexibility of applying current Service Packs or Hot Fixes to Alfresco, without the need to reinstall.

The Update Assistant runs from the command line, and is included in the package that contains your Service Pack or Hot Fix. It can be downloaded from the [Alfresco Support Portal](#).

Before running the Update Assistant, you must ensure that you have:

- A full, validated backup of your Alfresco repository before running the Update Assistant. This includes the Alfresco configuration, database, content store, and indexes.
- Stopped your Alfresco server.

The Update Assistant is available with Service Packs and Hot Fixes that you apply to 5.1 and later.

You can't use the Update Assistant to upgrade from one major or minor Alfresco version to another. For example, you can't upgrade from Alfresco One 5.0 to Alfresco One 5.1, but you can move from 5.1.0 to Service Pack 5.1.1. You can also move, for example, from 5.1.0 to Hot Fix 5.1.0.100.

You can apply only a newer Service Pack or Hot Fix than the one that is already installed. For example, you can't downgrade from 5.1.1 to 5.1.0, or from 5.1.0.100 to 5.1.0.99.

Prerequisites for using the Update Assistant

Ensure that you understand the framework for using the Update Assistant before you run the tool.

The Alfresco Update Assistant requires the following framework:

- A standard Alfresco installation performed using one of the following:

For Alfresco One:

- Alfresco One Installer
- Alfresco One Share Installer
- Alfresco One Platform Installer
- The standard Tomcat application server, as shipped with Alfresco (with any supported database)
- Java (the same version as you use in Alfresco or later)
- The JAVA_HOME environment variable must contain the path of your Java installation

The Update Assistant performs:

- Updates to the Alfresco, Share, and Solr WAR files
- Updates to the modules shipped as standard with Alfresco (for example, JAR or AMP files)

All customizations must be packaged as either:

- AMP files that you reapply after the update
- JAR files declared on the classpath

For example, AMP files that you have applied to the Alfresco and Share WAR files must reside in the `amps` and `amps_share` folders. Customisations must not be placed in the unpacked (*exploded*) WAR directories, because the WAR files and their unpacked directories are overwritten or removed as part of the update process.

Installing Service Packs and Hot Fixes using the Update Assistant

Use this information to apply Service Packs or Hot Fixes to your Alfresco installation.

1. Stop the Alfresco server, and shut down all nodes in a cluster.
2. Download `alfresco-one-update-package-201603.zip` from the [Alfresco Support Portal](#), and unzip the contents to a folder; for example, to your `Downloads` directory.

For example, if you are downloading a 5.1.4.111 Hot Fix, you would download:

```
alfresco-one-update-package-5.1.4.111.zip
```

The Update Assistant is packaged with the Service Pack or Hot Fix that you are applying.

The package contains the following items:

- `lib` directory, containing the `alfresco-update-tool.jar` file
 - `resources` directory, containing the Service Pack or Hot Fix for you to apply
 - `apply_updates.sh` file
 - `apply_updates.bat` file
 - `README.txt`
3. Perform a full backup of your Alfresco repository.
This includes the Alfresco configuration, database, content store, and indexes.
 4. Check the `README` file to determine whether any third party components; for example, Java or Tomcat, are affected by the new Service Pack or Hot Fix.
Update the third party component to the required software level, as indicated in [Supported Platforms](#), before running the Update Assistant.
 5. Verify that all AMP files that you have applied to the Alfresco and Share WAR files are in the `amps` and `amps_share` folders.
This is because the Update Assistant reapplies the AMPs to the updated WAR files.
 6. From a command line, navigate to the folder where you extracted the package (for example, `Downloads`) and run the `apply_updates` command relevant to your operating system, specifying your Alfresco home directory (`ALF_HOME`).

The format is as follows:

```
apply_updates [options] <Alfresco installation directory>
```

These are the options:

- `-f` or `--force`: forces the update. This option is required, for example, if you are downgrading an AMP file
- `-h` or `--help`: provides a listing of all the update options
- `-v` or `--version`: displays the version number of the Update Assistant
- `-y` or `--assumeyes`: performs the update without prompts (assuming that the answers are `yes`, i.e; that you have a full, validated backup, and that you are happy to proceed with the update)

For example, to run the Assistant without prompts:

- For Windows: `apply_updates.bat -y C:\alfresco-one`

- For Linux: `./apply_updates.sh -y /Applications/alfresco-one`

Using the help option, `apply_updates --help`, gives you a list of the options.

The Update Assistant creates a directory in the Alfresco home directory called `applied-updates`, where it stores working files including logs, a backup of any Alfresco files that were changed, and a copy of the update. If required, you can delete these directories after you've validated the update.

7. Type `y` (Yes) or `n` (No) when you are prompted to confirm that you have a full, validated Alfresco backup.

If you answer `n`, the Update Assistant will exit, and will not apply the updates.

If you have specified `-y` or `--assumeyes` in [6](#) on page 118, you will not be prompted, and it is assumed that you have a backup.

8. Type `y` (Yes) or `n` (No) when you are prompted to confirm that you want to continue with the update.

The Update Assistant has verified the package and is ready to update your system. If you answer `n`, the Update Assistant will not apply the updates.

9. When you see a message specifying that the update was applied successfully, the Update Assistant has completed the update.

The contents of the applied package, including the Update Assistant itself, are copied to the `applied-updates` directory. All changed files are backed up to the `applied-updates/<date and timestamp>/backups` directory.

The `applied-updates` directory, and its contents are created to allow you to check the updates that have been made to your system. It is safe to remove this directory after the Assistant has run successfully, and you have verified the updates.

Check the output from the script to ensure that the Service Pack or Hot Fix has been applied successfully. For more information on log messages, see [Troubleshooting the Update Assistant](#) on page 119.

In the event of a problem, the Update Assistant attempts to roll back to the previous Alfresco configuration, but does not alter the contents of the Alfresco repository to either back it up or restore it.

10. Restart the Alfresco server.

11. Open Alfresco Share, and click the Alfresco One logo at the bottom of the page to check that the Alfresco installation number has incremented.

For example, from Alfresco One 5.1.1 to Alfresco One 5.1.2.

Troubleshooting the Update Assistant

The Update Assistant log files can help you determine the status of your update and help with any problems.

The Update Assistant logs are located in `applied-updates/<date and timestamp>/logs/update-assistant.log` in your Alfresco home directory, for example:

`Alfresco/applied-updates/2015-11-29-13-54-17/logs/update-assistant.log`

Use these steps to assist with the messages that you find there.

1. A message says that the update package has not been found.

The Update Assistant looks for the `alfresco-update-tool` jar file. Check that the file exists in the package that you downloaded in [Installing Service Packs and Hot Fixes using the Update Assistant](#) on page 118.

2. A message says that I do not have the right version of Java.

Make sure you are running the level of Java required, or later.

3. A message says that it can't find the required files.

The Update Assistant must be run from the directory where you extracted your `zip` or `tgz` file, as it relies on files in that directory to run.

4. A message says that my application server is running.

You must stop your Alfresco application server before running the Update Assistant. Updates will not be made if the application server is running.

5. A message says that I do not own the Alfresco installation directory.

You must be the directory owner (and have write permissions) to run the Update Assistant.

6. A message asks me whether I have a full, validated backup of my Alfresco repository.

If you answer `N` (No), the Update Assistant will stop and no updates are applied. It is important that you have a full, validated backup before you start. The Update Assistant will not perform the backup for you.

7. A message says that I cannot update to a certain version of Alfresco.

Ensure that you are applying a Service Pack or Hot Fix, and do not attempt to upgrade from one Alfresco version to another using the Update Assistant.

8. A message says that I do not have enough disk space for the installation.

The disk space required is the sum of the size of the Alfresco and Share WAR files in the `webapps` directory and the update package, plus the AMPs in the existing `amps` and `amps_share` directories.

9. A message says that the Update Assistant can't find the host and port for the application server.

The Update Assistant can't determine whether Tomcat is running, because either:

- It can't find the `alfresco-global.properties` file, which defines the host and port properties
- Or the host or port properties in the `alfresco-global.properties` file are missing or corrupt

10. My update was unsuccessful. What next?

In the event of a problem, the Update Assistant attempts to roll back to the previous Alfresco configuration, but does not alter the contents of the Alfresco repository to either back it up or restore it.

Take another look at the log to see if there is any information on why the update was unsuccessful. If there is no further information, contact Alfresco Support.

11. I've validated that my update was successful. Can I delete the `applied-updates` directory in my Alfresco home directory?

Yes, if you are happy with the update, you can delete any files and directories that were created by the Update Assistant.

Upgrading search

Use this information to migrate the search subsystem during an upgrade to Alfresco One 5.1.



The Lucene search subsystem is not available in Alfresco One 5.1.



During an upgrade, Solr 4 needs to reindex the entire repository. While reindexing is in progress, you may use Solr 1 for basic search functionality - new functionality enabled by

Solr 4 (such as filtered searches) will not be available, and you may encounter other issues with search capabilities.

This information describes the migration path of the following two examples:

- Upgrading from Alfresco 4.x with Lucene to Alfresco One 5.0 with Solr 4
- Upgrading from Alfresco 4.x with Solr 1.4 to Alfresco One 5.0 with Solr 4

Issues to consider before upgrading search

Before beginning an upgrade of the search subsystems, there are some important issues you should consider.

Setup wizard installation

Alfresco 5.0 uses the Solr 1 search subsystem only during the upgrade process. When upgrading to Alfresco One using the setup wizards, you should install both the Solr 1 and Solr 4 search subsystems. If you wish to minimize the necessary downtime of the search subsystem while the Solr4 indexes are being build, you must run Solr 1. Once the Solr 4 indexes are up to date, you must enable the Solr 4 subsystem and disable the Solr 1 subsystem.

-  Note that when you have both the subsystems, you will need more memory. So, you might consider installing them as separate web applications on separate Tomcat instances.
-  You do not have to use the Solr 1 search service during the upgrade process. Instead, you can let Solr 4 build its indexes, but during this time, any search carried out might return incomplete results. This is because only those documents that have been indexed are available for searching. Set the `NoIndex` option to avoid incomplete and/or misleading results. For more information, see [transactional metadata query](#).

Solr 4 suggester configuration

Alfresco uses the suggester component in Solr to provide users with automatic suggestions for query terms.

- With new Alfresco One 5.0 installations, the suggester is enabled for the workspace store, by default.
- If you are upgrading to Alfresco One 5.0 with Solr 4, before building the new index, Alfresco recommends that you disable the suggester property in the `<solrRootDir>/workspace-SpacesStore/conf/solrcore.properties` file:

```
solr.suggester.enabled=false
```

This is because on a low specification system, building the suggester can cause CPU and IO load issues, which can affect other operations. When the Solr 4 index is up to date, the `solr.suggester.enabled` property can be reset to `true`.

The Solr 4 suggester holds a view of the index. Normally, there is only one live view of the index. An old view can exist for a few seconds or minutes until all the running queries are complete. If there are two or more live views of the index, index tracking will not run. This only happens while the suggester is being build and if the process of building the suggester is slow. To solve this issue, you can configure how often the suggester will run, if it is enabled.

An out-of-the-box Alfresco application allows you to use three word phrase suggestions across the repository. Suggestions are not limited by permissions. To limit the scope, you can configure the suggester to use single words or two word phrases by changing `schema.xml` before you rebuild the index.

Upgrading from Lucene to Solr 4 search

You can upgrade from Alfresco Enterprise 4.x with the Lucene

search server to Alfresco One 5.0 with the Solr 4 search server.

 In this documentation, we are referring to **Solr 1.4** search subsystem as **Solr**.

1. For versions prior to Alfresco 4.x, upgrade to Alfresco 4.x and continue to use the Lucene search subsystem as before.
2. Install and configure Solr on Alfresco 4.x to track the repository.
For more information, see [Installing and configuring Solr](#).
3. Monitor progress using the `SUMMARY` report.
`http://localhost:8080/solr/admin/cores?action=SUMMARY&wt=xml`
4. When the Solr index is updated as reported by the `SUMMARY` report, enable the Solr subsystem and disable the Lucene subsystem.
5. Follow the instructions for [upgrading from Solr to Solr 4 search](#).

Upgrading from Solr 1.4 to Solr 4 search

Use this information to upgrade from Alfresco Enterprise 4.x with the Solr 1.4 search server to Alfresco One 5.0 with the Solr 4 search server.

To determine the current search server, navigate to the `Search Manager` page at **Alfresco Share Admin Console > Repository Services > Search Service**. Select the search subsystem from the **Search Service In Use** list.

Follow the steps to migrate from Alfresco Enterprise 4.x with Solr 1.4 search service to Alfresco One 5.0 with Solr 4 search service.

1. Upgrade to Alfresco One 5.0 and continue to use the Solr 1.4 search service as before.
For information on migrating the Solr 1.4 indexes with Alfresco One 4.x to Solr 1.4 with Alfresco One 5.0, see [Upgrading Solr 1.4 search service](#).
2. Configure Solr 4 to track the repository. For details, see the [Installing and Configuring Solr 4](#) topic.
3. While Solr 4 builds its indexes, you can monitor progress using the `SUMMARY` report.
`http://localhost:8080/solr4/admin/cores?action=SUMMARY&wt=xml`

For details, see the [Unindexed Solr Transactions](#) topic.

4. Optionally, you can use the Solr Admin Web interface to view Solr configuration details, run queries, and analyze document fields.
 - a. Open the FireFox **Certificate Manager** by selecting **Firefox > Preferences... > Advanced > Certificates > View Certificates > Your Certificates**.
 - b. Import the browser keystore `browser.p12` that is located in your `<ALFRESCO_HOME>/alf_data/keystore` directory.
 - c. Enter the password `alfresco`.
A window displays showing that the keystore has been imported successfully. The **Certificate Manager** now contains the imported keystore with the Alfresco repository certificate under the **Your Certificates** tab.
 - d. Close the **Certificate Manager** by clicking **OK**.

- e. In the browser, navigate to a Solr URL.

For example, use <http://localhost:8080/solr> for Solr and <http://localhost:8080/solr4> for Solr 4.

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.

- f. Expand **I understand the risks**.

- g. Select **Add Exception**.

- h. Click **View**.

This displays the certificate.

- i. Confirm that the certificate was issued by Alfresco Certificate Authority, and then confirm the **Security Exception**.

Access to Solr 1.4/Solr 4 is then granted. The Solr 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, there is a sidebar with links for Dashboard, Logging, Core Admin, Java Properties, Thread Dump, and a dropdown for 'alfresco'. Below this is an 'Overview' section with links for Analysis, Dataloaders, Documents, Filters, Ping, Plugins / Stats, Query, Replication, and Schema Browser. The main content area has three tabs: 'Statistics', 'Instance', and 'Alfresco Core - Summary Report'. The 'Statistics' tab shows basic stats like Num Docs (803), Max Doc (931), and Heap Memory Usage (1093688). The 'Instance' tab shows CWD, Instance, Data, Index, and Impl details. The 'Alfresco Core - Summary Report' tab shows Alfresco-specific metrics like Nodes in Index (798), Transactions in Index (17), and FTS Status Clean (163).

5. Monitor the progress of both the Solr 1.4 and Solr 4.0 subsystems via the JMX client or the **SUMMARY** report.
- ⚠** Do not use the Alfresco Share **Admin Console** tool to monitor the status of the subsystems as it will change the subsystem used for query. Only use the JMX client.
6. When the index is updated as reported by the **SUMMARY** report, you can use the **REPORT** option and check the following:

- In the REPORT option, node count should match the number of live nodes in the repository (assuming nothing is changing and the index is updated). The index contains a document for failed nodes, so failures need to be considered separately.
- Any missing transactions; if there are issues, use the FIX option.

`http://localhost:8080/solr4/admin/cores?action=FIX`

For more information, see the [Troubleshooting Solr Index](#) topic.

- Find errors with specific nodes using `DOC_TYPE:ErrorNode` option.

`https://localhost:8446/solr4/alfresco/afts?q=DOC_TYPE:ErrorNode`

- If there are any issues, use the REINDEX option with the relevant node id.

`http://localhost:8080/solr4/admin/cores?action=REINDEX&txid=1&acltxid=2&nodeid=3&aclid=4`

For more information, see the [Troubleshooting Solr Index](#) topic.

- When the Solr 4 index is updated, you must enable the Solr 4 subsystem and disable the Solr 1.4 subsystem.
- (Optional) To decommission (now redundant) Solr 1.4, follow the steps below:
 - Stop the Solr 1.4 search service.
 - Delete the `solr` directory from `<ALFRESCO_HOME>/tomcat/webapps`.
 - Delete the `solr.xml` file from `<ALFRESCO_HOME>/tomcat/conf/Catalina/localhost`.
 - Delete the `solr` directory from `<ALFRESCO_HOME>/alf_data`.

Upgrading the Solr 1.4 search service

In order to upgrade to the Solr 4 search service, whilst the Solr 4 indexes are being built, you must transition from a previous version of Alfresco (for example, Alfresco One 4.2.x) with the Solr 1.4 search service to Alfresco One 5.0 with the Solr 1.4 search service.

- Install Alfresco One 5.0 with both the Solr 1.4 and Solr 4 search services.
- From your old Solr 1.4 installation, copy all the indexes to the new Solr 1.4 installation.
 - Copy the `alf_data/solr/workspace/SpacesStore/index` directory from the old Solr 1.4 installation to the `alf_data/solr/workspace/SpacesStore/index` directory of the new Solr 1.4 installation.
 - Copy the `alf_data/solr/archive/SpacesStore/index` directory from the old Solr 1.4 installation to the `alf_data/solr/archive/SpacesStore/index` directory of the new Solr 1.4 installation.
- Reapply the configuration changes made to the `solrcore.properties` file of the old Solr 1.4 installation to the `solrcore.properties` file of the new Solr 1.4 installation.
 - Reapply the changes made to the `alf_data/solr/workspace-SpacesStore/conf/solrcore.properties` file from the old Solr 1.4 installation to the `alf_data/solr/workspace-SpacesStore/conf/solrcore.properties` file of the new Solr 1.4 installation.
 - Reapply the changes made to the `alf_data/solr/archive-SpacesStore/conf/solrcore.properties` file from the old Solr 1.4 installation to the `alf_data/solr/archive-SpacesStore/conf/solrcore.properties` file of the new Solr 1.4 installation.
- Ensure that your Alfresco One 5.0 instance is set to use Solr 1.4 search service during this process.

To validate that your old Solr 1.4 indexes and configuration changes are correctly copied over to the new Solr 1.4 installation, follow these steps:

1. Run Alfresco One 4.x with the old Solr 1.4 installation and Alfresco One 5.0 with the new Solr 1.4 installation at the same time.
2. Generate the `SUMMARY` report for both the old and the new Solr indexes.
3. Compare the two `SUMMARY` reports to ensure that both have the same number of nodes, transactions, and ACLs.

Validating an upgrade

1. Restart the Alfresco server.
The configuration overrides ensure the server immediately directs data to the appropriate locations.
2. Monitor the startup log messages for information on the status of the upgrade.
3. Validate the new installation using a blank repository.
4. Configure the new installation with a new repository (not the existing one).
5. Verify the database connection details and Alfresco data folder locations are set according to the environment in which the server is running.
6. Start Alfresco and validate the system works correctly.
7. Shut down Alfresco.
8. When you are certain the new installation is thoroughly validated, remove the old Alfresco installation and repository.

Testing an Alfresco upgrade

Testing an upgrade checks that Alfresco is successfully upgraded and is working as expected after the upgrade.

Test and familiarize after upgrading Alfresco

You have successfully upgraded Alfresco. Now test that the core features and functionalities of Alfresco that you intend to use work as expected.

Here are some of the tips to help you familiarize yourself with Alfresco.

-  Alfresco recommends that you create one or two test sites for testing purpose and put all your test data in those sites. After finishing the tests, you can delete the test sites in order to clear your database.
 - Check if the roles users had in the previous Alfresco version are still valid in the new upgraded Alfresco.
 - Check if your data or document in the previous Alfresco version are available in the new upgraded Alfresco.

Test the Alfresco server after customizing an upgrade

There are a number of tests that you can perform in Alfresco after customizing an upgrade.

- Make sure that the Alfresco server is up and running.
- Make sure that the errors in the `alfresco.log` file. are checked and understood.

Test and familiarize after upgrading and configuring Alfresco

You have successfully upgraded and configured Alfresco. Now make sure that the features and customizations you have added are operational.

Here are some of the tips to help you test your Alfresco customizations.

-  Alfresco recommends that you create one or two test sites for testing purpose and put all your test data in those sites. After finishing the tests, you can delete the test sites in order to clear your database.
 - Check if the users or groups created previously, still exist.
 - Check if all the dashboards created previously, still exist.
 - Check if the folders in the document library that were created prior to the upgrade, still exist.

Test and familiarize after upgrading Alfresco in a cluster

You have successfully upgraded and configured Alfresco in a distributed/clustered environment. Now make sure that the features and customizations you have added are operational.

Here are some of the tips to help you test your Alfresco customizations.

-  Alfresco recommends that you create one or two test sites for testing purpose and put all your test data in those sites. After finishing the tests, you can delete the test sites in order to clear your database.
 - Check that various Alfresco components are communicating with each other.
 - For a clustered upgrade, check if one node is down, check if the request is forwarded to the next available node.

Test and familiarize after upgrading and configuring Alfresco in a cluster

You have successfully upgraded and configured Alfresco in a distributed/clustered environment. Now make sure that the features and customizations you have added are operational.

Here are some of the tips to help you test your Alfresco customizations.

-  Alfresco recommends that you create one or two test sites for testing purpose and put all your test data in those sites. After finishing the tests, you can delete the test sites in order to clear your database.
 - Check if the users or groups created previously, still exist.
 - Check if all the dashboards created previously, still exist.
 - Check if the folders in the document library that were created prior to the upgrade, still exist.
 - Check if clustering is working properly by running the [cluster validation tool](#) in the Admin Console.

Restoring production data

Follow these steps to restore production data.

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.

1. Restore the backup into the new repository.

If Solr is being used, put the following directories from your backup to the `dir.root` directory on a new Alfresco 4.2 instance.

- `contentstore` directory

- `solr/workspace` directory (optional)
- `solr/archive` directory (optional)
- `contentstore.deleted` directory

Some of the above mentioned directories are optional. This is because if the indexes are not copied over from the previous Alfresco installation, Solr will query Alfresco and rebuild its index in background after the startup. It may take more time to rebuild indexes on large repositories. Alfresco applications will be accessible during reindex process.

2. Point the new Alfresco deployment to the old database via the `db.*` properties in `alfresco-global.properties` by providing the JDBC URL, database name, login credentials, and any other relevant configuration options. Remember to specify the relevant JDBC driver into your application server's classpath.

Configuring

Follow these links if you are configuring integrations or modules that are external to Alfresco:

- [Installing and configuring Alfresco Analytics](#)
- [Installing and configuring Alfresco Outlook Integration](#)
- [Installing and configuring Alfresco Media Management](#)
- [Configuring Alfresco Records Management](#)

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.

Use the following methods to configure Alfresco:

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

 If you use multiple methods to configure Alfresco, updates made using a JMX client will override any other settings, and updates in the Admin Console and Admin Tools override settings in `alfresco-global.properties`. These settings also persist in the database, and are not reflected back into the `alfresco-global.properties` file.

Alfresco Admin Console

The Alfresco Admin Console is an administrator's tool to manage your Alfresco configuration. You can run the Admin Console from a browser without having to start Alfresco Share. See [Using the Admin Console](#) for more information.

Share Admin Tools

Share Admin Tools is an administrator's tool to create and manage users and groups from Share, set application preferences, manage categories and tags, and browse the system information in the node browser. See [Using the Share Admin Tools](#) on page 133 for more information.

Global properties file

The global properties file (`alfresco-global.properties`) is used by Alfresco to detect extended properties. For example, when you install Alfresco, many of the installation settings are saved in the global properties file. The global properties file is used by Alfresco to detect the extended properties. You can use the global properties to set all your property settings; whenever you make a change, you must restart the Alfresco server to apply those changes. See [Using the alfresco-global.properties file](#) on page 134 for more information.

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 in 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. See [Using a JMX client to change settings dynamically](#) on page 136 for more information.

Using the Admin Console

The Admin Console 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:

- System Summary: [Viewing the System Summary](#) on page 130
- Consoles:
 - GoogleDocs: [Configuring Google Docs](#) on page 108
 - Model and Messages: [Managing models using the Admin Console \(Enterprise ONLY\)](#)
 - Tenant: [Managing tenants](#) on page 480
 - Workflow: [The Workflow Console](#) on page 488
- Email Services:
 - Inbound Email: [Managing inbound emails](#) on page 367
 - Outbound Email: [Managing outbound emails](#) on page 370
- General:
 - License: [Uploading a new license](#) on page 463
 - Repository Information: [Viewing Repository Information](#) on page 132
 - System Settings: [Configuring server administration settings](#) on page 342
- Repository Services:
 - Activities Feed: [Configuring the Activities Feed](#) on page 374
 - Repository Server Clustering: [Managing members of a cluster](#) on page 473
 - Index Server Sharding: [Configuring Solr sharding using the Admin Console](#) on page 312
 - Process Engines: [Enabling workflow process engines](#) on page 491
 - Replication Service: [Enabling the Replication Service](#) on page 593
 - Search Service: [Working with the Search Service](#) on page 277
 - Subscription Service: [Enabling the Subscription Service](#) on page 374
 - Transformation Services: [Changing the Office subsystems](#) on page 381
-  Use Repository Services to manage individual repository servers. This function not be accessed through a load balancer.
- Support Tools:
 - Download JMX Dump: [Downloading the JMX Dump](#) on page 609
 - Node Browser: [Using the Node Browser](#) on page 616
- Directory Management: [Managing authentication directories](#) on page 228

- Virtual File Systems:
 - File Servers: [Enabling file servers](#) on page 358
 - IMAP Service: [Enabling the IMAP Service using the Admin Console](#) on page 376

The links provide more information on configuring these activities.

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 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.

Viewing the System Summary

System Summary in the Admin Console 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 [Changing the Office subsystems](#) on page 381 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.

Module Packages

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

Users and Groups

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

Viewing 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.

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:

Enterprise

More Values

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

RemainingDays: -1

Save

Cancel

Using the Share Admin Tools

Share Admin Tools help you to manage your administration operations.

Alfresco administrators can use the Share Admin Tools to create and manage users and groups directly in 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 moved to the Repository Administration Console (Admin Console), which is available in Alfresco Enterprise only.

Admin Tools is visible on the toolbar if you are an Administrator or a user who is a member of the ALFRESCO_ADMINISTRATORS or ALFRESCO_MODEL_ADMINISTRATORS groups. If you are a member of SITES_ADMINISTRATORS, you'll have an additional **Sites Manager** option on the Alfresco toolbar instead of **Admin Tools**.

About the Share Admin Tools

You can 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. Use the links to see more information about each tool.

1. Click **Admin Tools**.

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

- **Application:** [Managing Share features](#) on page 452
- **Category Manager:** [Managing categories](#) on page 454
- **Module Browser:** [Viewing module packages](#) on page 50
- **Node Browser:** [Using the Node Browser in Share Admin Tools](#)
- **Tag Manager:** [Managing tags](#) on page 454
- **Model Manager:** [Content modeling with Model Manager](#) on page 406
- **Sites Manager:** [Managing sites](#) on page 455

The remaining tools are grouped into the following categories:

- **Repository**
 - **Replication Jobs:** [Managing replication jobs](#) on page 595
- **Users and Groups**
 - **Groups:** [Managing groups](#) on page 460
 - **Users:** [Managing users](#) on page 456

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

Using 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.

The `alfresco-global.properties` file is created when you install Alfresco with the setup wizards, and it logs many of the settings that you specify in the process. You can then use this file to add further property settings. If you are installing Alfresco manually, then you can use the

`alfresco-global.properties.sample` file. The `.sample` file contains some of the common properties required for setting up Alfresco.

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. See [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 LibreOffice installation.
<code>ooo.enabled=</code>	Specifies whether to use the Direct LibreOffice subsystem.
<code>jodconverter.officeHome=</code>	Specifies the location of the LibreOffice 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.

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.

 Ensure that you use single-byte character sets (ISO-8859-1 Latin 1) in your `alfresco-global.properties` settings, particularly the `system.webdav.rootPath` setting. If you require other characters, you can use Unicode equivalents. For

example, if your root path in Cyrillic was #####, which means folder in English, a valid value would be:

```
system.webdav.rootPath=/app:company_home/cm:\u0444\u043E\u043B\u0434\u0434\u0435\u0440
```

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

The `imap.server.mountPoints` property is used as an example for setting composite properties.

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
```

Using a JMX client to change settings dynamically

By default, you can reconfigure Alfresco by shutting down the server, editing the relevant property in the configuration files, and then restarting the server. If you have installed the Oracle Java SE Development Kit (JDK), there are some support operations that can be performed 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)
- 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

 Restrict JMX RMI connections to an internal administration group, due to security vulnerabilities. JMX/RMI deserializes data from a client before authentication, which means that password protection does not provide adequate security.

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.

There are two types of property that you can edit using a JMX interface:

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 the Alfresco 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.

Connecting to Alfresco through JMX client

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

-  Restrict JMX RMI connections to an internal administration group, due to security vulnerabilities. JMX/RMI deserializes data from a client before authentication, which means that password protection does not provide adequate security.

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
```

Enabling JMX

Remote JMX functionality is disabled by default in Alfresco. You can enable JMX by setting the `alfresco.jmx.connector.enabled` property in the `alfresco-global.properties` file.



Restrict JMX RMI connections to an internal administration group, due to security vulnerabilities. JMX/RMI deserializes data from a client before authentication, which means that password protection does not provide adequate security.

To enable the JMX server:

1. Open the `alfresco-global.properties` file.
2. Set the following property: `alfresco.jmx.connector.enabled=true`
3. Save the file.

Configuring Alfresco with JConsole

If you have installed the Oracle Java SE Development Kit (JDK), you can use the JMX client, JConsole, for Alfresco runtime administration.

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

Using the Java command line to change settings dynamically

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.

You can use the `-D` options for 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=xxxx
```

Customizing Alfresco applications

You can make basic configuration updates to customize Alfresco, or modify properties files 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.

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 the Activity Email Summary

The Spring bean definition for the ActivitiesFeed subsystem is called `activities-feed-context.xml` and can be downloaded from the Alfresco SVN: [activities-feed-context.xml](#).

1. Download the file and save to the `<subsystems/ActivitiesFeed/default>` directory.

The file contains the following bean override for the `file-previewed` and `file-downloaded` values:

```
<?xml version='1.0' encoding='UTF-8'?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">
    <bean id="feedModelBuilderPrototype"
          class="org.alfresco.repo.activities.feed.DefaultActivitiesFeedModelBuilder"
          scope="prototype">
        <property name="ignoredActivityTypes">
            <set>
                <value>org.alfresco.documentlibrary.file-previewed</value>
                <value>org.alfresco.documentlibrary.file-downloaded</value>
            </set>
        </property>
    </bean>
</beans>
```

2. Remove or comment out the following lines to include the `file-previewed` and `file-downloaded` entries in your Activity Email Summary:

```
<property name="ignoredActivityTypes">
    <set>
        <value>org.alfresco.documentlibrary.file-previewed</value>
        <value>org.alfresco.documentlibrary.file-downloaded</value>
    </set>
</property>
```

3. Save your file.

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.

System path conventions



- 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.

<configRoot>

The <configRoot> directory contains the default application configuration. For example, for Tomcat, <configRoot> is <TOMCAT_HOME>\webapps\alfresco\WEB-INF.

<configRootShare>

The <configRootShare> directory contains the default application configuration for Share. For example, for Tomcat, <configRootShare> is <TOMCAT_HOME>\webapps\share\WEB-INF.

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 considered as an 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. This means that it can reference all the beans in the main application context. However, the subsystem 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 used with different parameters in different instances. For example, this allows you to chain, or combine functions of more than one subsystem, through property file edits.

Dynamic existence

The subsystem has JMX-based server configuration capabilities.

Own bean namespace

You do not need unique bean names if you use multiple instances of the same subsystem.

This simplifies the problem of building an authentication chain as there is no need to edit a template Spring configuration.

Clearly defined interfaces with the rest of the system

The subsystem interfaces must be imported 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 beans are hidden in a private container.

Swapping of alternative implementations

To switch from native Alfresco authentication to [NTLM pass-through authentication](#), you switch to an Alfresco `passthru` authentication subsystem and the correct components are swapped in.

Separate product from configuration

A subsystem binds its configuration settings to properties. There is no need to edit or extend a 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:

Subsystem name	Functional area	More information
ActivitiesFeed	Activities notifications	Configuring the Activities Feed on page 374
Audit	Audit related functions	Auditing Alfresco
Authentication	Authentication related functions	Setting up authentication and security on page 165
ContentStore	Properties for the encrypted and non-encrypted Content Stores	Setting up content stores on page 571
email	Outbound and inbound SMTP property settings	Configuring inbound and outbound email on page 367
fileServers	Properties for the CIFS and FTP servers.	Configuring file servers on page 358
googledocs	Properties for Google Docs integration	Installing and configuring Google Docs integration on page 107
imap	Properties for the IMAP service	Configuring the email client with IMAP on page 375

Subsystem name	Functional area	More information
OOoDirect	OpenOffice transformations settings (use LibreOffice where possible)	Configuring OpenOffice transformations in place of LibreOffice on page 383
OOoJodconverter	Default settings for LibreOffice transformations	Configuring LibreOffice on page 381
Replication	Settings for the replication jobs tool	Setting up and managing content replication on page 592
Search	Search mechanism for Alfresco	Configuring search on page 264
Subscriptions	Settings for the activities feeds	Enabling the Subscription Service on page 374
Synchronization	Synchronization of local user and group information with the user registry exporters (usually LDAP directories) in the authentication chain	Configuring synchronization on page 225
sysAdmin	Properties for server administration	Configuring server administration settings on page 342
thirdparty	Properties for third-party software that is used by Alfresco, for example, ImageMagick	Changing the Office subsystems on page 381
Transformers	Properties for the transformation server	Managing transformations on page 507
wcm_deployment_receiver	Properties for WCM Deployment Receiver	The Workflow Console on page 488

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 `mysubsystem.properties` file, containing the following:

```
ooo.user=${dir.root}/ouser
```

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
```

Configuring databases

Configuring Amazon RDS databases

Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale a relational database in the cloud.

It is a web service running in the cloud and provides relational database for use in Alfresco. Amazon RDS supports and gives you online access to the capabilities of a MySQL, Oracle, Microsoft SQL Server, PostgreSQL, or Amazon Aurora relational database management system.

As a good practice, when using Amazon EC2 environment you may want to use S3 bucket for content store. For more information, see [Alfresco S3 Connector](#).

For configuring different databases for Amazon RDS, see the topics below.

[Configuring an Aurora database on Amazon RDS](#)

Use this information to configure an Aurora database on Amazon RDS for use with Alfresco. Amazon Aurora is a MySQL-compatible relational database management system.

Prerequisites:

- Aurora support is only available when running in AWS.
 - Setup Amazon RDS using the AWS Management Console. For more information, see the [AWS documentation](#).
 - Alfresco is deployed on an Amazon EC2 instance
1. Use the `ssh` command to connect to the Amazon EC2 instance using a provided `.ppk` key. For Amazon Linux, the user name is `ec2-user`. For RHEL5, the user name is either `root` or `ec2-user`. For Ubuntu, the user name is `ubuntu`. For SUSE Linux, the user name is `root`.

For Amazon Linux, the user name is `ec2-user`. For RHEL5, the user name is either `root` or `ec2-user`. For Ubuntu, the user name is `ubuntu`. For SUSE Linux, the user name is `root`.
 2. Execute `sudo su` to change to root.
 3. Download the Alfresco installer for Linux from the [Alfresco Support Portal](#).

4. Install the downloaded Alfresco installer using the following commands:

```
chmod 777 alfresco-enterprise-5.1.x-installer-linux-x64.bin
sudo ./alfresco-enterprise-5.1.x-installer-linux-x64.bin
```

5. Install the Aurora database connector.

This release requires the MySQL JDBC driver 4.0 for compatibility with the MySQL database.

a. Download the mysql-connector driver from the MySQL JDBC driver download site.

b. Copy the JDBC driver into the <TOMCAT_HOME>/lib directory.

6. Install and use a database tool to connect to the Amazon RDS.

7. Create a database named alfresco.

8. Create a user named alfresco.

9. Set the new user's password to alfresco.

10. Open the <classpathRoot>/alfresco-global.properties file.

11. Locate the following property:

```
dir.root=
```

12. 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

13. Set and uncomment the database connection properties as shown below:

```
db.name=alfresco2
db.username=alfresco
db.password=alfresco
db.host=auroraqaadb-cluster.cluster-clqevmd2v8y9.us-
east-1.rds.amazonaws.com
db.port=13306
db.prefix=mysql
db.pool.max=275

# MySQL database connection

db.driver=org.gjt.mm.mysql.Driver
db.url=jdbc:mysql://${db.host}/${db.name}?${db.params}
OR
db.url=jdbc:mysql://${db.host}:${db.port}/${db.name}?${db.params}
```

14. Save the file.

15. Restart the Alfresco server.

Configuring the MySQL database on Amazon RDS

Prerequisites:

- Setup Amazon RDS using the AWS Management Console. For more information, see the [AWS documentation](#).
- Amazon EC2 instance

1. Use the ssh command to connect to the Amazon EC2 instance using a provided .ppk key.

For Amazon Linux, the user name is ec2-user. For RHEL5, the user name is either root or ec2-user. For Ubuntu, the user name is ubuntu. For SUSE Linux, the user name is root.

2. Execute sudo su to change to root.

3. Download the Alfresco installer for Linux from the [Alfresco Support Portal](#).

4. Install the downloaded Alfresco installer using the following commands:

```
chmod 777 alfresco-enterprise-5.1.x-installer-linux-x64.bin
```

```
sudo ./alfresco-enterprise-5.1.x-installer-linux-x64.bin
```

5. 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.

- Download `mysql-connector-java-5.1.32` from the MySQL download site: <http://dev.mysql.com/>.

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

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

6. Install and use a database tool to connect to the Amazon RDS.

7. Create a database named `alfresco`.

8. Create a user named `alfresco`.

9. Set the new user's password to `alfresco`.

10. Open the `<classpathRoot>/alfresco-global.properties` file.

11. 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`

12. Set and uncomment the database connection properties as shown below:

```
db.name=alfresco
db.username=alfresco
db.password=alfresco
db.host=alfqa-mysql5-6-19a.cw4mo3qj8qdu.us-east-1.rds.amazonaws.com
db.port=1433
db.pool.max=275
db.txn.isolation=4096

# MySQL connection

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

 Ensure that these database connection properties are not commented out.

13. Save the file.

14. Restart the Alfresco server.

Configuring an Oracle database on Amazon RDS

Use this information to configure an Oracle database on Amazon RDS for use with Alfresco.

Prerequisites:

- Setup Amazon RDS using the AWS Management Console. For more information, see the [AWS documentation](#).
- Amazon EC2 instance

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.

1. Use the `ssh` command to connect to the Amazon EC2 instance using a provided `.ppk` key.

For Amazon Linux, the user name is `ec2-user`. For RHEL5, the user name is either `root` or `ec2-user`. For Ubuntu, the user name is `ubuntu`. For SUSE Linux, the user name is `root`.

2. Execute `sudo su` to change to root.

3. Download the Alfresco installer for Linux from the [Alfresco Support Portal](#).

4. Install the downloaded Alfresco installer using the following commands:

```
chmod 777 alfresco-enterprise-5.1.x-installer-linux-x64.bin
sudo ./alfresco-enterprise-5.1.x-installer-linux-x64.bin
```

5. Install the Oracle database connector. The database connector allows Oracle database to talk to the Alfresco server.
 - a. 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.
 - b. Copy the JAR file into the `/lib` directory.
For example, for Tomcat, copy the JAR file into the `<TOMCAT_HOME>/lib` directory.
6. Install and use a database tool to connect to the Amazon RDS.
7. Increase the available connections.

- a. In the SQL*Plus Console, run these commands:

```
alter system set processes=275 scope=spfile sid='*';
alter system set sessions=305 scope=spfile sid='*';
alter system set transactions=330 scope=spfile sid='*';
```

- b. Restart the database.
8. Create a database named `alfresco`.
9. Create a user named `alfresco`.
10. Set the new user's password to `alfresco`.
11. Open the `<classpathRoot>/alfresco-global.properties.sample` file.
12. 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`

13. Set and uncomment the database connection properties as shown below:

```
db.name=alfresco
db.username=alfresco
db.password=alfresco
db.host=alfrescooral12.cw4mo3qj8qdu.us-east-1.rds.amazonaws.com
db.port=1433
db.pool.max=275
db.txn.isolation=4096

# Oracle database connection

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

 Ensure that these database connection properties are not commented out.

14. Save the file without the `.sample` extension.
15. Restart the Alfresco server.

Configuring a PostgreSQL database on Amazon RDS

Use this information to configure a PostgreSQL database on Amazon RDS for use with Alfresco.

Prerequisites:

- Setup Amazon RDS using the AWS Management Console. For more information, see the [AWS documentation](#).
 - Amazon EC2 instance
1. Use the `ssh` command to connect to the Amazon EC2 instance using a provided `.ppk` key.

For Amazon Linux, the user name is `ec2-user`. For RHEL5, the user name is either `root` or `ec2-user`. For Ubuntu, the user name is `ubuntu`. For SUSE Linux, the user name is `root`.

2. Execute `sudo su` to change to root.
3. Download the Alfresco installer for Linux from the [Alfresco Support Portal](#).
4. Install the downloaded Alfresco installer using the following commands:

```
chmod 777 alfresco-enterprise-5.1.x-installer-linux-x64.bin
sudo ./alfresco-enterprise-5.1.x-installer-linux-x64.bin
```

5. 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.

6. Install and use a database tool to connect to the Amazon RDS Postgresl datasource. If Alfresco is installed as plain vanilla, `psql` from the Alfresco installation folder can be used.

7. Create a database named `alfresco`.

8. Create a user named `alfresco`.

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

9. Set the new user's password to `alfresco`.

10. Open the `<classpathRoot>/alfresco-global.properties` file.

11. Locate the following line:

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

12. 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`

13. Uncomment and set the database connection properties.

```
db.name=alfresco
db.username=alfresco
db.password=alfresco
db.host=postgresql-alfresco.cw4mo3qj8qdu.us-east-1.rds.amazonaws.com
db.port=5432
db.pool.max=275

# 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}
```

 Ensure that these database connection properties are not commented out.

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

Configuring a SQL Server database on Amazon RDS

Use this information to configure a SQL Server database on Amazon RDS for use with Alfresco.

Prerequisites:

- Setup Amazon RDS using the AWS Management Console. For more information, see the [AWS documentation](#).

- Amazon EC2 instance
- Use the `ssh` command to connect to the Amazon EC2 instance using a provided `.ppk` key. For Amazon Linux, the user name is `ec2-user`. For RHEL5, the user name is either `root` or `ec2-user`. For Ubuntu, the user name is `ubuntu`. For SUSE Linux, the user name is `root`.
 - Execute `sudo su` to change to root.
 - Download the Alfresco installer for Linux from the [Alfresco Support Portal](#).
 - Install the downloaded Alfresco installer using the following commands:
- ```
chmod 777 alfresco-enterprise-5.1.x-installer-linux-x64.bin
sudo ./alfresco-enterprise-5.1.x-installer-linux-x64.bin
```
- 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.
  - Install and use a database tool to connect to the Amazon RDS.
  - Create a database named `alfresco`.
  - Enable snapshot isolation mode with the following command:
- ```
ALTER DATABASE alfresco SET ALLOW_SNAPSHOT_ISOLATION ON;
```
- Create a user named `alfresco`.
 - Set the new user's password to `alfresco`.
 - Open the `<classpathRoot>/alfresco-global.properties` file.
 - Locate the following property:
- ```
dir.root=
```
- 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`
  - Set and uncomment the database connection properties as shown below:
- ```
db.name=alfresco
db.username=alfresco
db.password=alfresco
db.host=sql-alfresco.cw4mo3qj8qdu.us-east-1.rds.amazonaws.com
db.port=1433
db.pool.max=275
db.txn.isolation=4096

# SQL Server connection

db.driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
db.url=jdbc:sqlserver://{$db.host}:{$db.port};databaseName={$db.name}
```
- Save the file.
 - Restart the Alfresco server.

Configuring a DB2 database

Use this information 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.

- a. Obtain a copy of db2jcc4.jar. This should be available in the /java or /jdbc directory of your DB2 installation.
- b. Copy the JAR file into the <TOMCAT_HOME>/lib directory for Tomcat.
2. Increase the available connections in DB2.
 - a. Follow these instructions to increase the max_connections setting to 275: [Setting max_connections](#)
 - b. Run the following command:

```
update dbm cfg using max_connections 275 automatic
```

3. 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.

4. 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.
5. 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.
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
9. When using a schema which does not match the DB2 username, set the currentSchema and hibernate.default_schema properties as shown below:

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

 Remember to uncomment the database connection properties.

10. Set the other database connection properties.

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

 Remember to uncomment the database connection properties.

11. Save the file without the .sample extension.
12. 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.

Configuring the MariaDB database connection

To configure a MariaDB database connection for use with Alfresco, use the MySQL JDBC driver and follow instructions in [Configuring the MySQL database](#).

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. Increase the maximum connections setting in the MySQL configuration file.

- a. Locate the configuration file:

- Linux: `/etc/my.cnf`
- Windows: `c:\Users\All Users\MySQL\MySQL Server 5.6\my.ini`

- b. In the `mysqld` section, add or edit the `max_connections` property:

```
max_connections = 275
```

- c. Restart the database.

4. Create a user named `alfresco`.

5. Set the new user's password to `alfresco`.

6. 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.

7. Open the `<classpathRoot>/alfresco-global.properties.sample` file.

8. 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`

9. 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
```

- 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 7.
- (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 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

There are some settings that are required for MySQL to work with Alfresco.

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.
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.

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 an Oracle database

Use this information to configure an Oracle RDBMS database for use with Oracle.

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.
 -  The Oracle Thin driver is recommended. Check the [supported platform](#) page for the proper driver and connection URL to use.
1. Create a database named `alfresco`.
 2. 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.
 3. Set the new user's password to `alfresco`.
 4. 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>
```

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. Set and uncomment the Oracle database connection properties as shown below:

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

# Oracle connection
```

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

If using the `oci` configuration, change the URL syntax as shown below:

```
db.url=jdbc:oracle:oci:@${db.host}:${db.port}:${db.name}
```

The Oracle connection URL in this example is basic. Typical Oracle connection strings can be used with the Alfresco supported Oracle driver (`thin/oci`). The Thin driver is recommended over the OCI driver.

 If you are using the OCI URL, you need an Oracle client on the Alfresco host. For more information, see [Oracle Instant Client](#).

For database URLs and specifiers, see Oracle documentation at [Database URLs and Database Specifiers](#) and [Thin-style Service Name Syntax](#).

You can use the standard (OCI/Thin) connection URL, Oracle service, and Oracle DNS service URLs.

9. Save the file without the `.sample` extension.
10. Copy the Oracle JDBC driver JAR into `/lib`.

 Do not put multiple driver jars in the application or the application server `lib` directory. Only include the driver jar which is advised in these instructions. Remove any others, if present.

11. Restart the Alfresco server.

 If you receive JDBC errors:

- Ensure the location of the Oracle JDBC drivers are on the system path or added to the relevant `lib` directory of the application server.
- Check if you have `LD_LIBRARY_PATH` in use in your environment to remove the old Oracle client (for example, `/home/oracle/app/oracle/product/11.2.0/client_1/lib`) and add the full path to the current `ojdbc7.jar`. If you do not have this environment variable, do not add it.

 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 PostgreSQL database

Use this information 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. Increase the maximum connections setting in the PostgreSQL configuration file.
 - a. Locate the configuration file:
 - Linux: `/var/lib/pgsql/9.3/data/postgresql.conf`
 - Windows: `C:\Program Files\PostgreSQL\9.3\data\postgresql.conf`
 - b. Add or edit the `max_connections` property:


```
max_connections = 275
```

- c. Restart the database.
- 3. Create a database named alfresco.
- 4. Create a user named alfresco.
This user must have write permissions on all tables and sequences.
- 5. Set the new user's password to alfresco.
- 6. Ensure the alfresco user has the required privileges to create and modify tables.
- 7. Open the <classpathRoot>/alfresco-global.properties.sample file.
- 8. Locate the following line:

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

- 9. 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
- 10. 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}
```

- 11. 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.

- 12. Save the file without the .sample extension.
- 13. 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
```

- 14. 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 a SQL Server database

Use this information 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.

1. 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.
 - a. Download sqljdbc4.jar from the Microsoft SQL Server download site.
 - b. Copy the JDBC driver into the <TOMCAT_HOME>/lib directory.
2. Increase the available connections setting in the Microsoft SQL Server configuration file. Follow these instructions to update the setting: [Configuring the user connections option](#).
3. Create a database named alfresco.

- Create the database using default collation settings.
4. Create a user named `alfresco`.
This user must have write permissions on all tables and sequences. For example, you can provide these permissions by granting your database user (in this case, the `alfresco` user) the `db_owner` role. See [Database-Level Roles](#) for more information.
 5. Set the new user's password to `alfresco`.
 6. 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.
 7. Enable snapshot isolation mode with the following command:

```
ALTER DATABASE alfresco SET ALLOW_SNAPSHOT_ISOLATION ON;
```
 8. Ensure that the TCP connectivity is enabled on the fixed port number 1433.
 9. Open the `<classpathRoot>/alfresco-global.properties.sample` file.
 10. Locate the following property:

```
dir.root=
```
 11. 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`
 12. Set the database connection properties.

```
db.name=alfresco
db.username=alfresco
db.password=alfresco
db.host=localhost
db.port=1433
db.pool.max=275
```
 13. 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
```
 14. Save the file without the `.sample` extension.
 15. 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.

Optimizing Microsoft SQL Server to work with Alfresco

Make sure you manage Microsoft SQL Server to optimise performance.

To ensure that your performance does not degrade, it is useful to carry out the following weekly maintenance operations on your SQL server, especially in repositories with a high transaction count and frequency:

- Recompute statistics by running the command: `EXEC sp_updatestats`
- Clear the buffers by running the command: `DBCC DROPCLEANBUFFERS`
- Clear the cache by running the command: `DBCC FREEPROCCACHE`
- Run an index fragmentation check and:
 - Rebuild anything that is >30% fragmented
 - Reorganize anything that is between 5 and 30% fragmented

See [Reorganize and Rebuild Indexes](#) for more information.

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 5.1 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
db.txn.isolation	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
db.pool.initial	The number of connections opened when the pool is initialized.	10
db.pool.validate.query	The SQL query that is 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> For PostgreSQL database, use <code>SELECT 1</code>

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

Property name	Description	Default value
db.pool.statements.enable	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>
db.pool.statements.max	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
db.pool.idle	The maximum number of connections that are not in use kept open.	10
db.pool.max	The maximum number of connections in the pool. See the Note below for more information on this property.	275
db.pool.min	The minimum number of connections in the pool.	10

Property name	Description	Default value
db.pool.wait.max	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
db.pool.validate.borrow	A Boolean property. When set to true it indicates that connections will be validated before being borrowed from the pool.	true
db.pool.validate.return	A Boolean property. When set to true it indicates that connections will be validated before being returned to the pool.	false
db.pool.evict.interval	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
db.pool.evict.idle.min	The minimum number of milliseconds that a connection may remain idle before it is eligible for eviction.	1800000
db.pool.evict.validate	A Boolean property. When set to true it indicates that the idle connections will be validated during eviction runs.	false
db.pool.abandoned.detect	A Boolean property. When set to true it indicates that a connection is considered abandoned and eligible for removal if it has been idle longer than the db.pool.abandoned.time.	false
db.pool.abandoned.time	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.

Validating your database

Validate your database to ensure that it meets the prerequisites for an Alfresco installation.

-  Alfresco is unable to provide specialized support for maintaining or tuning your relational database. You MUST have an experienced, certified DBA on staff to support your Alfresco installation(s). Typically this is not a full time role once the database is configured and tuned and automated maintenance processes are in place. However an experienced, certified DBA is required to get to this point.

Maintenance and Tuning:

As with any application that uses a relational database, regular maintenance and tuning of the Alfresco database and schema is necessary. Specifically, all of the database servers that Alfresco supports require a minimum level of index statistics maintenance at frequent, regular intervals. Unless your DBA suggests otherwise, Alfresco recommends daily maintenance.

-  Relying on your database's automated statistics gathering mechanism might not be optimal – consult an experienced, certified DBA for your database to confirm this.
-  Index maintenance on most databases is an expensive, and in some cases blocking, operation that can severely impact Alfresco performance while in progress. Consult your experienced, certified DBA regarding best practices for scheduling these operations in your database.

The following table describes example commands for specific databases. These commands are for illustration only. You must validate the commands required for your environment with your DBA.

Database	Example maintenance commands
MySQL	ANALYZE - consult with an experienced, certified MySQL DBA who has InnoDB experience (Alfresco cannot use a MyISAM database and hence an InnoDB-experienced MySQL DBA is required). Refer to the following link: http://dev.mysql.com/doc/refman/5.6/en/analyze-table.html .
PostgreSQL	VACUUM and ANALYZE – consult with an experienced, certified PostgreSQL DBA. Refer to the following link: http://www.postgresql.org/docs/8.4/static/maintenance.html .
Oracle	Depends on version – consult with an experienced, certified Oracle DBA. Refer to the following link: http://download.oracle.com/docs/cd/B19306_01/server.102/b14211/stats.htm#PFGRF003 .
Microsoft SQL Server	ALTER INDEX REBUILD (http://technet.microsoft.com/en-us/library/ms188388.aspx), UPDATE STATISTICS (http://technet.microsoft.com/en-us/library/ms187348.aspx) – consult with an experienced, certified MS SQL Server DBA

Database	Example maintenance commands
DB2	<p>REORGCHK ()</p> <p>http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.cmd.doc/doc/r0001971.html</p> <p>RUNSTATS ()</p> <p>http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.cmd.doc/doc/r0001980.html</p>

Setting up authentication and security

Use this information to manage authentication in Alfresco, and to configure keystores for repository encryption.

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 in Alfresco.

-  Choose a strong, unique password for your admin account, and consider changing it regularly.

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
- Protect your repository and Solr communications using encryption

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 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.

Managing Alfresco keystores

The out-of-the-box Alfresco installation has a pre-configured main keystore, which contains a secret key generated by Alfresco. If you want to use encrypted properties, you should create your own keystore with your own password, and update the metadata file appropriately.

The default keystore configuration protects the keys by using two levels of passwords - a keystore password and a password for each key. Currently, the keystore contains only a metadata secret key that is used for encrypting and decrypting node properties that are of type d:encrypted.

You can also configure a backup keystore. This is useful in case the keys need to be changed. The user can back up the main keystore to the backup keystore location and create a new keystore in its place.

If both the main and backup keystores are configured, the repository encryption works in the *fallback* mode. In this mode, the node properties are decrypted with the main keystore's metadata key first. If that fails, the backup keystore's metadata key is tried. This allows the keystores to be changed on the disk and reloaded without affecting the running of the repository.

Keystores are used also to protect repository/Solr communications using encryption and mutual authentication. In this case, the keystores store RSA keys and certificates. For more information, see [Solr 4 security](#).

Keystore configuration

You can configure the main and backup keystores using the `alfresco-global.properties` file.

To configure the main keystore, set the following properties in the `alfresco-global.properties` file:

Property	Description with example
<code>encryption.keystore.location</code>	Specifies the location of the main keystore. <code>encryption.keystore.location= \${dir.keystore}/keystore</code>
<code>encryption.keystore.keyMetaData.location</code>	Specifies the location of the main keystore's metadata file. <code>encryption.keystore.keyMetaData.location= \${dir.keystore}</code>
<code>encryption.keystore.provider</code>	Specifies the main keystore provider.
<code>encryption.keystore.type</code>	Specifies the main keystore type. <code>encryption.keystore.type=JCEKS</code>
<code>encryption.keystore.backup.location</code>	Specifies the location of the backup keystore. <code>encryption.keystore.backup.location= \${dir.keystore}/backup-keystore</code>
<code>encryption.keystore.backup.keyMetaData.location</code>	Specifies the location of the backup keystore's metadata file. <code>encryption.keystore.backup.keyMetaData.location= \${dir.keystore}</code>
<code>encryption.keystore.backup.provider</code>	Specifies the backup keystore provider.
<code>encryption.keystore.backup.type</code>	Specifies the backup keystore type. <code>encryption.keystore.backup.type=JCEKS</code>

Some other general encryption properties are:

```
encryption.keySpec.class=org.alfresco.encryption.DESedeKeyGenerator
encryption.keyAlgorithm=DESede
encryption.cipherAlgorithm=DESede/CBC/PKCS5Padding
```

Because of these encryption properties, the keystores and metadata files can be easily located. Also, the metadata file uses a clear text password to access the keystore. For this reason, appropriate operating system permissions should be applied so that the files cannot be accidentally changed nor read by anyone other than an administrator and the username running the repository.

Each keystore must have a corresponding keystore metadata file. This file contains the passwords, its keys, and other metadata relevant to the keystore. The metadata file must contain three entries:

- `aliases=<active key aliases in the key store>`
- `keystore.password=<key store password>`
- `metadata.password=<metadata key password>`

At bootstrap, the repository checks if the metadata key in the main keystore has been changed (unless running in the fallback mode, in which case the backup keystore is checked instead).

This prevents accidental changes to the keystore. If it detects that the metadata key has been changed, an exception will occur and the bootstrap will stop.

Keystore generation

Keystore generation can be automatic or manual.

Automatic keystore generation

During bootstrap, if the repository detects a missing secret key keystore, it will dynamically create a keystore containing a single metadata secret key. In order to do this, the repository assumes the existence of a keystore metadata file containing information about the metadata key. Specifically, it expects the following properties to be set:

Property	Description
metadata.keyData	Specifies the key data used to generate the secret key.
metadata.algorithm	Specifies the key algorithm used to generate the secret key.

The `keyData` can be generated by executing the class `org.alfresco.encryption.GenerateSecretKey` as shown below:

```
java -classpath "projects/3rd-party/lib/commons/commons-codec-1.4.jar:projects/core/build/dist/alfresco-core-4.0.a.jar"
  org.alfresco.encryption.GenerateSecretKey
```

Manual keystore generation

A new keystore can be generated using the Java `keytool` command as shown below:

```
keytool -gensecretkey -alias metadata -keypass <metadata key password> -storepass
<key store password> -keystore keystore
-storetype JCEKS -keyalg DESede
```

- ✎ Make sure the keystore is placed in the location specified by the property `encryption.keystore.location` and that the passwords you have used in the `keytool` commands are placed in the file specified by the property `encryption.keystore.keyMetaData.location`.

Keystore key registration

The keystore keys are registered with the repository to ensure that they are not accidentally changed.

During bootstrap and JMX keystore reload and re-encryption operations, the repository checks if the main keystore's keys and the metadata key have changed. If they have changed, the repository throws an exception.

Encrypted node properties

Data encryption in Alfresco uses secret keys which are stored in the Alfresco Java keystore.

Alfresco provides a type `d:encrypted` that can be used to store sensitive property values in the database in a sealed encrypted format.

Using encrypted node properties

Node properties can be encrypted in the repository by setting their type to `d:encrypted` in the model.

By default, the node service will not automatically encrypt and decrypt these properties as they pass in and out of the node service. Encryption and decryption should be handled by the trusted custom code (that uses the `metadataEncryptor`) running in the repository. Clients, such as Alfresco Share will not automatically be able to decrypt and display encrypted property values.



The encrypted node properties will not available in Alfresco Share without the code.

The `org.alfresco.repo.node.encryption.MetadataEncryptor` class (defined as the Alfresco Spring bean with name `metadataEncryptor`) provides an interface to encrypt and decrypt encryptable properties. The repository's node integrity checking will ensure that encryptable properties are actually encrypted (by the `MetaDataEncryptor`) when the transaction commits. If they are not encrypted, an integrity violation exception is raised.

For example, given the model:

```
<model name="test:encryptedPropModel" xmlns="http://www.alfresco.org/model/
dictionary/1.0">
    <description>Alfresco Content Model</description>
    <author>Alfresco</author>
    <published>2005-05-30</published>
    <version>1.0</version>

    <imports>
        <import uri="http://www.alfresco.org/model/dictionary/1.0" prefix="d"/>
        <import uri="http://www.alfresco.org/model/content/1.0" prefix="cm"/>
    </imports>

    <namespaces>
        <namespace uri="http://www.alfresco.org/test/encryptedPropModel/1.0"
prefix="test"/>
    </namespaces>

    <constraints>
    </constraints>

    .....
    <types>
        <type name="test:encrypted">
            <title>Encrypted</title>
            <description>The Base Type</description>
            <parent>cm:content</parent>

            <properties>
                <property name="test:prop1">
                    <type>d:encrypted</type>
                    <protected>true</protected>
                    <default></default>
                    <constraints>
                    </constraints>
                </property>
            </properties>

            <associations>
            </associations>

            <mandatory-aspects>
            </mandatory-aspects>
        </type>
    </types>
</model>
```

the following code creates a node of type `test:encrypted` using `MetadataEncryptor` to encrypt the property.

```
MetadataEncryptor metadataEncryptor =
(MetadataEncryptor)ctx.getBean("metadataEncryptor");

.....
Map<QName, Serializable> allProperties = new PropertyMap();
```

```

allProperties.put(ENCRYPTED_PROP_QNAME, "ABC");
allProperties = metadataEncryptor.encrypt(allProperties);

try
{
    // Create a node using the thread's locale
    NodeRef nodeRef2 = nodeService.createNode(
        nodeRef1,
        ContentModel.ASSOC_CONTAINS,
        QName.createQName(NamespaceService.CONTENT_MODEL_1_0_URI, getName()),
        ENCRYPTED_TYPE_QNAME, allProperties).getChildRef();
}

```

The property can be decrypted as follows:

```

Serializable encryptedPropertyValue = nodeService.getProperty(nodeRef2,
    ENCRYPTED_PROP_QNAME);
Serializable decryptedPropertyValue =
    metadataEncryptor.decrypt(ENCRYPTED_PROP_QNAME, encryptedPropertyValue);
assertEquals("ABC", decryptedPropertyValue);

```

Changing encrypted properties keystore keys and re-encryption

During bootstrap, the repository checks whether the keys in the main encrypted properties keystore have been changed in order to detect any accidental keystore changes.

However if you purposely want to change your keys, you can do so and the repository will re-encrypt any existing encrypted node properties for you. The newly encrypted node properties will be encrypted using the new keys.

Changing your keys involves backing up your keystore to a specific location and creating a new keystore in its place. This can be done in two ways:

- During bootstrap
- During runtime (Enterprise-only)

Bootstrap Re-encryption

Re-encryption occurs during the repository bootstrap. For bootstrap re-encryption, follow the steps below:

1. Stop the Alfresco server.
2. Set the following property in the `alfresco-global.properties` file.
`encryption.bootstrap.reencrypt=true`
3. Backup the current keystore to `backup-keystore` as shown below:
`mv keystore backup-keystore`
`mv keystore-passwords.properties backup-keystore-passwords.properties`
4. Copy your new keystore over the old keystore.
5. Update `keystore-passwords.properties` with the passwords you used to create the keystore. In other words, update the `keystore.password` property with the keystore password and the `metadata.password` property with the metadata key password.
6. Restart the Alfresco server.

Runtime Re-encryption

Re-encryption occurs while the repository is running. For runtime re-encryption, follow the steps below:

1. Backup the current keystore to `backup-keystore`.
`mv keystore backup-keystore`
`mv keystore-passwords.properties backup-keystore-passwords.properties`
2. Copy your new keystore over the old keystore.

3. In your JMX console, execute the operation **Encryption > Operations > Encrypt**.

This will re-read the main and backup keystores and re-encrypt the encrypted properties. The repository can continue to run during this operation; any newly-created encrypted properties will be encrypted with the new key.

 Only a single re-encryption can be done at a particular time. If a re-encrypt is already running then subsequent requests have no effect.

Cryptographic password hashing

Alfresco uses cryptographic password hashing technique to securely store passwords.

All versions of Alfresco prior to Alfresco One 5.1 used the MD4 (Message Digest 4) and SHA256 hash algorithms (mainly to support NTLM and CIFS) to store critical data. But this is no longer considered a secure approach as the hashed password is very easy to decrypt. As an improvement, Alfresco One 5.1 can now use Bcrypt to store passwords but this is configurable. By default, the system uses MD4 to allow users to use MD4 hashed passwords for NTLM and CIFS authentication.

Bcrypt is an adaptive hash function based on the Blowfish symmetric block cipher cryptographic algorithm. It is incredibly slow to hash input compared to other functions, but this results in a much better output hash. Alfresco One 5.1 is configured to use a strength of 10 to provide a good compromise of speed and strength.

With Bcrypt, the hashing algorithm (also called an encoder) can be configured by setting the `system.preferred.password.encoding` property in the `alfresco-global.properties` file. The supported values for this property are:

- `md4`
- `sha256`
- `bcrypt10`

If you provide a different value, the Alfresco repository won't start.

To maintain backwards compatibility with previous Alfresco versions, the default setting for this property is:

```
system.preferred.password.encoding=md4
```

After upgrading to Alfresco One 5.1, when the user logs in or changes the password, the system rehashes the password using the preferred encoding mechanism and stores the mechanism being used. If the preferred encoding is set to `md4`, the system moves the current hashed passwords for that user.

You can run a background job to completely remove all the old hashed passwords for those users that have not logged in yet. If the system is still set to `md4`, all user objects will be upgraded. However, the background job will maintain the current hash.

If the background job is executed after the `system.preferred.password.encoding` property has been changed, it will double-hash all the user objects in the system (unless they have already been upgraded by the user logging in). As a result, the system will temporarily hash (until the user logs in) the current hashed password, store the list of encoders used, and clean out the old hashes.

The background job uses the repository's `BatchProcessor` to execute the job. The execution of the job can be controlled if necessary via the following properties:

- `system.upgradePasswordHash.jobBatchSize`: Specifies the number of user objects to process in each batch.
- `system.upgradePasswordHash.jobQueryRange`: Specifies the `nodeId` range to search for in each iteration.

- `system.upgradePasswordHash.jobThreadCount`: Specifies the number of threads the batch processor uses.

Out of the box, this background job is enabled but set to a future date. To configure it, set the `system.upgradePasswordHash.jobCronExpression` property in the `alfresco-global.properties` file. For example, the following setting runs the job every 10 minutes:

```
system.upgradePasswordHash.jobCronExpression=0 0/10 * * * ?
```

Alternatively, the job can be executed immediately via a JMX console. The job makes use of `JobLockService` so it is safe to run in a clustered environment.

 Once you change the preferred encoding from `md4`, the NTLM SSO authentication will no longer function. Also, the CIFS authentication will only work if the Kerberos authentication is enabled.

If the password upgrade job is enabled, make sure you enable the `log4j.logger.org.alfresco.repo.security.authentication.UpgradePasswordHashWorker` logging in `log4j.properties`.

You can either set it to `trace` or `debug` as shown below:

```
log4j.logger.org.alfresco.repo.security.authentication.UpgradePasswordHashWorker=trace
```

OR

```
log4j.logger.org.alfresco.repo.security.authentication.UpgradePasswordHashWorker=trace
```

`Trace` displays a list of all the processed users. `Debug` is a slightly less verbose output; it displays a list of only those users whose password was changed.

To monitor users that have their passwords upgraded when they log in, add the following in `log4j.properties`:

```
log4j.logger.org.alfresco.repo.security.authentication.HashPasswordTransactionListener=
```

Encrypting 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 5.1 provides support for encrypting the following configuration properties:

- `dir.root`
- `db.driver`
- `db.username`
- `db.password`
- `db.name`
- `db.pool.validate.query`
- `ooo.exe`
- `jodconverter.officeHome`
- `alfresco_user_store.adminpassword`
- `dir.license.external`
- `index.subsystem.name`
- `cryptodoc.jce.keystore.path`

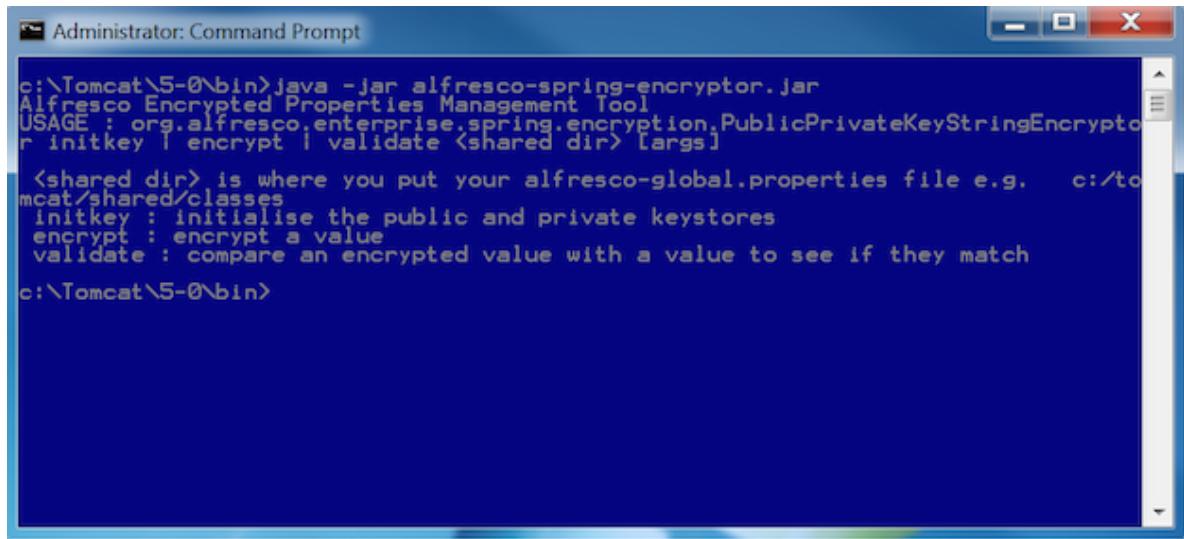
- cryptodoc.jce.keystore.password
- cryptodoc.jce.key.aliases
- cryptodoc.jce.key.passwords

Encrypting configuration properties

You can 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
```

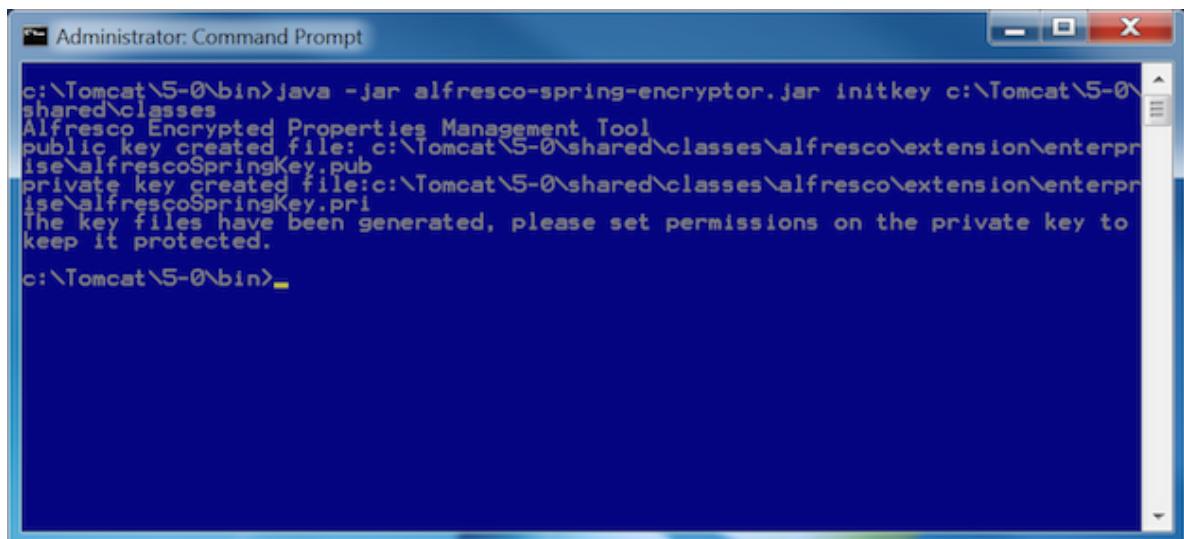


The screenshot shows an Administrator Command Prompt window. The command entered is `java -jar alfresco-spring-encryptor.jar`. The output displays the usage information for the Alfresco Encrypted Properties Management Tool, which includes options for `initkey`, `encrypt`, and `validate`, along with a note about the shared directory path.

```
c:\Tomcat\5-0\bin>java -jar alfresco-spring-encryptor.jar
Alfresco Encrypted Properties Management Tool
USAGE : org.alfresco.enterprise.spring.encryption.PublicPrivateKeyStringEncryptor initkey | encrypt | validate <shared dir> [args]
<shared dir> is where you put your alfresco-global.properties file e.g. c:/tomcat/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
```



The screenshot shows an Administrator Command Prompt window. The command entered is `java -jar alfresco-spring-encryptor.jar initkey c:/Tomcat/5-0/shared/classes`. The output shows the creation of a public key at `c:/Tomcat/5-0/shared/classes/alfresco/extension/enterprise/alfrescoSpringKey.pub` and a private key at `c:/Tomcat/5-0/shared/classes/alfresco/extension/enterprise/alfrescoSpringKey.pri`. A note at the end of the output advises setting permissions on the private key.

```
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\enterprise\alfrescoSpringKey.pub
private key created file:c:/Tomcat/5-0\shared\classes\alfresco\extension\enterprise\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.

```
c:\Tomcat\5-0\bin>java -jar alfresco-spring-encryptor.jar encrypt c:\Tomcat\5-0\shared\classes womble
U37nu8D0xIBZMLU+7eXdFjmUWymPBdZBryojFTLkc4e0dsrVlb0139eIZiQbr9Yx/rWcXYU/Hyv/ZhuwH85qFw==

c:\Tomcat\5-0\bin>java -jar alfresco-spring-encryptor.jar encrypt c:\Tomcat\5-0\shared\classes password
bBZ6V85jKdTMEsMqWHvLKdwLbmTark6xT3YimTfFbu/BT0dwfuao/kKC6MJwC7kd1gCvxaC1pEQMwD3B61LfXQ==

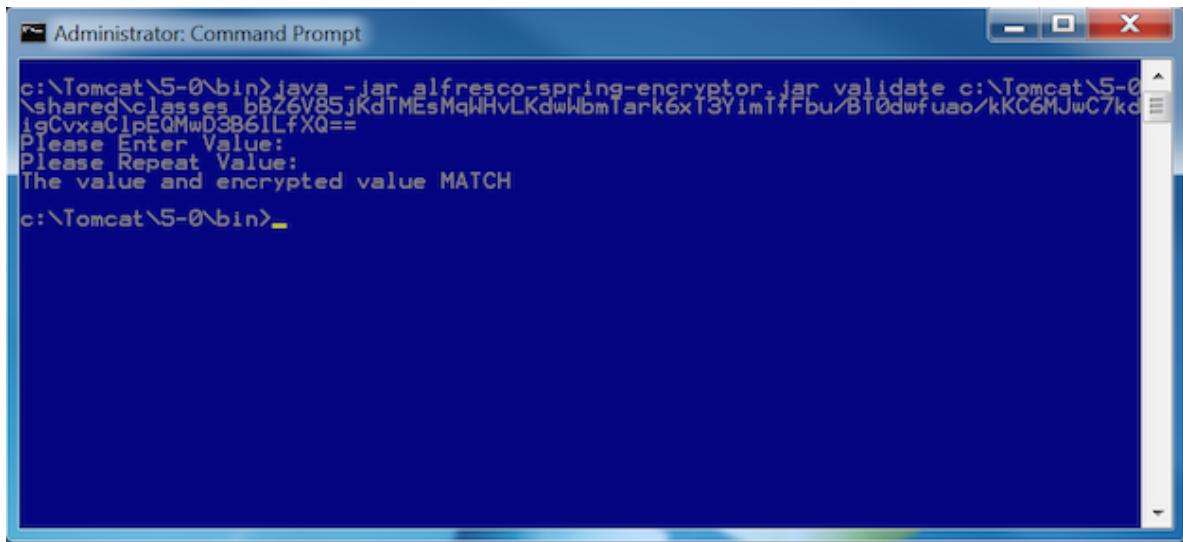
c:\Tomcat\5-0\bin>
```

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.



- 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.

- 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.

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 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
<code>alfrescoNtlm</code>	Native Alfresco authentication	Yes, NTLM	Yes	No
<code>ldap</code>	Authentication and user registry export through the LDAP protocol (for example, OpenLDAP)	No	No	Yes
<code>ldap-ad</code>	Authentication and user registry export from Active Directory through the LDAP protocol	No	No	Yes
<code>passthru</code>	Authentication through a Windows domain server	Yes, NTLM	Yes	No
<code>kerberos</code>	Authentication through a Kerberos realm	Yes, SPNEGO	Yes	No
<code>external</code>	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.
- ⚠ Support for Microsoft Office depends on the authentication mechanism provided by the `external` subsystem. See [How is Microsoft SharePoint support related to external authentication?](#) on page 184 for more information.



If you are using a proxy (load balancer) with Kerberos authentication, either:

- Use the `external` authentication subsystem on Alfresco and set up the proxy to implement `kerberos`
- Set up the `kerberos` authentication subsystem on Alfresco and create the Service Principal Name (SPN) in Active Directory to include the proxy DNS name

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

Alfresco authentication chain

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 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.

Default authentication chain

The default product configuration has a simple chain with one member. This is an instance of the alfrescoNt1m subsystem type with and ID of alfrescoNt1m1.

This is expressed in the built-in defaults in the `repository.properties` file as:

```
authentication.chain=alfrescoNt1m1:alfrescoNt1m
```

You can configure the properties of alfrescoNt1m1 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

You can add to or completely replace the default authentication chain.

Chained functions combine authentication subsystems. The chain is controlled by the `authentication.chain` global property. When a user logs in, Alfresco tries the user's credentials against each of the subsystems in the order specified in the chain, until the credentials are accepted (the login is successful) or until each subsystem has been tried (and the login fails).

Some functions cannot be chained (passthru function), for example, CIFS authentication, and NTLM / SPEGNO based Single Sign-On (SSO). These functions are handled by the first subsystem in the chain that supports that function and has it enabled. This means that only a subset of your users might be able to use SSO and CIFS.

1. Open the `alfresco-global.properties` file.
2. Locate, or if it does not already exist, create the `authentication.chain` global property.

This is a comma separated list of the form:

```
instance_name1:type1,...,instance_namen:typen
```

for example,

```
authentication.chain=alfrescoNtlm1:alfrescoNtlm
```

3. Set the property to the required values.

The default authentication chain specifies one instance of the `alfrescoNtlm` subsystem type with ID `alfrescoNtlm1`.

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, a new authentication subsystem instance called `ldap1` is created and added to the end of the authentication chain.

4. Save the file.

The following examples specify an advanced Active Directory chain, and an advanced LDAP chain.

You can integrate Alfresco with Active Directory so that:

- Built-in Alfresco users and Windows users can log in, with Alfresco taking precedence
 - The Windows domain server handles CIFS authentication directly
 - LDAP synchronizes user and group details
1. Configure the following authentication chain:

```
alfrescoNtlm1:alfrescoNtlm,passthru1:passthru,ldap1:ldap
```

2. 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 following properties:

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
```

You can integrate Alfresco with two LDAP directories so that:

- User passwords are validated directly against the LDAP servers for web, SharePoint and FTP login
 - The CIFS server is deactivated because neither server can handle CIFS-style authentication
 - LDAP is used to synchronize user and group details from both directories
 - Users in the first directory, `ldap1`, take precedence over those in the second directory, `ldap2`
1. In the `alfresco-global.properties`, specify this setting:

```
authentication.chain=ldap1:ldap,ldap2:ldap
```

2. Copy `<configRootShare>/classes/alfresco/subsystems/Authentication/ldap/ldap-authentication.properties` to both the `<classpathRoot>/alfresco/extension/subsystems/Authentication/ldap/ldap1/ldap-authentication.properties` and `<classpathRoot>/alfresco/extension/subsystems/Authentication/ldap/ldap2/ldap-authentication.properties` files.
3. Edit the properties for `ldap1` and `ldap2` with appropriate settings to complete the configuration. See [LDAP configuration properties](#) on page 203 for information on each of the properties.

Configuring authentication subsystems

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 required configuration of the subsystem instance.

Configuring external authentication

Use this information to enable the external authentication subsystem using the `alfresco-global.properties` file and the Admin Console in Share.

Configuring/enabling external authentication subsystem using the `alfresco-global.properties` file:

To enable external authentication subsystem:

1. Open the `alfresco-global.properties` file.
2. Set the following properties to enable external authentication:

```
authentication.chain=external1:external
external.authentication.proxyUserName=
external.authentication.enabled=true
external.authentication.defaultAdministratorUserNames=admin
external.authentication.proxyHeader=X-Alfresco-Remote-User
```



The default setting for `external.authentication.proxyUserName` is `alfresco-system`. This should only be specified if you are using SSL. See [External authentication basics](#) on page 181 for more information.

3. Save the `alfresco-global.properties` file.
4. Restart the Alfresco server.

For more information on the external authentication properties, see [external configuration properties](#).

Configuring/enabling external authentication subsystem using the Share Admin Console:

To enable external authentication subsystem using the Share Admin Console, see [configuring external authentication](#).

External authentication basics

Use this information to understand the differences between external authentication and Single Sign-On (SSO), and for Microsoft SharePoint Protocol support limitations.

What is external authentication?

The external authentication subsystem can be used to integrate Alfresco with any external authentication system. External authentication is set with the `authentication.chain` parameter in your `alfresco-global.properties` file to use the external authentication subsystem.

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.

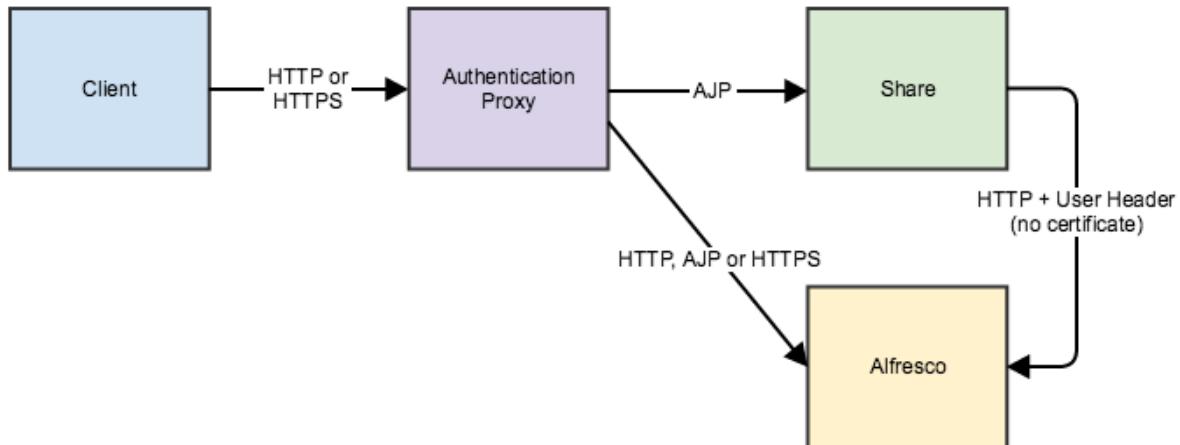
-  Activating external authentication makes Alfresco accept external authentication tokens. Make sure that no untrusted direct access to Alfresco's HTTP or AJP ports is allowed.

Here are two scenarios where external authentication is configured with Alfresco and Share. In both scenarios, an HTTP or HTTPS request is sent to an authentication proxy. If authentication is OK, the proxy passes the request to Share using the AJP protocol.

In the first scenario, the Share endpoint-url (`http://localhost:8080/alfresco/wcs`) sends the request directly to Alfresco using HTTP and a User Header. No certificate is used and the `external.authentication.proxyUserName` is blank:

```
external.authentication.proxyUserName=
```

Alfresco trusts the header (defined by `external.authentication.proxyHeader`) sent by Share. This scenario is typically used if you want to prohibit direct access to Alfresco and enforce using the proxy, for example, by using firewall rules to the proxy.

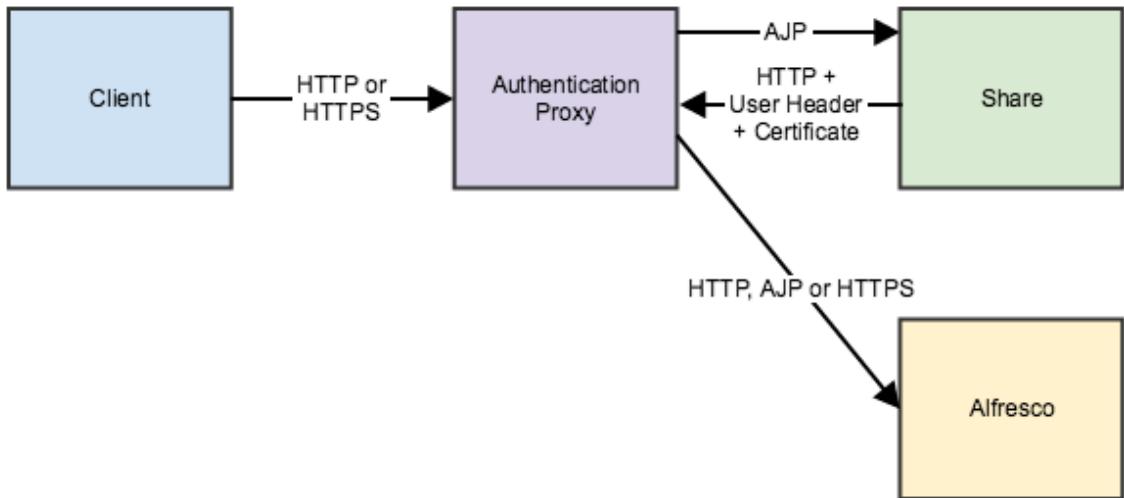


In the second scenario, the Share endpoint-url (`http://your.server.com/alfresco/wcs`) sends the request back to Apache, using HTTP and a User Header (defined by `external.authentication.proxyHeader`), and a certificate. `external.authentication.proxyUserName` is set:

```
external.authentication.proxyUserName=alfresco-system
```

Apache uses the certificate to check that the request is coming from Share with the correct user (that is, the value of `external.authentication.proxyUserName`) and forwards the request to

Alfresco. This scenario is typically used to allow direct access to Alfresco, using HTTPS and the originator (the proxy) sends a client certificate when establishing the SSL tunnel.



[Default authentication chain on page 178](#) and [Configuring external authentication on page 181](#) provide more information on the parameter and the external authentication subsystem.

Using the external authentication subsystem means that:

- The complexity of authentication moves to an external software layer (a proxy). Alfresco listens to the authenticated user name that it receives using a custom HTTP header, or it reads the CGI REMOTE_USER variable that can be passed using the AJP protocol. [Configuring external authentication on page 237](#) provides more information on configuring the external authentication subsystem.
- Most of the responsibility for authentication is not controlled by Alfresco, but controlled by the external software layer. Unless there is a problem when the authenticated user name is transmitted to Alfresco, the issue is located in the external software layer. In these cases, work with your proxy vendor or implementer of the authentication proxy to resolve the issue.

How is Single-Sign On (SSO) related to external authentication?

SSO is a property of an authentication scheme. You can use more than one method to set up SSO. For example:

- If you are using Kerberos, you can use either the `kerberos` authentication subsystem, or the external authentication subsystem with a proxy that handles Kerberos authentication.
- If you are using NTLM, you can use either the `alfrescoNtlm` or `passthru` authentication subsystems, or the external authentication subsystem with a proxy that handles NTLM authentication.
- If you are using CAS, you must use the external authentication subsystem with a proxy that handles CAS authentication.

In summary, external authentication and SSO are not interdependent: you can set up external authentication that is not SSO (for example, using an Apache proxy with a `mod_auth_basic` setting), and you can set up an SSO system that is not using the external authentication subsystem (for example, using the `kerberos` authentication subsystem).

See [Authentication subsystem types on page 176](#) for a listing of the authentication subsystems and the features that they support.

How is Microsoft SharePoint support related to external authentication?

There are some limitations when using Microsoft SharePoint support (as provided by Alfresco Office Services) with the Alfresco external authentication subsystem. External authentication can work well when using a web browser client, but not when using the MS Office client. This is because no authentication information is sent with the file URL, and MS Office does not store authentication information, so starts a new authentication process.

An example of this is when using CAS. CAS authenticates using an HTML form and a web browser that follows an HTTP redirect. The web authentication works correctly, but MS Office authentication will not work because it does not permit completion of the form. This problem is caused by the limited set of authentication protocols that MS Office supports.

MS Office supports the following authentication mechanisms:

- HTTP Basic
- HTTP Digest
- NTLM
- Kerberos

NTLM and Kerberos can be used in an SSO environment.

External configuration properties

The external subsystem supports a number of 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. 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.

-  The default setting for `external.authentication.proxyUserName` is alfresco-system, but this should only be specified if you are using SSL. See [External authentication basics](#) on page 181 for more information.

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.

Using Alfresco with CAS authentication through Apache mod_auth_cas

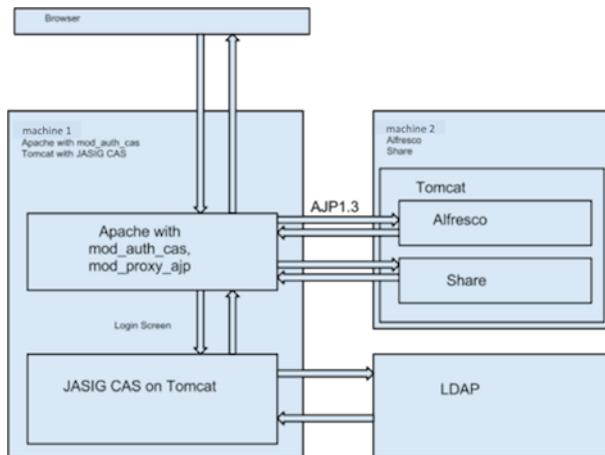
This information provides an overview on how to install and use Alfresco on a Red Hat Enterprise Linux server that uses a Central Authentication Service (CAS) server as a Single Sign-On (SSO) solution.

-  CAS 3.3.5 is the only version that has been confirmed to work with this configuration of Alfresco. CAS 3.4.3.1 requires changes to this configuration.

Overview of using Alfresco with CAS authentication

use this information to configure Alfresco to work with Jasig CAS authentication through Apache mod_auth_cas.

As shown in the diagram, the approach uses two separate machines: machine 1, the SSO server, with Apache and CAS on Tomcat and machine 2, the Alfresco server with its own Tomcat containing the Alfresco and Share web applications.



Install supporting tools

Follow these steps to prepare machine 1, the SSO server.

1. Download the Oracle JDK 7 rpm.bin from <http://www.oracle.com/technetwork/java/index.html> and run it to install Oracle JDK 7.
2. Download and extract the following under /opt.
 - Apache Maven 2.2
 - Apache Tomcat 7
3. For authentication, we use the Apache HTTP Server rather than Tomcat, so in /opt/apache-tomcat-6.0.35/conf/server.xml, disable Tomcat authentication on the AJP 1.3 connector by setting tomcatAuthentication=false:


```
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443"
tomcatAuthentication="false"/>
```
4. Edit /root/.bashrc to set up your environment as shown below:


```
export JAVA_OPTS="-Xmx1024M -XX:MaxPermSize=128M "
export JAVA_HOME=/usr/java/latest
export MVN_HOME=/opt/apache-maven-2.2.1
export CATALINA_HOME=/opt/apache-tomcat-6.0.35

export PATH=$JAVA_HOME/bin:$CATALINA_HOME/bin:$MVN_HOME/bin:$PATH
```

Install required packages

There are a number of packages required by machine 1 to enable Alfresco to work with mod_auth_cas.

The installation requires an Apache server with mod_ssl installed, as well as the packages required to compile and install other apache modules. So, ensure the following packages are installed on machine 1:

- httpd
- mod_ssl

- httpd-devel
- apr
- apr-devel
- apr-util
- apr-util-devel
- subversion

 Ensure that Red Hat's Tomcat packages are not installed, as CAS and Alfresco will require their own Tomcat 7 installation.

Set up Certificate Authority and issue Server and Client Certificates

The topic describes the instructions on how to set up a Certificate Authority (CA) and issue Server and Client Certificates on machine 1.

1. Open the /etc/pki/tls/openssl.cnf file in a text editor. Edit the [req_distinguished_name] section so that it has defaults appropriate for your organization. For example:

```
countryName_default = GB
stateOrProvinceName_default = Berkshire
localityName_default = Maidenhead
0.organizationName_default = Alfresco Software Inc.
```

2. Create the self-signed certificate for Alfresco CA. Use the exact paths listed here, as they are already referenced in openssl.cnf. For CA certificate, use any name that best describes your CA, for example Alfresco Demo.

```
mkdir /etc/pki/CA/{certs,crl,newcerts,private}
touch /etc/pki/CA/index.txt
echo 01 > /etc/pki/CA/serial
cd /etc/pki/CA/
umask 077
openssl genrsa -out private/cakey.pem -des3 2048
openssl req -new -x509 -days 365 -key private/cakey.pem -out cacert.pem -
subj '/CN=Alfresco Demo'
```

The following certificates are issued by CA:

```
/etc/pki/CA/private/cakey.pem - CA key
/etc/pki/CA/cacert.pem - CA certificate
```

3. Ensure that certificates issued by Alfresco CA are trusted by the Apache HTTP server. We need to create a symbolic link to the certificate using a computed hash to add it to the chain of trust.

```
cp cacert.pem /etc/pki/tls/certs/
cd /etc/pki/tls/certs/
ln -s cacert.pem `openssl x509 -hash -noout -in cacert.pem`.0
```

4. Replace the HTTP server's test certificate with the certificate issued by Alfresco CA. The advantages of it being issued by the same CA are that fewer certificates need to be added to Alfresco Share's truststore later. When prompted for a certificate subject (CN), you must specify the external DNS name of machine 1. The use of -nodes option avoids the need to enter the key password every time Apache is started.

```
cd /etc/pki/tls/certs
openssl req -nodes -new -out localhost.csr -keyout ../private/
localhost.key
openssl x509 -req -days 365 -in localhost.csr -CA /etc/pki/CA/cacert.pem
-CAkey /etc/pki/CA/private/cakey.pem -set_serial 01 -out localhost.crt
rm localhost.csr
```

5. Create Client Certificates that will be used in Alfresco Share to access repository. The Client Certificate securely identifies the Alfresco Share application to the Alfresco

repository. You need to protect the private key with a password. Also, export the key and its certificate chain to a password protected PKCS12 keystore `alfresco-system.p12` in the Tomcat classspath so that it can be used by the Share application. Use the same password for both the key and the keystore. For the subject name, use `alfresco-system`.

```
openssl genrsa -des3 -out ../private/alfresco-system.key 1024
openssl req -new -key ../private/alfresco-system.key -out alfresco-
system.csr -subj '/CN=alfresco-system'
```

Sign the client certificate with our CA certificate:

```
openssl x509 -req -days 365 -in alfresco-system.csr -CA /etc/pki/CA/
cacert.pem -CAkey/etc/pki/CA/private/cakey.pem -set_serial 02 -out
alfresco-system.crt
```

Package the client private and public keys in a P12:

```
openssl pkcs12 -export -out alfresco-system.p12 -in alfresco-system.crt -
inkey ../private/alfresco-system.key -certfile /etc/pki/CA/cacert.pem
```

Finally, copy `alfresco-system.p12` to machine 2, the Alfresco server, into the `/opt/alfresco/tomcat/shared/classes/alfresco/web-extension/` folder.

Build and install mod_auth_cas

Use this information to build and install `mod_auth_cas` on machine 1.

1. Use subversion to check out the latest `mod_auth_cas` source.

```
cd /root
svn co https://source.jasig.org/cas-clients/mod_auth_cas/tags/
mod_auth_cas-1.0.9.1 mod_auth_cas-1.0.9.1
cd mod_auth_cas-1.0.9.1
./configure; make; make install
```

2. Create a directory for `mod_auth_cas` to store cookie data. Note that this must be writeable by the Apache user.

```
mkdir /tmp/cas
chown apache:apache /tmp/cas
chmod 0700 /tmp/cas
```

Configure mod_auth_cas

Follow these steps to configure `mod_auth_cas`.

Configure `mod_auth_cas` by creating a file `/etc/httpd/conf.d/mod_auth_cas.conf` with the following code. This ensures that Alfresco, Share and example applications are protected by CAS. Remember to edit the host name in `CASLoginURL` and `CASValidateURL` appropriately.

```
LoadModule auth_cas_module modules/mod_auth_cas.so
CASCookiePath /tmp/cas/
CASLoginURL https://rhel-CAS-112/cas/login
CASValidateURL https://rhel-CAS-112/cas/serviceValidate
CASValidateServer Off
CASDebug On
CASCertificatePath /etc/pki/tls/certs

<LocationMatch ^/alfresco/(?!service/|service$|webdav/|webdav$|s/|s$|
scripts/|css/|images/).*>
AuthType CAS
AuthName "CAS"
require valid-user
CASScope /alfresco
</LocationMatch>

<Location /share>
AuthType CAS
AuthName "CAS"
```

```

require valid-user
CASScope /share
</Location>

<Location /examples>
AuthType CAS
AuthName "CAS"
require valid-user
CASScope /examples
</Location>

```

 **rhel-CAS-112** is the DNS name of machine 1, the host where JASIG CAS is set up. This name must match the CN of the server certificate created in the [Set up Certificate Authority and issue Server and Client Certificates](#) topic.

Configure, build and install Jasig CAS Server

Follow these steps to build your own pre-configured CAS Server on machine 1 using Maven that integrates with Alfresco's authentication systems.

1. Execute the following commands to set the appropriate directory structure.

```

cd /root
mkdir -p custom-cas-server/src/main/webapp/WEB-INF/classes
cd custom-cas-server

```

2. Create the Maven Project Object Model (POM) for the customized server. Create a file `pom.xml` that pulls in the required CAS Server dependencies. Note that this is using CAS 3.3.5. Now, include LDAP and X509 certificate support as shown below:

```

<?xml version="1.0"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://
maven.apache.org/maven-v4_0_0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>org.alfresco.cas</groupId>
    <artifactId>alfresco-cas</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>war</packaging>
    <name>Alfresco CAS webapp</name>
    <organization>
        <name>Alfresco</name>
        <url>http://www.alfresco.com</url>
    </organization>
    <description>Alfresco's configuration of the JA-SIG CAS server.</description>
    <dependencies>
        <dependency>
            <groupId>org.jasig.cas</groupId>
            <artifactId>cas-server-webapp</artifactId>
            <version>3.3.5</version>
            <type>war</type>
        </dependency>

        <dependency>
            <groupId>org.jasig.cas</groupId>
            <artifactId>cas-server-core</artifactId>
            <version>3.3.5</version>
        </dependency>

        <!-- if you need LDAP handler -->
        <dependency>
            <groupId>org.jasig.cas</groupId>
            <artifactId>cas-server-support-ldap</artifactId>
            <version>3.3.5</version>
        </dependency>
    </dependencies>

```

```

<!-- if you need X509 handler -->
<dependency>
    <groupId>org.jasig.cas</groupId>
    <artifactId>cas-server-support-x509</artifactId>
    <version>3.3.5</version>
</dependency>

</dependencies>
<build>
    <finalName>cas</finalName>
    <plugins>
        <plugin>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>RELEASE</version>
            <configuration>
                <source>1.5</source>
                <target>1.5</target>
            </configuration>
        </plugin>
    </plugins>
</build>
<repositories>
    <repository>
        <id>jasig-repository</id>
        <name>JA-SIG Maven2 Repository</name>
        <url>http://developer.ja-sig.org/maven2</url>
    </repository>
</repositories>
<reporting>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-project-info-reports-plugin</
artifactId>
        </plugin>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-javadoc-plugin</artifactId>
        </plugin>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-changelog-plugin</artifactId>
        </plugin>
    </plugins>
</reporting>
</project>

```

3. Add the files that you need to customize. Create the main Spring configuration file in `src/main/webapp/WEB-INF/deployerConfigContext.xml`.

In the example below:

- the LDAP directory is Active Directory, so the user names are in the form of User Principal Names (UPNs). These consist of a user name followed by an @ sign followed by the UPN suffix. This indicates that the `userDn` property of `contextSource` and the `filter` property of `FastBindLdapAuthenticationHandler` must use this same UPN format. For any other type of LDAP directory, you may need to bind with a full Distinguished Name (DN), in which case `userDn` should also be a DN and you will have to use `BindLdapAuthenticationHandler` instead of `FastBindLdapAuthenticationHandler` in order to resolve a user's DN from their user ID.
- authentication is enabled via SSL certificates with `X509CertificateCredentialsToIdentifierPrincipalResolver` and

`X509CredentialsAuthenticationHandler`. This enables the Alfresco Share to securely authenticate itself as a client of Alfresco web application.

- an extra `x509Check` bean is used. This is required by the Spring web flow configuration to automatically forward clients with a trusted SSL certificate.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:p="http://www.springframework.org/schema/p"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-2.0.xsd">

    <!-- Preconfigure our LDAP directory -->
    <bean id="contextSource"
        class="org.springframework.ldap.core.support.LdapContextSource">
        <property name="pooled" value="true" />
        <property name="urls">
            <list>
                <value>ldap://ldap.host.com:389/</value>
            </list>
        </property>
        <property name="userDn" value="alfresco@domain" />
        <property name="password" value="secret" />
        <property name="baseEnvironmentProperties">
            <map>
                <entry>
                    <key>
                        <value>java.naming.security.authentication</value>
                    </key>
                    <value>simple</value>
                </entry>
            </map>
        </property>
    </bean>

    <bean id="authenticationManager"
        class="org.jasig.cas.authentication.AuthenticationManagerImpl">
        <property name="credentialsToPrincipalResolvers">
            <list>
                <bean
                    class="org.jasig.cas.adaptors.x509.authentication.principal.X509CertificateCreden
                    <property name="identifier" value="$CN" />
                </bean>
                <bean
                    class="org.jasig.cas.authentication.principal.UsernamePasswordCredentialsToPrinci
                >
                    <bean
                    class="org.jasig.cas.authentication.principal.HttpBasedServiceCredentialsToPrinci
                >
                    </list>
                </property>

                <property name="authenticationHandlers">
                    <list>
                        <bean
                            class="org.jasig.cas.authentication.handler.support.HttpBasedServiceCredentialsAu
                            <property name="httpClient" ref="httpClient" />
                        </bean>
                        <bean
                            class="org.jasig.cas.adaptors.x509.authentication.handler.support.X509Credentials
                            <property name="trustedIssuerDnPattern"
                                value="^.*CN=Dave's Certificate Authority, O=Alfresco Software Inc\., L=Maidenhead, ST=Berkshire, C=GB\$" />
                        </bean>
                    </list>
                </property>
            </list>
        </property>
    </bean>

```

```

        <bean
class="org.jasig.cas.adaptors.ldap.FastBindLdapAuthenticationHandler">
            <property name="filter" value="%u@domain" />
            <property name="contextSource" ref="contextSource" />
        </bean>
    </list>
</property>
</bean>

<bean id="userDetailsService"
class="org.springframework.security.userdetails.memory.InMemoryDaoImpl">
    <property name="userMap">
        <value>
        </value>
    </property>
</bean>

<bean id="attributeRepository"
class="org.jasig.services.persondir.support.StubPersonAttributeDao">
    <property name="backingMap">
        <map>
            <entry key="uid" value="uid" />
            <entry key="eduPersonAffiliation"
value="eduPersonAffiliation" />
            <entry key="groupMembership" value="groupMembership" />
        </map>
    </property>
</bean>

<bean id="serviceRegistryDao"
class="org.jasig.cas.services.InMemoryServiceRegistryDaoImpl" />

<!-- Extra x509 bean -->
<bean id="x509Check" p:centralAuthenticationService-
ref="centralAuthenticationService"

class="org.jasig.cas.adaptors.x509.web.flow.X509CertificateCredentialsNonInteractive"
    <property name="centralAuthenticationService"
ref="centralAuthenticationService" />
</bean>
</beans>
```

- Add your logging configuration to `src/main/webapp/WEB-INF/classes/log4j.properties`.

```

# For JBoss: Avoid to setup Log4J outside $JBoss_HOME/server/default/
deploy/log4j.xml!
# For all other servers: Comment out the Log4J listener in web.xml to
activate Log4J.
log4j.rootLogger=ERROR, stdout, logfile

log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d %p [%c] - <%m>%n

log4j.appender.logfile=org.apache.log4j.RollingFileAppender
log4j.appender.logfile.File=cas.log
log4j.appender.logfile.MaxFileSize=512KB
# Keep three backup files.
log4j.appender.logfile.MaxBackupIndex=3
# Pattern to output: date priority [category] - message
log4j.appender.logfile.layout=org.apache.log4j.PatternLayout
log4j.appender.logfile.layout.ConversionPattern=%d %p [%c] - %m%n

# WARNING: Setting the org.springframework logger to DEBUG displays debug
information about
```

```

# the request parameter values being bound to the command objects. This
# could expose your
# password in the log file. If you are sharing your log files, it is
# recommend you selectively
# apply DEBUG level logging on a an org.springframework.* package level
# (i.e. org.springframework.dao)
log4j.logger.org.springframework=WARN
#log4j.logger.org.springframework.web.servlet.i18n=DEBUG
#log4j.logger.org.springframework.web.servlet.view=DEBUG
#log4j.logger.org.quartz=DEBUG

log4j.logger.org.jasig=INFO
# WARNING: Setting the flow package to DEBUG will display
# the parameters posted to the login servlet including
# cleartext authentication credentials
log4j.logger.org.jasig.cas.web.flow=INFO
#log4j.logger.org.jasig.cas.authentication=DEBUG
#log4j.logger.org.jasig.cas.web.flow.TicketGrantingTicketCheckAction=DEBUG
#log4j.logger.org.jasig.cas.services.DefaultServiceRegistry=DEBUG
#log4j.logger.org.jasig.cas.services=DEBUG
log4j.logger.org.jasig.cas.adapters.X509AuthenticationHandler.support.X509Creden

```

5. Create a file `src/main/webapp/WEB-INF/login-webflow.xml` with the following code to ensure auto-redirect through the login screen with the client certificate that was exported to `alfresco-system.p12`.

```

<?xml version="1.0" encoding="UTF-8"?>
<flow xmlns="http://www.springframework.org/schema/webflow"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="
          http://www.springframework.org/schema/webflow
          http://www.springframework.org/schema/webflow/spring-
          webflow-1.0.xsd">

    <start-state idref="initialFlowSetup" />

    <action-state id="initialFlowSetup">
        <action bean="initialFlowSetupAction" />
        <transition on="success" to="ticketGrantingTicketExistsCheck" />
    </action-state>

    <decision-state id="ticketGrantingTicketExistsCheck">
        <if test="${flowScope.ticketGrantingTicketId != null}" then="hasServiceCheck" else="gatewayRequestCheck" />
    </decision-state>

    <decision-state id="gatewayRequestCheck">
        <if test="${externalContext.requestParameterMap['gateway'] != '' && externalContext.requestParameterMap['gateway'] != null && flowScope.service != null}" then="redirect" else="startAuthenticate" /> (!!!NOTE: This line is a part of the if statement and should be written in a single line)
    </decision-state>

    <decision-state id="hasServiceCheck">
        <if test="${flowScope.service != null}" then="renewRequestCheck" else="viewGenericLoginSuccess" />
    </decision-state>

    <decision-state id="renewRequestCheck">
        <if test="${externalContext.requestParameterMap['renew'] != '' && externalContext.requestParameterMap['renew'] != null}" then="startAuthenticate" else="generateServiceTicket" /> (!!!NOTE: This line is a part of the if statement and should be written in a single line)
    </decision-state>

```

```

<!--
  The "warn" action makes the determination of whether to redirect
  directly to the
      requested service or display the "confirmation" page to go back
  to the server.
-->
<decision-state id="warn">
  <if test="${flowScope.warnCookieValue}" then="showWarningView"
  else="redirect" />
</decision-state>

<action-state id="startAuthenticate">
  <action bean="x509Check" />
  <transition on="success" to="sendTicketGrantingTicket" />
  <transition on="error" to="viewLoginForm" />
</action-state>

<view-state id="viewLoginForm" view="casLoginView">
  <render-actions>
    <action bean="authenticationViaFormAction" method="setupForm"/>
    <action bean="authenticationViaFormAction" method="referenceData"/>
  </render-actions>
  <transition on="submit" to="bindAndValidate" />
</view-state>

<action-state id="bindAndValidate">
  <action bean="authenticationViaFormAction" />
  <transition on="success" to="submit" />
  <transition on="error" to="viewLoginForm" />
</action-state>

<action-state id="submit">
  <action bean="authenticationViaFormAction" method="submit" />
  <transition on="warn" to="warn" />
  <transition on="success" to="sendTicketGrantingTicket" />
  <transition on="error" to="viewLoginForm" />
</action-state>

<action-state id="sendTicketGrantingTicket">
  <action bean="sendTicketGrantingTicketAction" />
  <transition on="success" to="serviceCheck" />
</action-state>

<decision-state id="serviceCheck">
  <if test="${flowScope.service != null}" then="generateServiceTicket"
  else="viewGenericLoginSuccess" />
</decision-state>

<action-state id="generateServiceTicket">
  <action bean="generateServiceTicketAction" />
  <transition on="success" to ="warn" />
  <transition on="error" to="viewLoginForm" />
  <transition on="gateway" to="redirect" />
</action-state>

<!--
  the "viewGenericLogin" is the end state for when a user attempts to
  login without
      coming directly from a service. They have only initialized their
  single-sign on
      session.
-->
<end-state id="viewGenericLoginSuccess"
view="casLoginGenericSuccessView" />

<!--
  The "showWarningView" end state is the end state for when the user has
  requested

```

```

        privacy settings (to be "warned") to be turned on. It delegates
        to a view defines
            in default_views.properties that display the "Please click here
            to go to the service.
            " message.
-->
<end-state id="showWarningView" view="casLoginConfirmView" />

<!--
    The "redirect" end state allows CAS to properly end the workflow while
    still
        redirecting the user back to the service required.
-->
<end-state id="redirect" view="bean:dynamicRedirectViewSelector" />

<end-state id="viewServiceErrorView" view="viewServiceErrorView" />

    <end-state id="viewServiceSsoErrorView"
view="viewServiceSsoErrorView" />

<global-transitions>
    <transition to="viewServiceErrorView" on-
exception="org.springframework.webflow.execution.repository.NoSuchFlowExecutionExc
>
    <transition to="viewServiceSsoErrorView" on-
exception="org.jasig.cas.services.UnauthorizedSsoServiceException" />
        <transition to="viewServiceErrorView" on-
exception="org.jasig.cas.services.UnauthorizedServiceException" />
</global-transitions>
</flow>

```

- Finally, build your configured version of the CAS war file and plug it into Tomcat:

```

cd /root/custom-cas-server
mvn -Dmaven.test.skip=true package install
cp target/cas.war /opt/apache-tomcat-6.0.35/webapps/

```

Configure mod_proxy_ajp

Use this information to configure `mod_proxy_ajp` on machine 1.

- Configure `mod_proxy_ajp` by creating a file `/etc/httpd/conf.d/mod_proxy_ajp.conf` with the following code:

```

LoadModule proxy_ajp_module modules/mod_proxy_ajp.so

LogLevel Debug

ProxyPass /alfresco ajp://172.30.40.144:8009/alfresco
ProxyPassReverse /alfresco ajp://172.30.40.144:8009/alfresco
ProxyPass /share ajp://172.30.40.144:8009/share
ProxyPassReverse /share ajp://172.30.40.144:8009/share
ProxyPass /cas ajp://rhel-cas-112:8009/cas
ProxyPassReverse /cas ajp://rhel-cas-112:8009/cas

```

where 172.30.40.144 is machine 2, the Alfresco server.

 Remember to configure `tomcatAuthentication=false` attribute for AJP connector in `server.xml` in Alfresco's Tomcat.

- Now, configure `mod_ssl` by editing `/etc/httpd/conf.d/ssl.conf` by adding the following lines near the end, just before the closing `</VirtualHost>` tag in order to mount the CAS and examples applications on the HTTPS port. Note that we also switch on client certificate verification and set the `SSLCACertificatePath` so that our CA certificate will be trusted by Apache. We set the `SSLOptions` necessary to ensure that Apache forwards client certificate information to Tomcat.

```

ProxyPass /cas ajp://rhel-cas-112:8009/cas
ProxyPassReverse /cas ajp://rhel-cas-112:8009/cas

```

```

ProxyPass /examples ajp://rhel-cas-112:8009/examples
ProxyPassReverse /examples ajp://rhel-cas-112:8009/examples

SSLVerifyClient optional
SSLCACertificatePath /etc/pki/tls/certs/
SSLOptions +StdEnvVars +ExportCertData

```

Configure Alfresco and Share to use SSO external authentication using AJP

Use this information to configure Alfresco and Share on machine 2 to use SSO external authentication with AJP.

1. Run the Alfresco installation wizard, installing it to, for example, /opt/alfresco directory.
2. Configure the AJP connector in `server.xml` in the /opt/alfresco/tomcat/conf directory. You must set `tomcatAuthentication=false`.

```
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443"
    tomcatAuthentication="false" />
```

3. Configure the `alfresco-global.properties` file at /opt/alfresco/tomcat/shared/classes/alfresco-global.properties as shown below:

```
authentication.chain=external1:external
# Note! This must be set up if you are using client certificate
external.authentication.proxyUserName=
external.authentication.proxyHeader=X-Alfresco-Remote-User
```

 The default setting for `external.authentication.proxyUserName` is `alfresco-system`. This should only be specified if you are using SSL. See [External authentication basics](#) on page 181 for more information.

4. Configure the Share default port by following the instructions specified in the [Configuring the Share default port](#) topic. Remember to replace `http://localhost:8080/` with `http://hostname-of-the-alfresco-repository/`.

Test it out

Follow these steps to ensure that CAS is authenticating Alfresco Explorer and Share applications properly.

1. Start up the servers and check for errors in their logs.

```
catalina.sh start
/etc/init.d/httpd start
```

2. Browse to the CAS login page at <http://your.host.com/examples/jsp/snp/snoop.jsp>.

When you enter a valid LDAP username and password, you will be redirected back to the *Snoop JSP* sample. It should display the name of the user you logged in as in the `Remote User` field, among other attributes.

3. To ensure that CAS is authenticating Alfresco Share's client certificate correctly, install `alfresco-system.p12` into your browser, delete all your cookies and then navigate to `snoop.jsp` again. This time, you should see `alfresco-system` as the `Remote User` without even being prompted for a username and password.
4. If you have successfully performed all the previous steps, try out the Alfresco Explorer and Share applications at <http://your.host.com/alfresco> and <http://your.host.com/share>.

 Remember to remove the `alfresco-system` certificate from your browser before trying out the Alfresco Explorer and Share applications. If not, then carefully check all Apache HTTP Server and Tomcat log files, and set DEBUG logging in your `CAS log4j.properties`, if necessary.

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).

This task assumes that you have already set up external authentication, as specified in [External configuration properties](#) on page 184.

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.

In this uncommented Remote section:

- a. Set the alfrescoHeader connector to use the same value that you defined for your external SSO property in [External configuration properties](#) on page 184:
Change the <userHeader> property to the same value as the external.authentication.proxyHeader. This sets the same HTTP header value for both Alfresco Share and the Alfresco repository.
- b. Set the alfresco endpoint to use the alfrescoHeader connector:
 1. Change the <connector-id> value from alfrescoCookie to alfrescoHeader
 2. Change the <endpoint-url> value to your Alfresco server URL; for example, <http://localhost:8080/alfresco/wcs>.

 This is an example file with entries shown in bold for the properties that should be modified:

```

<!--
      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>
    <ssl-config>
```

```

        <keystore-path>alfresco/web-extension/alfresco-
system.p12</keystore-path>
            <keystore-type>pkcs12</keystore-type>
            <keystore-password> alfresco-system</keystore-password>

            <truststore-path> alfresco/web-extension/ssl-
truststore</truststore-path>
                <truststore-type>JCEKS</truststore-type>
                <truststore-password>password</truststore-password>

                <verify-hostname>true</verify-hostname>
            </ssl-config>

        <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>ChangeThisToExternalSsoSystemHeader</
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>

```

4. 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 HTTP or AJP ports is allowed.

You have configured Alfresco Share to use an external SSO.

Setting Alfresco SSO with client certificates

Use this information to set up Alfresco SSO with client certificates.

1. Setup Apache as proxy server in front of Alfresco and configure it to use SSL as described in [Configuring SSL for a production environment](#).
2. Activate external authentication as described in [Configuring external authentication](#).
3. To extend the SSL configuration in `httpd.conf` to request client authentication and forward the user name as HTTP header, add this configuration to the `<VirtualHost>` node:

```
SSLVerifyClient      require
SSLCACertificateFile /path/to/your/enterprise-CA.pem
RequestHeader        append X-Alfresco-Remote-User
                     "%{SSL_CLIENT_S_DN_Email}e"
```

This will accept all client certificates that have been signed by the CA identified by the certificate stored in `enterprise-CE.pem`. It will use the email address stored in this certificate as the user name in Alfresco.

Configuring alfrescoNtIml

`alfrescoNtIml` 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 `alfrescoNtIml` 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.

alfrescoNtIml configuration properties

The `alfrescoNtIml` 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.sso.fallback.enabled

If SSO fails, a fallback authentication mechanism is used. The default value is `true`.

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

Use this information 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 $2^*(\text{expected number of concurrent requests})$. 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 NTLM v2 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 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 Idap-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.

Edit the `alfresco-global.properties` file to specify your authentication method:

- a. Append an instance of `passthru` to the authentication chain.

Name the instance `passthru1`, and declare it by changing the `authentication.chain` property in 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`.

- b. Edit the `ldap.authentication.active` property in the `alfresco-global.properties` file as follows:

```
ldap.authentication.active=false
```

### *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 OpenLDAP and Oracle Directory Server, 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. Oracle'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 Oracle 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 Oracle 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. Oracle'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 ([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.userOrganizationallIdAttributeName**

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 `Idap` or `Idap-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 `Idap` type is preconfigured with defaults appropriate for OpenLDAP, whereas `Idap-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 `Idap-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 `Idap-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 `Idap-ad` instance came first.

This example uses an Active Directory server and configures an instance of the `Idap-ad` subsystem.

1. This example uses the second option to append an instance of `Idap-ad` to the authentication chain. This instance name is `Idap1` 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 `Idap-ad` and `Idap` 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 Idap-ad, which demonstrates the flexibility of Alfresco's LDAP infrastructure. Luckily, because Idap-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 Idap-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 'ldap'

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.

- 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.

- 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).

- 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

```

- 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
-  If you are using a proxy (load balancer) with Kerberos authentication, either:
- Use the `external` authentication subsystem on Alfresco and set up the proxy to implement `kerberos`
  - Set up the `kerberos` authentication subsystem on Alfresco and create the Service Principal Name (SPN) in Active Directory to include the proxy DNS name. With this option, the load balancer relays the `negotiate` headers to the Alfresco repository, but the client sees the proxy as a DNS name. You must set Active Directory to allow this by creating the SPN for the proxy.

For some scenarios on using Kerberos with a proxy, see [Load Balancers and Kerberos](#).

For some pointers and background information on JAAS, the Java Authentication and Authorization Service, refer to the following web sites:

- [Oracle Java SE Security](#)
- [JAAS](#)

### Kerberos configuration properties

To enable full Kerberos support in Alfresco, the CIFS server and the SSO authentication filters each need 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 name in upper case; for example, if the domain is `alfresco.org` then the realm should be `ALFRESCO.ORG`.

#### **kerberos.authentication.sso.enabled**

A value of `true` enables SPNEGO/Kerberos based Single Sign On (SSO) functionality in the web client. If the value is `false` and no other members of the authentication chain support SSO, password-based login is used.

#### **kerberos.authentication.sso.fallback.enabled**

If SSO fails, a fallback authentication mechanism is used. The default value is `true`.

#### **kerberos.authentication.authenticateCIFS**

A value of `true` enables Kerberos authentication in the CIFS server. If the value is `false` and no other members of the authentication chain support CIFS authentication, the CIFS server is disabled.

#### **kerberos.authentication.user.configEntryName**

The name of the entry in the JAAS configuration file that is 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 is 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 is used for web-based Single-Sign On (SSO). The default value `AlfrescoHTTP` is recommended.

#### **kerberos.authentication.defaultAdministratorUserNames**

A comma separated list of user names that are treated as administrators by default.

#### **kerberos.authentication.browser.ticketLogons**

Authentication with Alfresco using a ticket parameter in the request URL. The default value is `true`. Note that WebDAV URLs always accept ticket parameters.

## kerberos.authentication.stripUsernameSuffix

A value of `true` strips the `@domain` suffix from Kerberos authenticated user names in CIFS, SPP, WebDAV and the Web Client. A value of `false` 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.

 When configuring Kerberos on a cluster through a load balancer, use the proxy name as the Service Principal Names (SPN).

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 all -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`).

- a. 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>`.
- 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 HTTP/<host>.<domain>@<REALM> -pass <password> -mapuser
<domainnetbios>\http<host> -crypto all -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>
```

- k. In the Active Directory Users and Computers application, right click on the `http<host>` user and select **Properties**.
- l. 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 2012 R2 x64. 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 2012**, and then click **Raise**.

- m. 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`. If the file does not already exist (for example, if the Kerberos libraries are not installed on the target server), you must copy these over or create them from scratch. See [Kerberos Help](#) for more information on the `krb5.conf` file. 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
```

 Specify the realm in uppercase.

The Kerberos `ini` file for Linux is `/etc/krb5.conf`.

- b. Set up the Java login configuration file.

For JBoss 6.2, open the `$JBoss_HOME/standalone/configuration/standalone.xml` file.

In the `<subsystem xmlns="urn:jboss:domain:security:1.2">` section, add the following:

```
<security-domain name="alfresco" cache-type="default">
 <authentication>
 <login-module
 code="com.sun.security.auth.module.Krb5LoginModule"
 flag="sufficient"/>
 </authentication>
</security-domain>
```

Add the following security-domain sections:

```
<security-domain name="AlfrescoCIFS" cache-type="default">
 <authentication>
 <login-module code="com.sun.security.auth.module.Krb5LoginModule"
 flag="required">
 <module-option name="debug" value="true"/>
 <module-option name="storeKey" value="true"/>
 <module-option name="useKeyTab" value="true"/>
 <module-option name="doNotPrompt" value="true"/>
 <module-option name="isInitiator" value="false"/>
 <module-option name="keyTab" value="C:/etc/cifs<host>.keytab"/>
 <module-option name="principal" value="cifs/
<hostnetbios>.domain"/>
 </login-module>
 </authentication>
</security-domain>
<security-domain name="AlfrescoHTTP" cache-type="default">
 <authentication>
```

```

<login-module code="com.sun.security.auth.module.Krb5LoginModule"
flag="required">
 <module-option name="debug" value="true"/>
 <module-option name="storeKey" value="true"/>
 <module-option name="useKeyTab" value="true"/>
 <module-option name="doNotPrompt" value="true"/>
 <module-option name="isInitiator" value="false"/>
 <module-option name="keyTab" value="C:/etc/http<host>.keytab"/>
 <module-option name="principal" value="HTTP/<host>.<domain>"/>
</login-module>
</authentication>
</security-domain>
<security-domain name="ShareHTTP" cache-type="default">
 <authentication>
 <login-module code="com.sun.security.auth.module.Krb5LoginModule"
flag="required">
 <module-option name="debug" value="true"/>
 <module-option name="storeKey" value="true"/>
 <module-option name="useKeyTab" value="true"/>
 <module-option name="doNotPrompt" value="true"/>
 <module-option name="isInitiator" value="false"/>
 <module-option name="keyTab" value="C:/etc/http<host>.keytab"/>
 <module-option name="principal" value="HTTP/<host>.<domain>"/>
 </login-module>
 </authentication>
</security-domain>

```

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 other environments, in the Java security folder (for example, `C:/Alfresco/java/lib/security`), create a file named `java.login.config` with entries as shown below. 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`).

```

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 `java\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

Configure the Kerberos client authentication on Windows using Internet Explorer, WebDav, Firefox, and Chrome browsers.

### Kerberos client configuration for Internet Explorer

To configure Internet Explorer to use Kerberos authentication, rather than NTLM, ensure that:

- Alfresco web server is in the Local Intranet security zone.

Check **Tools > Internet Options > Security > Local Intranet > Sites > Advanced**, and then add the necessary domain name, for example, `http://server.com` or `http://*.company.com`.

- Automatic log on is enabled.

Check **Tools > Internet Options > Security > Local Intranet > Custom Level > User Authentication > Logon**, and then select **Automatic logon with current user name and password**.

### Kerberos client configuration for WebDav

To enable a Windows Vista or Windows 7 computer to use WebDav access to a fully qualified domain name (FQDN) site, ensure that you create a registry entry:

1. Click **Start**, type `regedit` in the **Start Search** box, and click **ENTER**.
2. Locate and click the following registry subkey:  
`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WebClient\Parameters`
3. From the **Edit** menu, point to **New**, and click **Multi-String Value**.
4. Type `AuthForwardServerList`, and then press **ENTER**.
5. From the **Edit** menu, click **Modify**.
6. In the **Value data** box, type the URL of the server that hosts the Web share, and click **OK**.

 You can type a list of URLs in the **Value data** box. For example, the following is a sample URL list:

```

http://*.domain.local
*.domain.local

```

7. Exit Registry Editor.
8. Restart the WebClient (WebDav) service after you modify the registry.

After creating this registry entry, WebDav works with the following URLs:

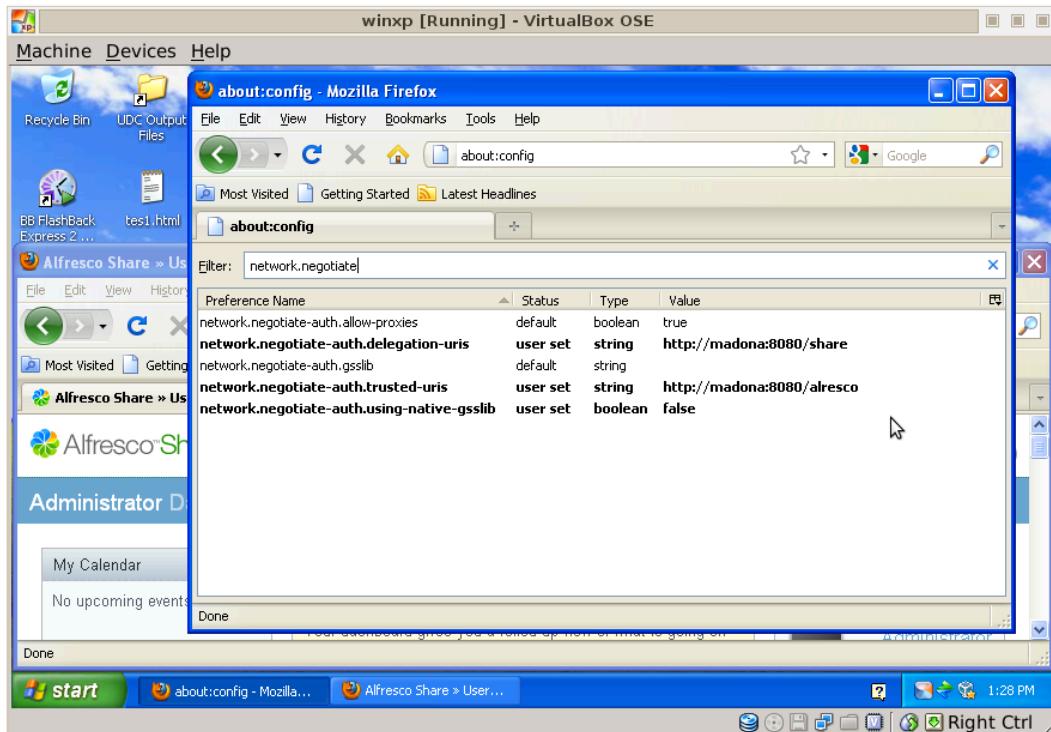
```
http://alfv0.domain.local:8080/alfresco/webdav
http://alfv0:8080/alfresco/webdav
```

## Kerberos client configuration for Firefox

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:



When using Firefox on Linux, add your Alfresco server name to `network.negotiate-auth.trusted-uris` and get a Kerberos ticket using the `kinit` command:

```
kinit -f <username>
```

For example, `kinit -f user1`, where `user1` is an Active Directory user. If the client and the server are on the same machine, go to the external interface. The loopback interface will not be able to authenticate. View your tickets using `klist`.

The ticket might correspond to a different user than your Linux user name.

## Kerberos client configuration for Chrome

When using Chrome on Linux as your client, follow these steps:

1. Create a ticket on the Linux client.

```
kinit -f -p user1
klist
Ticket cache: FILE:/tmp/krb5cc_1000 Default principal:
 user1@EXAMPLE.FOO
Valid starting Expires Service principal 14/12/2012 12:10 14/12/2012
 22:10 krbtgt/EXAMPLE.FOO@EXAMPLE.FOO renew until 15/12/2012 12:10
```

2. To use Alfresco Share, use:

```
google-chrome --auth-server-whitelist=madona:8080 --auth-negotiate-
delegate-whitelist=madona:8080
```

```
http://madona:8080/alfresco
```

When using Chrome on Windows to access Alfresco Share, if the command-line switch is not present, the permitted list consists of those servers in the Local Machine or Local Intranet security zone. This is the behavior in Internet Explorer. For example, when the host in the URL includes a ". " character, it is outside the Local Intranet security zone. Treating servers that bypass proxies as being in the Intranet zone is currently not supported.

On Windows, HTTP authentication is achieved by adding the Kerberos delegation server whitelist policy, `AuthNegotiateDelegateWhitelist`. Note that the `AuthNegotiateDelegateWhitelist` policy:

- Specifies the servers that Chrome may delegate to
- Has a Windows registry location of `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Google\Chrome\AuthNegotiateDelegateWhitelist`
- Has separate multiple server names with commas
- Allows wildcards (\*)
- If you do not set this policy, Chrome does not delegate user credentials, even if a server is detected as Intranet

To set the `AuthNegotiateDelegateWhitelist` policy, follow these steps:

1. Download the Administrative policy template from [http://dl.google.com/dl/edgedl/chrome/policy/policy\\_templates.zip](http://dl.google.com/dl/edgedl/chrome/policy/policy_templates.zip).
2. Use the command line, `gpedit.msc` to open the local group policy management.
3. In the Group Policy Editor console tree, navigate to **Local Computer Policy > Computer Configuration > Administrative Templates**.
4. Right click on **Administrative Templates**.
5. Click **Add/Remove Templates**.
6. Click the **Add** button.
7. Select `windows/adm/en-US/chrome.adm` from the `policy_templates.zip` download.
8. In the **Local Computer Policy Editor** console tree, navigate to **Local Computer Policy > Computer Configuration > Administrative Templates > Classic Administrative Templates (ADM) > Google > Google Chrome > Policies for HTTP Authentication > Kerberos delegation server whitelist**.
9. On the **Kerberos delegation server whitelist** window, click **Enabled**.
10. Specify your Share server name(s) as value in **Kerberos delegation server whitelist**.
11. To activate the policy, open Chrome.
12. Type `chrome://policy` to list the settings as viewed by Chrome.

### Configuring cross-domain support for Kerberos

In this task, we will prepare Active Directory Domain Services, configure SSO, and configure cross-domain support for Kerberos SSO.

1. Install Active Directory Domain Services.
  - a. Log in to your Windows server and start the **Server Manager**.
  - b. In the **Server Manager** menu, click **Roles > Add Roles**, and **Next**.
  - c. In the **Add Roles** wizard, select the **Active Directory Domain Services** role.
  - d. Click **Add Required Features** when prompted.
  - e. Click **Install** to complete the installation, and **Close**.
2. Configure Active Directory Domain Services.

- a. In the **Server Manager** menu, click **Roles > Active Directory Domain Services**, and click the link to start the **Active Directory Domain Services Installation Wizard**, and click **Next**.
  - b. On the **Choose a Deployment Configuration** window, click **Create a new domain in the forest**, and click **Next**.
  - c. Type your domain name in the **Name the Forest Root Domain** window.
  - d. In the **Set Forest Functional Level** window, choose the oldest operating system that you support in your network.
  - e. In the **Additional Domain Controller Options** window, ensure that **DNS server** checked.
  - f. Specify folders to contain the Active Directory controller database, log files, and **SYSVOL**, and click **Next**.
  - g. Choose a password for the **Restore Mode Administrator** account.  
This is an additional account (separate from the **Domain Administrator**) that is used for recovery.
  - h. Check **Reboot on completion** on the last window of the wizard.
- Your first Active Directory server is now installed and configured with a DNS server. Repeat [step 1](#) and [step 2](#) to install and configure your second Active Directory server.
3. Install and configure Alfresco with Kerberos SSO.  
Follow the steps in [Configuring Share Kerberos SSO](#) on page 223 to set this up.
  4. Configure cross-domain support by assigning a conditional forwarder for a domain name.
    - a. On your Windows server, open the **DNS Manager** (**Start > Administrative Tools > DNS**).
    - b. In the **DNS Manager** menu, select your DNS server and expand it to see a number of folders.
    - c. Click the **Conditional Forwarders** folder.
    - d. From the **DNS Manager** menu, click **Action > New Conditional Forwarder**, and the **New Conditional Forwarder** appears.
    - e. In the **DNS domain** field, type the fully qualified domain name (FQDN) that you want to use to forward queries.
    - f. Click the **IP address of the master servers** table and type the IP address of the server that you want to use to forward queries for the specified DNS domain, and click **OK**.
    - g. Restart your network adapter.

Your first Active Directory server now has a conditional forwarder assigned for a domain name. Repeat this step to set up a conditional forwarder on your second Active Directory server.
  5. Configure cross-domain support by creating a two-way forest trust between the Active Directory servers.
    - a. On your Windows server, open **Active Directory Domains and Trusts** (**Start > Administrative Tools > Active Directory Domains and Trusts**).
    - b. In the **Active Directory Domains and Trusts** menu, right-click the domain that you want to administer and click **Properties**.
    - c. From the Properties window, select the **Trusts** tab and click **New Trust**. The Trust wizard starts. Click **Next**.

- d. In the **Trust Name** window, type the Domain Name System (DNS) name, or NetBIOS name of the domain, and click **Next**.
- e. In the **Trust Type** window, select **Forest trust**, and click **Next**.
- f. In the **Direction of Trust** window, select **Two-way**, and click **Next**.
- g. In the **Sides of Trust** window, select **Both this domain and the specified domain**, and click **Next**.
- h. In the **User Name and Password** window, enter Administrator credentials for the trusted domain, and click **Next**.
- i. In the **Outgoing Trust Authentication Level-Local Forest** window, select **Forest-wide authentication**, and click **Next**.
- j. In the **Outgoing Trust Authentication Level-Specified Forest** window, select **Forest-wide authentication**, and click **Next**.
- k. In the **Trust Selection Complete** window, click **Next**.
- l. In the **Trust Creation Complete** window, click **Next**.
- m. In the **Confirm Outgoing Trust** window, select **Yes, confirm the outgoing trust**, and click **Next**.
- n. In the **Confirm Incoming Trust** window, select **Yes, confirm the incoming trust**, and click **Next**, and click **Finish**.

You can now log into Alfresco from the first trusted domain using Kerberos SSO, but not from the second domain.

6. Add realm information for the trusted domain into your `krb5.ini` file:

In the `[realms]` section, where `domain2.local` is the name of your second trusted domain:

```
[realms]
...
DOMAIN2.LOCAL = {
 kdc = ad2.domain2.local:88
 admin_server = ad2.domain2.local:749
 default_domain = domain2.local
}
```

and in the `[domain_realm]` section:

```
[domain_realm]
...
.domain2.local = DOMAIN2.LOCAL
domain2.local = DOMAIN2.LOCAL
```

7. Restart the Alfresco server.

When the server has restarted, check that you can access Alfresco Share from both domains.

## 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>

 </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>
 <ssl-config>
 <keystore-path>alfresco/web-extension/alfresco-
system.p12</keystore-path>
 <keystore-type>pkcs12</keystore-type>
 <keystore-password> alfresco-system</keystore-password>

 <truststore-path> alfresco/web-extension/ssl-truststore</
truststore-path>
 <truststore-type>JCEKS</truststore-type>
 <truststore-password>password</truststore-password>

 <verify-hostname>true</verify-hostname>
 </ssl-config>

 <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>

```

```
</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 (Oracle 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 2012 R2 x64.

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**, click **Windows Server 2012**, and then click **Raise**.
- Configure the client. See [Kerberos client configuration](#).

## 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.

For more information about the cron expression, see the [CronTrigger tutorial](#).

### **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.

## 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.
- Enables: Specifies if authentication is enabled or not.
- Synchronized: Specifies if the authentication chain is synchronized or not.
- Actions: 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.
-  You can only edit a directory after it has been added and saved. If you have not yet saved the entry, the only option available is **Remove**.
- 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.
  - To start the user directory sync of all users and groups, click **Run Synchronize**.
  - 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. You can manage the various subsystems using their configuration properties.

Click the relevant authentication directory for more information.

#### *Configuring OpenLDAP or Oracle Directory Server*

Use these instructions to configure OpenLDAP or Oracle Directory Server 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 OpenLDAP or Oracle Directory Server directory.

 You can only edit a directory after it has been added and saved. If you have not yet saved the entry, the only option available is **Remove**.

- You see the **Edit LDAP Directory** page.
- Set the configuration properties.

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

Synchronization property	Example setting	What is it?
<b>Security</b>	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 <a href="#">LDAP configuration properties</a> for more information.
<b>Default Administrator User Names</b>	-	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.
<b>Authenticate FTP</b>	Yes	This enables authentication for FTP access.
<b>Synchronization Enabled</b>	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.
<b>Security Principal Name</b>	cn=Manager,dc=company,dc=companyName	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.userNameFormat</code> but with a real user ID instead of %s.
<b>Security</b>	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 <a href="#">LDAP configuration properties</a> for more information.
<b>Group query</b>	(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 ).

Synchronization property	Example setting	What is it?
<b>Security Principal Credentials</b>	secret	This specifies the password for the default principal (only used for LDAP sync). Click <b>Show Password</b> to reveal the password. Click <b>Hide Password</b> to hide the password.
<b>User Search Base</b>	ou=People,dc=company,dc=com	This specifies the DN below which to run the user queries.
<b>Group Search Base</b>	ou=Groups,dc=company,dc=com	This specifies the DN below which to run the group queries.
<b>Person Differential Query</b>	(&(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.
<b>Person Query</b>	(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.

- Click **Save** to apply the changes you have made to the OpenLDAP or Oracle Directory Server directory.

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

### Configuring LDAP (Active Directory)

Use these instructions to configure LDAP-AD 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 LDAP (Active Directory) directory.

 You can only edit a directory after it has been added and saved. If you have not yet saved the entry, the only option available is **Remove**.

You see the **Edit LDAP-AD Directory** page.

4. Set the configuration properties.

Synchronization property	Example setting	What is it?
<b>Authentication Enabled</b>	Yes	This specifies that the directory will be used to authenticate users.
<b>User Name Format</b>	%s@domain	This specifies how to map the user identifier entered by the user to that passed through to LDAP.
<b>LDAP Server URL</b>	ldap://\$LDAP_HOST:\$LDAP_HOST_PORT	This specifies the URL of your LDAP server, containing its name and port. The standard ports for LDAP are 389 (and 636 for SSL)
<b>Security</b>	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 <a href="#">LDAP configuration properties</a> for more information.
<b>Default Administrator User Names</b>	Administrator	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.
<b>Authenticate FTP</b>	Yes	This enables authentication for FTP access.
<b>Synchronization Enabled</b>	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.
<b>Security Principal Name</b>	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.userNameFormat</code> but with a real user ID instead of %s.

Synchronization property	Example setting	What is it?
<b>Security</b>	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 <a href="#">LDAP configuration properties</a> for more information.
<b>Group query</b>	(objectclass=group)	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).
<b>Security Principal Credentials</b>	secret	This specifies the password for the default principal (only used for LDAP sync). Click <b>Show Password</b> to reveal the password. Click <b>Hide Password</b> to hide the password.
<b>User Search Base</b>	ou=People,dc=company,dc=com	This specifies the DN below which to run the user queries.
<b>Group Search Base</b>	ou=Groups,dc=company,dc=com	This specifies the DN below which to run the group queries.
<b>Person Differential Query</b>	(&(objectclass=user)(userAccountControl \:1.2.840.113556.1.4.803\:=512)(!(whenChanged<\={0})))	The query to select the objects that represent the users to import to Alfresco 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.
<b>Person Query</b>	(objectclass=user)	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.

- 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 pass-through (passthru) authentication*

Use these instructions to configure passthru authentication 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 Passthru directory.

 You can only edit a directory after it has been added and saved. If you have not yet saved the entry, the only option available is **Remove**.

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

4. Set the configuration properties.

Synchronization property	Example setting	What is it?
<b>Use Local Server</b>	No	This enables the local server to be used for passthru authentication by using loopback connections into the server.
<b>Map Unknown User to Guest</b>	No	This specifies whether unknown users are automatically logged in as the Alfresco guest user during SSO.
<b>Allow Guest Login</b>	No	This enables the guest logins to Alfresco.
<b>Administrator User Names</b>	-	This specifies a comma separated list of user names to be considered administrators by default.
<b>Authenticate FTP</b>	Yes	This enables passthru authentication for FTP access.
<b>Authenticate Domain</b>	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.

Synchronization property	Example setting	What is it?
<b>Authentication Servers</b>	-	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.
<b>Authentication Protocol Order</b>	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.
<b>Connection Timeout</b>	5000	This specifies the timeout value in milliseconds when opening a session to an authentication server. The default is 5000.
<b>Offline Check Interval</b>	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.

5. 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

Use these instructions to configure Kerberos 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 Kerberos directory.

 You can only edit a directory after it has been added and saved. If you have not yet saved the entry, the only option available is **Remove**.

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

4. Set the configuration properties.

Synchronization property	Example setting	What is it?
<b>User Config Entry Name</b>	Alfresco	This specifies the entry in the JAAS configuration file that should be used for password-based authentication. The recommended default value is Alfresco.
<b>Administrator User Names</b>	-	This specifies a comma separated list of user names to be considered administrators by default.
<b>CIFS Config Entry Name</b>	AlfrescoCIFS	This specifies an entry in the JAAS configuration file that should be used for CIFS authentication. The recommended default value is AlfrescoCIFS.
<b>Kerberos Authentication Realm</b>	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.
<b>CIFS Password</b>	secret	This specifies the password for the CIFS Kerberos principal. Click <b>Show Password</b> to reveal the password. Click <b>Hide Password</b> to hide the password.
<b>HTTP Config Entry Name</b>	AlfrescoHTTP	This specifies the entry in the JAAS configuration file used for web-based SSO. The recommended default value is AlfrescoHTTP.
<b>Strip Username Suffix</b>	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.
<b>HTTP Password</b>	secret	This specifies the password for the HTTP Kerberos principal. Click <b>Show Password</b> to reveal the password. Click <b>Hide Password</b> to hide the password.
<b>Authenticate FTP</b>	Yes	This enables authentication for FTP access.

5. 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

Use these instructions to configure external authentication 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, if no element of type **External** exists in the authentication chain list, follow the steps below to add a new **External** type element:
  - a. Specify a name in the **Name** text box.
  - b. Set type to **External**.
  - c. Click **Add**.
  - d. Click **Save** to add the new **External** type element in the authentication chain list.
4. In the **Authentication Chain** section, under **Actions**, click **Edit** corresponding to the External directory.

 You can only edit a directory after it has been added and saved. If you have not yet saved the entry, the only option available is **Remove**.

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

5. Set the configuration properties.

Synchronization property	Example setting	What is it?
<b>Authentication Enabled</b>	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.
<b>Proxy Username</b>	alfresco-system	This specifies the remote user that is considered as the proxy user.   The default setting for <code>external.authentication.proxyUsername</code> is <code>alfresco-system</code> . This should only be specified if you are using SSL. See <a href="#">External authentication basics</a> on page 181 for more information.
<b>Administrator User Names</b>	-	This specifies a comma separated list of user names to be considered administrators by default.
<b>Proxy Header</b>	X-Alfresco-Remote-User	This specifies the HTTP header that carries the name of a proxied user. The default is <code>X-Alfresco-Remote-User</code> .

Synchronization property	Example setting	What is it?
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.

- 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 alfrescoNtIml

Use these instructions to configure alfrescoNtIml 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 alfrescoNtIml1 directory.  
You see the **Edit Internal Alfresco Directory** page.
- 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.
Authenticate FTP	Yes	This enables authentication for FTP access.

- 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. Use this information to configure the synchronization subsystem.

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

Synchronization property	Example setting	What is it?
<b>Sync on Startup</b>	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.
<b>Sync When Missing People Login</b>	Yes	This triggers synchronization when a user, who does not yet exist in Alfresco, is successfully authenticated. The default is true.
<b>Allow Deletions</b>	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.
<b>Logging Interval</b>	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 log4j.properties: <code>log4j.logger.org.alfresco.repo.s...</code> The default is 100.
<b>Auto Create People On Login</b>	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 <b>Sync When Missing People Login</b> property). The default is true.
<b>Sync Changes Only</b>	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.

Synchronization property	Example setting	What is it?
<b>Import CRON Expression</b>	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.
<b>Sync Worker Threads</b>	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.
- 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.
- Click **Save** to apply the changes you have made to the authentication chain.  
If you do not want to save the changes, click **Cancel**.

## 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.

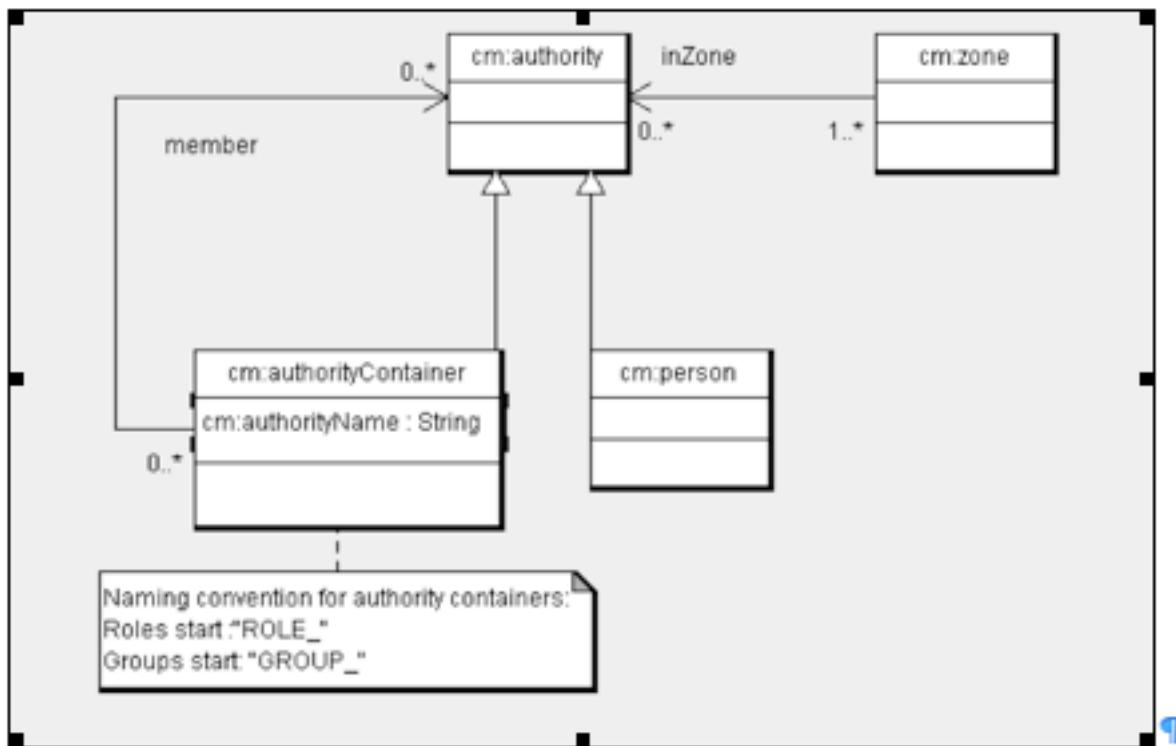
## Zones

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.

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 following code and add it to the `<extension>/custom-model-context.xml` file.

```

<?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="SiteService_security"
 class="org.alfresco.repo.security.permissions.impl.acegi.MethodSecurityInterceptor">
 <property name="authenticationManager"><ref
 bean="authenticationManager"/></property>
 <property name="accessDecisionManager"><ref
 bean="accessDecisionManager"/></property>
 <property name="afterInvocationManager"><ref
 bean="afterInvocationManager"/></property>
 <property name="objectDefinitionSource">
 <value>
 org.alfresco.service.cmr.site.SiteService.cleanSitePermissions=ACL_NODE.0.sys:base
 org.alfresco.service.cmr.site.SiteService.createContainer=ACL_ALLOW,AFTER_ACL_NODE
 org.alfresco.service.cmr.site.SiteService.createSite=ACL_METHOD.GROUP_SITE_CREATE
 org.alfresco.service.cmr.site.SiteService.deleteSite=ACL_ALLOW
 org.alfresco.service.cmr.site.SiteService.findSites=ACL_ALLOW,AFTER_ACL_NODE.sys:base
 org.alfresco.service.cmr.site.SiteService.getContainer=ACL_ALLOW,AFTER_ACL_NODE.sys:base
 org.alfresco.service.cmr.site.SiteService.listContainers=ACL_ALLOW,AFTER_ACL_NODE.sys:base
 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:base
 org.alfresco.service.cmr.site.SiteService.getSiteShortName=ACL_ALLOW,AFTER_ACL_NODE.sys:base
 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.sys:base
 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.listSiteMemberships=ACL_ALLOW
 </value>
 </property>
 </bean>
</beans>

```

```

 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>
</beans>
```

2. 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.
```

3. Save the file.
4. 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:

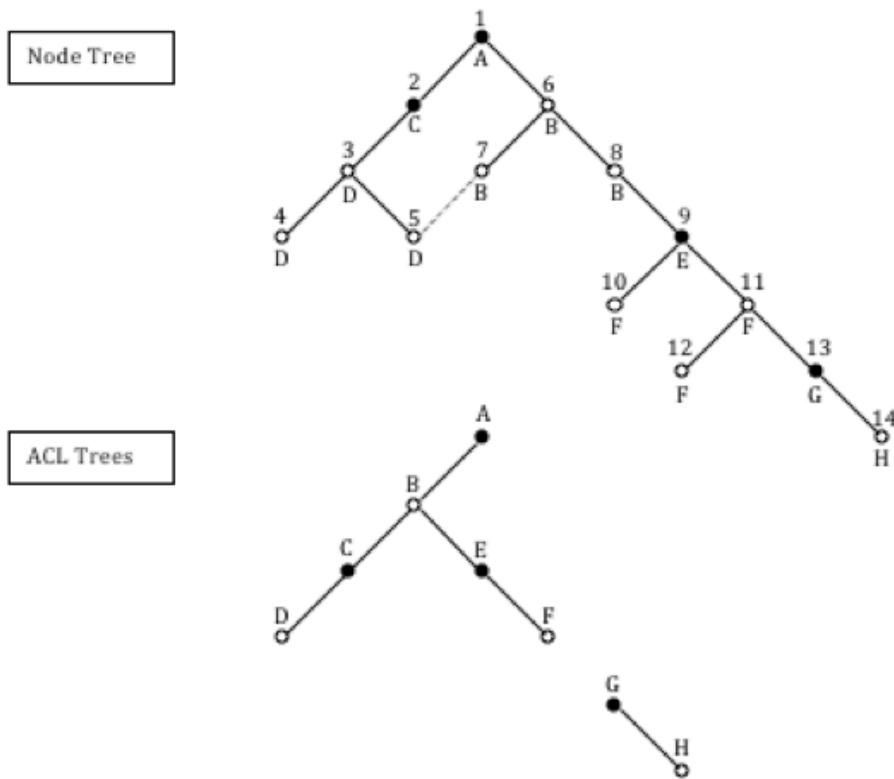
```
security.anyDenyDenies=true
```

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 3: 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 ROLE_OWNER GROUP_A GROUP_A	Read All Write CreateChildren	Allow Allow Allow Allow	2 0 0 0
ACL D (Shared, inherits from ACL C)	ALL ROLE_OWNER GROUP_A GROUP_A	Read All Write CreateChildren	Allow Allow Allow Allow	3 1 1 1
ACL E (Defining, inherits from ACL B)	All Andy Bob Bob	Read All Write WriteContent	Allow Allow Allow Deny	2 0 0 0
ACL F (Shared, inherits from ACL E)	All Andy Bob Bob	Read All Write WriteContent	Allow Allow Allow Deny	3 1 1 1
ACL G (Defining, no inheritance)	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\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\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.Ta
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.

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

Use this information to understand and configure authentication service.

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.

### Configuring multiple tickets for authentication

For each authentication attempt, Alfresco returns a different session ID, but the same ticket for each user. You can configure multiple tickets using the `authentication.ticket.useSingleTicketPerUser` option.

The `TicketComponent` configuration setting, in `alfresco-global.properties`, has an option called `authentication.ticket.useSingleTicketPerUser`. This option has a default setting of `true`, which means that only one ticket is created for each user, and this ticket is returned for every authentication attempt by that user. If the ticket is invalidated, then Alfresco requires the user to re-authenticate before using the repository.

To set multiple tickets for each user, set  
`authentication.ticket.useSingleTicketPerUser=false`.

### Person service

Use this information to understand and configure of person service.

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

Use this information to understand and configure authority service, 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

Use this information 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></pre>

```

4. Save the file and then restart the Alfresco server.

## Permission service

Use this information to understand and configure permission service.

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

Use this information to understand and configure ownable service.

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

The Admin user password 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.

-  Choose a strong, unique password for your admin account, and consider changing it regularly.

### How to reset the Admin password?

If you lose or forget the password for the Admin user, you can reset the password in the database using one of the following methods:

- 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 known user.
  4. Reset the Admin user's password.
  5. Reset the configuration.
- Reset the Admin password without knowing any user password:
  1. Configure the authentication component to accept all logins using org.alfresco.repo.security.authentication.SimpleAcceptOrRejectAllAuthenticationComponent
  2. Login as a user with Admin rights.
  3. Reset the Admin user's password.
  4. Revert the configuration.
- Change the password directly in the database (**for Alfresco version 3.1 to 5.0**):
  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)
```

- To update the password, use the following command:

```
UPDATE alf_node_properties
SET string_value='209c6174da490caeb422f3fa5a7ae634'
WHERE
node_id=THENODEIDABOVE
and
qname_id=THEQNAMEVALUEABOVE
```

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.

- Restart Alfresco.

- Change the password directly in the database (**for Alfresco version 5.1 onwards**):

  - Run the following query to find out which encoder is being used to store the Admin password. Check that you have only one row in the output.

 You must encode the password using the result of the query.

```
SELECT anpl.node_id,
 anpl.qname_id,
 anpl.string_value
FROM alf_node_properties anpl
 INNER JOIN alf_qname aq1 ON aq1.id = anpl.qname_id
 INNER JOIN alf_node_properties anp2 ON anp2.node_id = anpl.node_id
 INNER JOIN alf_qname aq2 ON aq2.id = anp2.qname_id
WHERE aq1.local_name = 'hashIndicator'
AND aq2.local_name = 'username'
AND anp2.string_value = 'admin';
```

The output shows the current password encoding being used.

```
+-----+-----+-----+
| node_id | qname_id | string_value |
+-----+-----+-----+
| 4 | 94 | bcrypt10 |
+-----+-----+
1 row in set (0.01 sec)
```

If no rows are returned, set the password using the instructions shown [above](#) (md4 encoding).

If a row is returned, encode the password using the result of the query, which can either be md4 or sha256 or bcrypt10 encoding.

Run the following query to find the identifying parameters for how the Admin password is stored.

```
SELECT anpl.node_id,
 anpl.qname_id,
 anpl.string_value
FROM alf_node_properties anpl
 INNER JOIN alf_qname aq1 ON aq1.id = anpl.qname_id
 INNER JOIN alf_node_properties anp2 ON anp2.node_id = anpl.node_id
```

```

 INNER JOIN alf_qname aq2 ON aq2.id = anp2.qname_id
WHERE aq1.local_name = 'passwordHash'
AND aq2.local_name = 'username'
AND anp2.string_value = 'admin';

```

The output shows the current hashed password for the Admin user. Here's an example output:

node_id	qname_id	string_value
4	93	\$2a\$10\$dq/2zNUA.MmECYipl1WMoOyGHYbaygh23PUa3Ox5xDHH7Z0guqF42

1 row in set (0.00 sec)

- To update the password, use the following command:

```

UPDATE alf_node_properties
SET string_value='209c6174da490caeb422f3fa5a7ae634'
WHERE
node_id=THENODEIDABOVE
and
qname_id=THEQNAMEVALUEABOVE

```

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 93, respectively.

 Ensure that you use appropriate AND conditions in the UPDATE query.

- Restart Alfresco.

## Security policies and filters

You can configure a number of policies and filters in Alfresco Share to mitigate security attacks.



### Cross-Site Request Forgery (CSRF) and Alfresco

The Open Web Application Security Project (OWASP) describes Cross-Site Request Forgery (CSRF) as a type of attack that occurs when a malicious web site, email, blog, instant message, or program causes a user's web browser to perform an unwanted action on a trusted site for which the user is currently authenticated (see the [Cross-Site\\_Request\\_Forgery\\_Prevention\\_Cheat\\_Sheet](#)).

Currently, the only web-accessible part of the Alfresco product that has CSRF protection is /share. The Share application must be accessible on the network to be available to users, and so it is protected with a CSRF filter.

Other parts of the product, such as /alfresco, do not have CSRF protection. This includes, for example, the Repository Admin Console. This part of the product does not have a CSRF filter because usually the repository would be protected by a firewall.

When setting up a production Alfresco instance, you should ensure that /alfresco is protected behind a firewall.

### Cross-Site Request Forgery (CSRF) filters in Alfresco Share

You can configure `CSRFPolicy` in Alfresco Share to prevent CSRF attacks that allow malicious requests to be unknowingly loaded by a user.

You can configure the CSRF filter to run with third party plugins and to stop specific repository services from being accessible directly through the Share proxy.

The filter is implemented in the `org.alfresco.web.site.servlet.CSRFFilter` that reads the `CSRFPolicy` configuration section in `share-security-config.xml`.

`CSRFPolicy` describes how and when the filter mitigates CSRF attacks:

- Each logged in user receives a secret CSRF token
  - The token is communicated to the browser using a `Alfresco-CSRF-Token` cookie
  - When a logged in user performs a POST, PUT or DELETE HTTP request against Alfresco Share the token must be passed in the request using one of the following methods:
    - As a custom HTTP request header called `Alfresco-CSRF-Token`
    - As a URL parameter called `Alfresco-CSRF-Token`
-  Usually the header is required, but in some circumstances a header cannot be used and in this case the token can be passed using a URL parameter. The default config only accepts the URL parameter when the `Content-Type` header starts with `multipart/`.
- Every time the logged in user visits a new Share page the token is renewed
  - The filter checks that the referrer and original HTTP request headers match the current domain (if this is present in the request)

## Do I need to alter my custom code?

Generally, you should not need to alter your custom code, for example, the following cases need no code alteration:

- You are reading data using GET requests only
- You are using the standard `Alfresco.util.Ajax`, `alfresco/core/CoreXhr` or `Alfresco.forms.Form` javascript classes when creating, updating or deleting data
- You are writing a non-browser client (for example, a mobile application)

However, in these situations you will need to alter your code:

1. You are making an `XMLHttpRequest` with POST, PUT or DELETE methods without using the `Alfresco.util.Ajax` or `alfresco/core/CoreXhr` classes. If you are using the native `XMLHttpRequest` object or a third party library such as jQuery, add code to pass the token, for example:

```
if (Alfresco.util.CSRFPolicy &&
 Alfresco.util.CSRFPolicy.isFilterEnabled())
{
 xhrHeadersObject[Alfresco.util.CSRFPolicy.getHeader()] =
 Alfresco.util.CSRFPolicy.getToken();
}
```

If you are using `YAHOO.util.DataSource` to load data with POST requests, add code similar to this example:

```
if (Alfresco.util.CSRFPolicy &&
 Alfresco.util.CSRFPolicy.isFilterEnabled())
{
 yuiDataSource.connMgr.initHeader(Alfresco.util.CSRFPolicy.getHeader(),
 Alfresco.util.CSRFPolicy.getToken(), false);
}
```

2. You are making a form upload with `enctype multipart/form-data` without using `Alfresco.forms.Form`.

When you upload a file by submitting a form with `enctype multipart/form-data` it is not possible to set a header on the request because it is not possible to set a header on any form submission in the browser. Pass the token as a URL parameter instead. If you are

using the `Alfresco.forms.Form` class, this is handled for you automatically, otherwise add the token as a URL parameter, for example:

```
if (Alfresco.util.CSRFPolicy &&
Alfresco.util.CSRFPolicy.isFilterEnabled())
{
 url += "?" + Alfresco.util.CSRFPolicy.getParameter() + "=" +
encodeURIComponent(Alfresco.util.CSRFPolicy.getToken());
}
```

3. You are using a Flash movie inside Share to send HTTP requests with method POST.

If you are using a Flash movie to upload files, using the `flash.net.FileReference` ActionScript class to perform a multipart/form-data request, add the token as a URL parameter in your Javascript before passing in the URL to the Flash movie. If your Flash movie is performing application/json or other text based POST requests, using the `flash.net.URLRequest` and/or `flash.net.navigateToURL` ActionScript classes and methods, pass the token and the name of the header so that it can be set from the Flash movie.

## When else might I need to make code updates?

If servers from other domains are allowed to POST requests to your system, then you need to reconfigure `CSRFPolicy` in your `share-config-custom.xml` file so that the token or header is not checked:

1. Copy the `CSRFPolicy` configuration in `share-security-config.xml`.
2. Paste the configuration into your `share-config-custom.xml` file, ensuring that it is replacing the old configuration section:

```
<config evaluator="string-compare"
 condition="CSRFPolicy" replace="true">
```

3. Copy the following code and add it as the first child of the `<filter>` element:

```
<rule>
 <request>
 <method>POST</method>
 <path>/page/trusted/call/1|/page/trusted/call/2</path>
 </request>
 <action name="assertReferer">
 <param name="always">false</param>
 <param name="referer">https://www.trustedserver.com/.*</param>
 </action>
 <action name="assertOrigin">
 <param name="always">false</param>
 <param name="origin">https://www.trustedserver.com</param>
 </action>
</rule>
```

The CSRF filter compares the incoming request with the rule request elements to find one that matches and then invokes the defined actions for that rule before normal Share processing begins.

If you want to completely block certain services in the repository, you can add these URLs to the CSRF filter:

1. Copy the `CSRFPolicy` configuration in `share-security-config.xml`.
2. Paste the configuration into your `share-config-custom.xml` file, ensuring that it is replacing the old configuration section:

```
<config evaluator="string-compare"
 condition="CSRFPolicy" replace="true">
```

3. Copy the following code and add it as the first child of the `<filter>` element:

```

<rule>
 <request>
 <path>/proxy/alfresco/acme/special/services/.*</path>
 </request>
 <action name="throwError">
 <param name="message">It is not allowed to access this url from
 your browser</param>
 </action>
</rule>

```

## Iframes and phishing attack mitigation in Alfresco Share

You can configure `IFramePolicy` to protect users against a phishing attack, which attempts to acquire information such as user names or passwords by simulating a trustworthy entity.

Alfresco allows you to control which domain pages or content are included in Share to create a whitelist of allowed domains. A whitelist is a list of email addresses or IP addresses that are considered to be safe for use within your organisation.

This `IFramePolicy` is applied when Share includes an `<iframe>` tag while constructing the Web View dashlet. The dashlet will allow only those URLs that have been added to the whitelist. Developers can use the `Alfresco.util.IFramePolicy.isUrlAllowed()` method to check if a URL is allowed for custom implementations of a Web View or `<iframe>` tag is included.

-  If you have a previous installation which includes a URL from a third-party domain, you will get an error message in your production environment prompting you to configure your `IFramePolicy` configuration by adding the domain to the whitelist.
-  URLs pointing to the same domain, such as documents or wiki pages inside Share, will continue to work as usual by default.

The whitelist of allowed domains is set in the `<configRootShare>/classes/alfresco/share-security-config.xml` configuration file:

```

<config evaluator="string-compare" condition="IFramePolicy">
 <same-domain>allow</same-domain>
 <cross-domain>
 <url>*</url>
 </cross-domain>
</config>

```

To deny URLs from the current domain, override the existing code in the `share-config-custom.xml` file with the following code:

```

<config evaluator="string-compare" condition="IFramePolicy" replace="true">
 <same-domain>deny</same-domain>
</config>

```

To allow all cross domain URLs, override the existing code in the `share-config-custom.xml` file with the following code:

```

<config evaluator="string-compare" condition="IFramePolicy" replace="true">
 <cross-domain>
 <url>*</url>
 </cross-domain>
</config>

```

To allow specific cross domain URLs, override the existing code in the `share-config-custom.xml` file with the following code:

```

<config evaluator="string-compare" condition="IFramePolicy" replace="true">
 <cross-domain>
 <url>https://www.owasp.org/</url>
 </cross-domain>
</config>

```

## Security filters and clickjacking mitigation in Alfresco Share

You can configure a security filter, `SecurityHeadersPolicy`, that mitigates clickjacking attacks in Alfresco Share.

`SecurityHeadersPolicy` is a Java Servlet filter that applies HTTP response headers to incoming requests in Alfresco Share. The headers that are returned are defined in a configuration section called `SecurityHeadersPolicy` in `alfresco-security-config.xml`.

Three headers are added by default; `X-Frame-Options`, `X-Content-Type-Options` and `X-XSS-Protection`:

```
<config evaluator="string-compare" condition="SecurityHeadersPolicy">
 <headers>
 <header>
 <name>X-Frame-Options</name>
 <value>SAMEORIGIN</value>
 </header>
 <header>
 <name>X-Content-Type-Options</name>
 <value>nosniff</value>
 </header>
 <header>
 <name>X-XSS-Protection</name>
 <value>1; mode=block</value>
 </header>
 </headers>
</config>
```

### X-Frame-Options header

Adding this header to an HTTP response tells the browser whether Share pages are permitted inside iframes. In our default configuration we have set this to `SAMEORIGIN` which means that Share pages are only permitted inside iFrames inside Share or in other web applications that live under the same domain. For example, it is possible to include `http://www.acme.com/share` inside an iframe on `http://www.acme.com/portal`.

You can override the configuration and set the header to return `DENY` instead, by placing the following configuration in your `share-config-custom.xml` file:

```
<config evaluator="string-compare" condition="SecurityHeadersPolicy">
 <headers>
 <header>
 <name>X-Frame-Options</name>
 <value>DENY</value>
 </header>
 </headers>
</config>
```

### X-Content-Type-Options

This header is valid for Internet Explorer (IE) only. Older versions of IE (8 and below) sniff the content of a returned resource and then execute the content as the content type that IE thinks the resource has, instead of the content type that the server returned. To stop IE from doing this, `nosniff` is returned in the header.

### X-XSS-Protection

This header is provided by Internet Explorer (IE) to rectify “sanitization” logic that can be used by an attacker to introduce an XSS flaw on your site.

By default Alfresco Share returns `1; mode=block` for this header, which stops IE from executing sanitized code.

It is also possible to set the value to `0` which stops IE from inspecting the code for XSS attacks.

## Adding other headers

Alfresco supports adding other headers to the configuration, for example, the `Strict-Transport-Security` header forces the browser to allow only `https` communication. This header is not provided by Alfresco Share, but can be added by using this code:

```
<config evaluator="string-compare" condition="SecurityHeadersPolicy">
 <headers>
 <header>
 <name>Strict-Transport-Security</name>
 <value>max-age=31536000</value>
 </header>
 </headers>
</config>
```

## Configuring search

From Alfresco One 5.0 onwards, Solr 4 is the default search subsystem. Use this information for an overview on the Solr 4 search service and how to configure it.

-  In this information, we refer to **Solr 4** search subsystem as **Solr**.
-  The Lucene search subsystem is not supported in Alfresco One 5.1.

## 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

Use this information to 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" replace="true">` 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

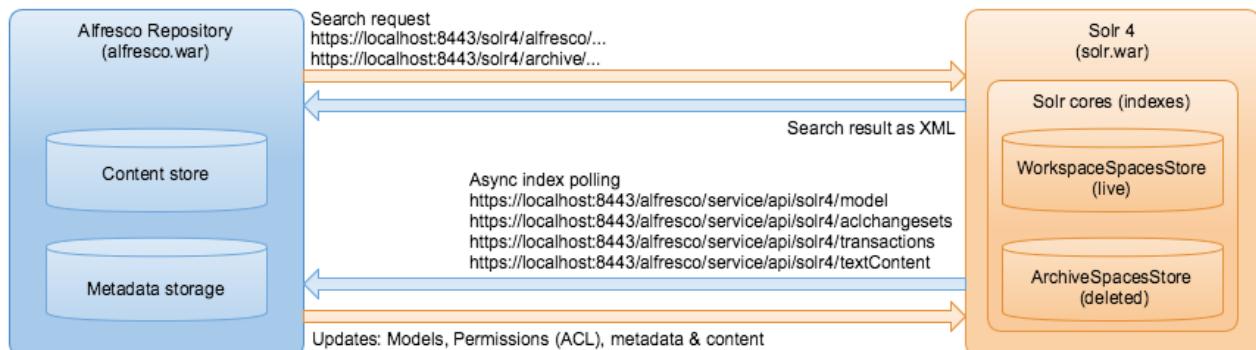
Alfresco supports use of the Solr search platform for searching within the Alfresco repository.

Solr is an open source enterprise search platform that uses lucene as indexing and search engine. Solr is written in Java and runs as a standalone search 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.

 To confirm which application servers Alfresco supports for running the Solr application on, see the [Supported Platforms](#) matrix for your version of Alfresco One.

There are two cores or indexes in Solr:

- **WorkspaceStore**: used for searching all live content stored at `alfresco/solr4` within the Solr search server.
- **ArchiveStore**: used for searching content that has been marked as deleted at `alfresco/solr4` within the Solr 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.
-  Every Alfresco installation is supplied with a generic certificate and SSL keys. For security reasons, it is advised that you generate a new set of keys to secure your Solr

communication and access to the Solr Admin Console. For more information, see [Configuring Solr](#) and [Generating secure keys for Solr communication](#).

## 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 5.1 introduces the concept of eventual consistency to overcome the scalability limitations of in-transaction indexing.

Alfresco One 5.1 with the Solr 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 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 search service

The way that you configure Alfresco to use Solr 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.

Use this information to configure the Solr search subsystem, and to understand the Solr directory structure, configuration files, and properties.

### Installing and configuring Solr

The Solr 4 search subsystem is installed by default when you install Alfresco One 5.1 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.

 To confirm which application servers Alfresco supports for running the Solr application on, see the [Supported Platforms](#) matrix for your version of Alfresco One.

The installation contains the following artifacts:

- a template SOLR home directory containing `solr.xml`, which is expected by Solr
- Solr 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 either to the same Tomcat application server as Alfresco or a separate Tomcat. The Solr 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 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 server. For security reasons, the communication channel between the Alfresco server and Solr 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 is installed. These can be the same or different directories, depending on whether you have chosen to install Solr on a standalone server.

1. Extract the alfresco-one-distribution-201603.zip file to a location. For example, <EXTRACTED-ARCHIVE>.
2. The <EXTRACTED-ARCHIVE> directory contains a solr4 directory. Copy the solr4 folder to the <ALFRESCO\_HOME> directory, for example, <ALFRESCO\_HOME>/solr4/. This directory now becomes <SOLR4-ARCHIVE>, which is the Solr base directory.
3. Copy the <ALFRESCO\_HOME>/solr4/context.xml file to <SOLR4\_TOMCAT\_HOME>\conf\Catalina\localhost\solr4.xml.
4. Edit solr/home in XML to point to the path for <SOLR4-ARCHIVE>, which is the Solr base directory mentioned in Step 2.

For example:

```
<?xml version="1.0" encoding="utf-8"?>
<Context debug="0" crossContext="true">
 <Environment name="solr/home" type="java.lang.String" value="" override="true"/>
 <Environment name="solr/model/dir" type="java.lang.String" value="@@ALFRESCO_SOLR4_MODEL_DIR@@" override="true"/>
 <Environment name="solr/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 model directory. For example, <ALFRESCO\_HOME>/alf\_data/solr4/model.
  - @@ALFRESCO\_SOLR4\_CONTENT\_DIR@@ should point to the location of the Solr 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 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 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 Communication](#) section.

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 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 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 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 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 Communication](#) section.

For example:

```
<user username="CN=Alfresco Repository Client, OU=Unknown, O=Alfresco
Software Ltd., L=Maidenhead, ST=UK, C=GB" roles="repoclient"
password="null"/>
```

12. To complete the installation, it is necessary to secure the two-way communication between Alfresco and Solr by generating your own keys. See the [Generating Secure Keys for Solr Communication](#) topic.

### Generating secure keys for Solr communication

This task describes how to replace or update the keys used to secure communication between Alfresco and Solr, using secure keys specific to your Alfresco installation.

The following instructions assume that Solr 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](#) 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.
2. If you are updating an environment created by the Alfresco installer, you only need to edit `ALFRESCO_HOME` and `SOLR_HOME` to specify the correct installation directory.
  - 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.

By default, for Solr `SOLR_HOME` refers to `<ALFRESCO_HOME>/solr4`.

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 directory structure

After you have installed Alfresco, several directories and configuration files related to Solr will be available in the Alfresco home directory.

`alfresco\solr4`

This is the Solr 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 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 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 web application context template to use when installing Solr in separate tomcat server.
- log4j-solr.properties: This is the configuration file for Solr-specific logging.
- solr.xml: This configuration file specifies the cores to be used by Solr.

**alfresco\alf\_data\solr4\**

The Solr directory contains the following sub-folders:

- content: This directory contains a compressed copy of all the Solr documents added to the index. The content directory:
  - Does not need to be backed up.
  - Works more efficiently on fast and local drives.
  - Size cannot be more than 20% of the contentstore size, but this varies considerably.
  - Besides being used for reindexing and future intended use for highlighting, the content directory saves transformations. If only the metadata is updated on a node, the cached content can be used to get the previous transformation results. If the content is updated on a node, it can be indexed with the new metadata and the old transformed content, until the new transformed content is available.
- index: This directory contains all the indexes of the archive and workspace cores.
- model: This directory contains all the models.

**alfresco\alf\_data\solr4Backup\**

This directory stores the Solr backup. It contains the alfresco and archive sub-directories.

### Solr configuration files

When you install Alfresco One 5.1, several Solr configuration files are made available to you. The section lists the Solr configuration files, their location in the Alfresco directory structure and description.

Configuration File	Location	Description
schema.xml	alfresco \solr4\<core>\conf \, where <core> 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.

Configuration File	Location	Description
solr4.xml	alfresco\tomcat\conf\catalina\localhost\	This file defines the Solr web application context. It specifies the location of the Solr war file and sets up the Solr home directory.
solr.xml	alfresco\alf_data\solr4	This file specifies the cores to be used by Solr.
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.
solrconfig.xml	alfresco\solr4\workspace-SpacesStore\conf or alfresco\solr4\archive-SpacesStore\conf	This file specifies the parameters for configuring Solr. Also, the Solr 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 supports system property substitution, so properties that need substitution can be put in to this file. There is one solrcore.properties file in each core's configuration directory. For details, see the <a href="#">Solr core configuration properties</a> topic. The properties of this file are managed by an Alfresco Administrator user.
context.xml	alfresco\solr4	This file specifies the Solr web application context template to use when installing Solr in separate tomcat server.
ssl.repo.client.keystore	alfresco\solr4\workspace-SpacesStore\conf or alfresco\solr4\archive-SpacesStore\conf	This keystore contains the Solr 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 ssl.repo.client.keystore.
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 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 core configuration properties

The `solrcore.properties` configuration file is the property configuration file for a Solr core. There is one `solrcore.properties` file in each core's configuration directory. Use this information to understand the properties of this file, their description, and the default value.

Property Name	Description	Default Value
<code>data.dir.root</code>	This property specifies the top level directory path for the indexes managed by Solr.	C:/Alfresco/alf_data/solr4/index
<code>data.dir.store</code>	This property specifies the directory relative to <code>data.dir.root</code> where the data for this core is stored.	workspace/SpacesStore
<code>enable.alfresco.tracking</code>	This property instructs Solr if it should index Alfresco content in the associated Alfresco repository store or not.	true
<code>max.field.length</code>	This property specifies the maximum number of tokens to include for each field. By default, all tokens are added.	2147483647
<code>alfresco.version</code>	This property specifies the Alfresco version installed.	5.1
<code>alfresco.host</code>	This property specifies the host name for the Alfresco instance that Solr should track and index. In a default installation, both Alfresco and Solr runs in the same Tomcat instance and on the same host, so host would be set to local host.	localhost
<code>alfresco.port</code>	This property specifies the HTTP port for the Alfresco instance that Solr should track and index.	8080
<code>alfresco.port.ssl</code>	This property specifies the HTTPS port for the Alfresco instance that Solr should track and index.	8443
<code>alfresco.cron</code>	This property specifies the cron expression that instructs Solr how often to track Alfresco and index new or updated content. The default value indicates that Solr tracks Alfresco every 15 seconds.	0/15 * * * ? *
<code>alfresco.stores</code>	This property specifies the Alfresco repository store that this core should index.	workspace://SpacesStore

Property Name	Description	Default Value
alfresco.baseUrl	This property configures the base URL to Alfresco web project. If you need to change the baseUrl value, see <a href="#">Deploying Alfresco with a different context path</a> on page 335 for configuring information.	/alfresco
alfresco.lag	When Solr tracking starts, it aims to get up to date to the current time (in seconds), less this lag.	1000
alfresco.hole.retention	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.	3600000
alfresco.batch.count	This property indicates the number of updates that should be made to this core before a commit is executed.	1000
alfresco.secureComms	This property instructs Solr if it should talk to Alfresco over HTTP or HTTPS. Set to none if a plain HTTP connection should be used.	https
alfresco.encryption.ssl.key.type	<i>This property specifies the CLIENT keystore type.</i>	JCEKS
alfresco.encryption.ssl.key.provider	<i>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 keystore type specified is used.</i>	
alfresco.encryption.ssl.key.location	<i>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.</i>	ssl.repo.client.keystore
alfresco.encryption.ssl.key.location.password	<i>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.</i>	ssl-keystore-passwords.properties
alfresco.encryption.ssl.truststore.type	<i>This property specifies the CLIENT truststore type.</i>	JCEKS

Property Name	Description	Default Value
alfresco.encryption.ssl.truststoreProvider	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.	
alfresco.encryption.ssl.truststoreLocation	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.	ssl.repo.client.truststore
alfresco.encryption.ssl.truststoreFileLocation	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.	ssltruststore-passwords.properties
alfresco.corePoolSize	This property specifies the pool size for multi-threaded tracking. It is used for indexing nodes.	3
alfresco.maximumPoolSize	This property specifies the maximum pool size for multi-threaded tracking.	-1
alfresco.keepAliveTime	This property specifies the time (in seconds) to keep non-core idle threads in the pool.	120
alfresco.threadPriority	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
alfresco.threadDaemon	This property sets whether the threads run as daemon threads or not. If set to false, shut down is blocked else it is left unblocked.	true
alfresco.workQueueSize	This property specifies the maximum number of queued work instances to keep before blocking against further adds.	-1
alfresco.maxTotalConnections	This property is used for HTTP client configuration.	40
alfresco.maxHostConnections	This property is used for HTTP client configuration.	40

<b>Property Name</b>	<b>Description</b>	<b>Default Value</b>
alfresco.socketTimeout	This property specifies the amount of time Solr tracker will take to notice if the Alfresco web app shuts down first, if Alfresco and Solr are running on the same web application.	60000
solr.filterCache.size	This property specifies the maximum number of entries in the Solr filter cache.	64
solr.filterCache.initialSize	This property specifies the initial capacity (number of entries) of the Solr filter cache.	64
solr.queryResultCache.size	This property configures the Solr result cache.	1024
solr.queryResultCache.initialSize	This property configures the Solr result cache.	1024
solr.documentCache.size	This property configures the Solr document cache.	64
solr.documentCache.initialSize	This property configures the Solr document cache.	64
solr.queryResultMaxDocsCache	This property configures the Solr result cache.	2000
solr.authorityCache.size	This property configures the Solr result cache.	64
solr.authorityCache.initialSize	This property configures the Solr result cache.	64
solr.pathCache.size	This property configures the Solr result cache.	64
solr.pathCache.initialSize	This property configures the Solr result cache.	64
solr.ownerCache.size	This property configures the Solr result cache.	4096
solr.ownerCache.initialSize	This property configures the Solr result cache.	1024
solr.readerCache.size	This property configures the Solr result cache.	4096
solr.readerCache.initialSize	This property configures the Solr result cache.	1024
solr.deniedCache.size	This property configures the Solr result cache.	4096
solr.deniedCache.initialSize	This property configures the Solr result cache.	1024
solr.nodeBatchSize	This property configures the Solr result cache.	10
solr.filterCache.autowarmCount	This property configures the Solr result cache.	128
solr.authorityCache.autowarmCount	This property configures the Solr result cache.	0

Property Name	Description	Default Value
solr.pathCache.autowarmCount	This property configures the Solr result cache.	128
solr.deniedCache.autowarmCount	This property configures the Solr result cache.	0
solr.readerCache.autowarmCount	This property configures the Solr result cache.	0
solr.ownerCache.autowarmCount	This property configures the Solr result cache.	0
solr.queryResultCache.autowarmCount	This property configures the Solr result cache.	0
solr.documentCache.autowarmCount	This property configures the Solr result cache.	0
solr.queryResultWindowSize	This property configures the Solr result cache.	200
alfresco.doPermissionCheck	This property configures the Solr result cache.	true
alfresco.metadata.skipDescriptorType	This property configures the Solr result cache. Type: false	false
alfresco.metadata.ignoreContent	This property configures the Solr result cache.	cm:person
alfresco.metadata.ignoreContentTypes	This property configures the Solr result cache.	app:configurations
solr.maxBooleanClauses	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
alfresco.transactionDocsBatchSize	This property is used for batch fetching updates during tracking.	100
alfresco.changeSetAclsBatchSize	This property is used for batch fetching updates during tracking.	100
alfresco.aclBatchSize	This property is used for batch fetching updates during tracking.	10
alfresco.index.transformContent	If this property is set to false, the index tracker will not transform any content and only the metadata will be indexed.	false

## Solr subsystem

Search is contained in a subsystem and it has an implementation of Solr.

The following properties in the `alfresco-global.properties` file are related to Solr and are setup as follows, by default:

```
Solr indexing
index.subsystem.name=solr4
dir.keystore=${dir.root}/keystore
solr.port.ssl=8443
```

## Activating Solr

Use this information to activate the Solr 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
index.subsystem.name	The subsystem type value. The default value is solr4.
solr.host	The host name where the Solr instance is located.
solr.port	The port number on which the Solr instance is running.
solr.port.ssl	The port number on which the Solr SSL support is running.

For example, some example properties for activating Solr 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.

### Admin Console

1. Open the Admin Console.
2. Edit the following properties:

Property	Description
index.subsystem.name	Select the subsystem type value as solr4.
solr.host	The host name where the Solr instance is located.
solr.port	The port number on which the Solr instance is running.
solr.port.ssl	The port number on which the Solr SSL support is running.

3. Click Save.

For more information, see [Working with the Search Service](#) on page 277.

### 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.

### Working with the Search Service

**Search Service** in the Admin Console helps 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.

-  The Solr 1 option should be used only for migration to Solr 4.

### *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.  
 The Solr 1 option should be used only for migration to Solr 4.
4. Set the Solr 4 search service properties:

Solr search property	Example setting	What is it?
<b>Content Tracking Enabled</b>	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.
<b>Solr Port (Non-SSL)</b>	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.
<b>Solr base URL</b>	/solr4	This specifies the base URL for the Solr 4 web application.
<b>Solr Hostname</b>	localhost	This specifies the hostname on which the Solr 4 server is running. Use localhost if running on the same machine.
<b>Solr SSL Port</b>	8443	This specifies the application server's https port on which Solr 4 is running.
<b>Auto Suggest Enabled</b>	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.
<b>Indexing in Progress</b>	No	This specifies if Solr 4 is currently indexing outstanding transactions.
<b>Last Indexed Transaction</b>	17	This specifies the transaction ID most recently indexed by Solr 4.
<b>Approx Index Time Remaining</b>	0 Seconds	This specifies the estimated time that Solr 4 will take to complete indexing the current outstanding transactions.

Solr search property	Example setting	What is it?
<b>Disk Usage (GB)</b>	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.
<b>Index Lag</b>	0 s	This specifies the time that indexing is currently behind the repository updates.
<b>Approx Transactions to Index</b>	0	This specifies the estimated number of outstanding transactions that require indexing.
<b>Memory Usage (GB)</b>	0	This specifies the current memory usage. The value may vary due to transient memory used by background processing.
<b>Indexing in Progress</b>	No	This specifies if Solr 4 is currently indexing outstanding transactions.
<b>Last Indexed Transaction</b>	17	This specifies the transaction ID most recently indexed by Solr 4.
<b>Approx Index Time Remaining</b>	0 Seconds	This specifies the estimated time that Solr 4 will take to complete indexing the current outstanding transactions.
<b>Disk Usage (GB)</b>	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.
<b>Index Lag</b>	0 s	This specifies the time that indexing is currently behind the repository updates.
<b>Approx Transactions to Index</b>	0	This specifies the estimated number of outstanding transactions that require indexing.
<b>Memory Usage (GB)</b>	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.
<b>Backup Location (Main Store)</b>	<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.

Solr search property	Example setting	What is it?
<b>Backup Cron Expression</b> (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.
<b>Backups To Keep</b> (Main Store)	3	This specifies the number of backups to keep (including the latest backup).
<b>Backup Location</b> (Archive Store properties)	\${dir.root}/solr4Backup/archive	This specifies the location where the index backup for ArchiveStore is stored on the Solr 4 server.
<b>Backup Cron Expression</b> (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.
<b>Backups To Keep</b> (Archive Store properties)	3	This specifies the number of backups to keep (including the latest backup).
<b>CMIS Query</b>	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.
<b>Alfresco Full Text Search</b>	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.

5. 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

If you use the **No Index** service, you only have the [transactional metadata query](#) functionality available until you build your Solr 4 indexes.

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 **No Index** from the **Search Service In Use** list.
4. Set the No Index search service properties:

No Index service property	Example setting	What is it?
<b>Content Tracking Enabled</b>	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.
<b>CMIS Query</b>	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.
<b>Alfresco Full Text Search</b>	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.

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

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

## Solr security

By default, communication between Alfresco repository and Solr is protected by SSL with mutual authentication. Both the repository and Solr have their own standard public/private key pair. To secure the two-way communication between the repository and Solr, you must generate your own keys.

-  Every Alfresco installation is supplied with a generic certificate and SSL keys. For security reasons, it is advised that you generate a new set of keys to secure your Solr communication and access to the Solr Admin Console. For more information, see [Configuring Solr](#) and [Generating secure keys for Solr communication](#).

### Repository SSL keystores

Use this information to understand the keystores used by the repository for SSL.

The repository has two keystores 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 certificates)

These keystores can be stored in any location.

Update the following keystore 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 keystore location.
<code>encryption.ssl.keystore.provider</code>	Specifies the keystore provider.
<code>encryption.ssl.keystore.type</code>	Specifies the keystore type.
<code>encryption.ssl.keystore.keyMetaData.location</code>	Specifies the keystore metadata file location.

`ssl truststore`

Property	Description
encryption.ssl.truststore.location	Specifies the trust store location.
encryption.ssl.truststore.provider	Specifies the trust store provider.
encryption.ssl.truststore.type	Specifies the trust store type.
encryption.ssl.truststore.keyMetaData.location	Specifies the trust store metadata file location.

## Solr SSL keystores

Solr core has two keystores that it uses for SSL.

These are:

- `ssl.repo.client.keystore` contains a Solr 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 certificates)

## Connecting to the SSL-protected Solr web application

The Solr Admin Web interface allows you to view Solr configuration details, run queries, and analyze document fields.

All Solr URLs, which are bundled within Alfresco, are protected by SSL. To use these URLs from a browser, you need to import a browser-compatible keystore to allow mutual authentication and decryption to work. The following steps describe how to import the keystore 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 keystore has been imported successfully. The **Certificate Manager** now contains the imported keystore with the Alfresco repository certificate under the **Your Certificates** tab.

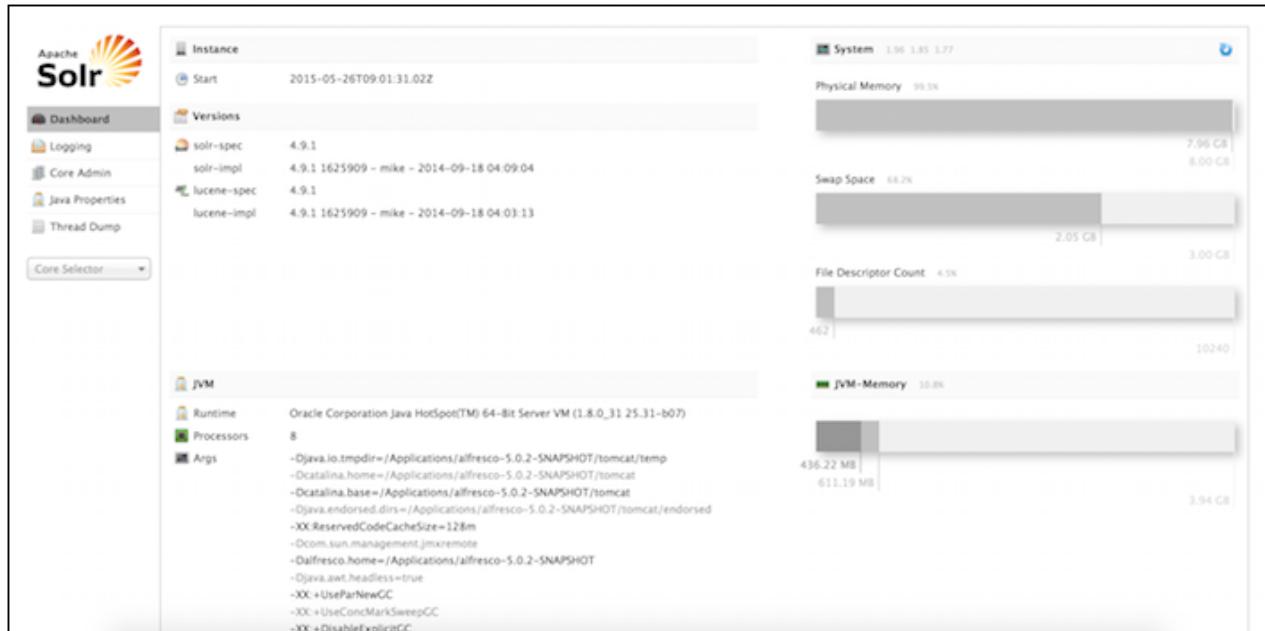
4. Close the **Certificate Manager** by clicking **OK**.
5. In the browser, navigate to a Solr URL, <https://localhost:8443/solr4>.  
The browser displays an error message window to indicate that the connection is untrusted. This is due to the Alfresco certificate not being tied to the server IP address. 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 is granted and the Solr Admin screen is displayed.

The Solr web interface makes it easy for administrators to view the Solr configuration details, run queries, and analyse document fields in order to calibrate a Solr configuration.

The main Solr Admin dashboard is divided into two parts.

Click on the left or the center of the Solr Admin UI below to learn more about it.



1. [Solr Admin UI - left panel](#) on page 283
2. [Solr Admin UI - center panel](#) on page 286

### Solr Admin UI - left panel

The left-side of the Solr Admin screen is a menu under the Solr logo that provides the navigation through the screens of the UI. The first set of links are for system-level information and configuration, and provide access to Logging, Core Admin and Java Properties, among other things.

After this information is a list of Solr cores configured for your Alfresco instance. Clicking on a core name shows a secondary menu of information and configuration options for that core specifically. Items in this list include the Schema, Config, Plugins, and an ability to perform queries on indexed data.

The different screens of the Solr Admin UI are described below:

### Logging

The **Logging** page shows messages from Solr's log files.

Under **Logging**, when you select **Level**, you see the hierarchy of classpaths andclassnames for your Level instance. A row highlighted in yellow indicates that the class has logging capabilities. Click on a highlighted row, and a menu will appear to allow you to change the log level for that class. Characters in bold indicate that the class will not be affected by level changes to root.

## Configuring

The screenshot shows the Apache Solr Logging interface. On the left is a navigation sidebar with links: Dashboard, Logging (selected), Level, Core Admin, Java Properties, and Thread Dump. Below this is a "Core Selector" dropdown. The main area is titled "Log4j (org.slf4j.impl.Log4jLoggerFactory)". It displays a hierarchical tree of log entries. A context menu is open over the entry "missingProperties" under the "alfresco" category, showing a list of log levels: ALL, TRACE, DEBUG, INFO, WARN, ERROR, FATAL, OFF, and UNSET. The "connection" entry under "sf" is highlighted in red. The log entries are color-coded by level: green for INFO/WARN, yellow for DEBUG, orange for ERROR/FATAL, and red for ALL/TRACE/OFF/UNSET.

## Core Admin

The **Core Admin** screen lets you manage your cores.

The buttons at the top of the screen enables you to add a new core, unload the core displayed, rename the currently displayed core, swap the existing core with one that you specify in a drop-down box, reload the current core, and optimize the current core.

The main display and available actions provide another way of working with your cores.

The screenshot shows the Apache Solr Core Admin interface. On the left is a navigation sidebar with links: Dashboard, Logging, Core Admin (selected), Java Properties, and Thread Dump. Below this is a "Core Selector" dropdown. The main area shows a table for the "alfresco" core. At the top are buttons: Add Core, Unload (red), Rename, Swap, Reload, and Optimize. The table has two sections: "Core" and "Index". The "Core" section contains: startTime (about 23 hours ago), instanceDir (/Applications/alfresco-5.0.2-SNAPSHOT/solr4/workspace-SpacesStore/), and dataDir (/Applications/alfresco-5.0.2-SNAPSHOT/alf\_data/solr4/index/workspace/SpacesStore/). The "Index" section contains: lastModified (about 23 hours ago), version (52), numDocs (911), maxDoc (914), deletedDocs (3), optimized (red circle with a question mark), current (green checkmark), and directory (org.apache.lucene.store.NRTCachingDirectory@/Applications/alfresco-5.0.2-SNAPSHOT/alf\_data/solr4/index/workspace/SpacesStore/index; lockFactory=NativeFSLockFactory@/Applications/alfresco-5.0.2-SNAPSHOT/alf\_data/solr4/index/workspace/SpacesStore/index; maxCacheMB=48.0 maxMergeSizeMB=4.0).

## Java Properties

The **Java Properties** screen displays all the properties of the JVM running Solr, including the classpaths, file encodings, JVM memory settings, operating system, and more.



alfresco.home	/Applications/alfresco-5.0.2-SNAPSHOT
alfresco.jmx.dir	/Applications/alfresco-5.0.2-SNAPSHOT/tomcat/webapps/alfresco/WEB-INF/classes/alfresco
awt.toolkit	sun.awt.macosx.LWCToolkit
catalina.base	/Applications/alfresco-5.0.2-SNAPSHOT/tomcat
catalina.home	/Applications/alfresco-5.0.2-SNAPSHOT/tomcat
catalina.useNaming	true
com.sun.management.jmxremote	
common.loader	\$[catalina.base]/lib,\$[catalina.base]/lib/*.jar,\$[catalina.home]/lib,\$[catalina.home]/lib/*.jar
file.encoding	UTF-8
file.encoding.pkg	sun.io
file.separator	/
ftp.nonProxyHosts	local *.local 169.254/16 *.169.254/16
gopherProxySet	false
hibernate.dialect	org.hibernate.dialect.PostgreSQLDialect
http.nonProxyHosts	local *.local 169.254/16 *.169.254/16
java.awt.graphicsenv	sun.awt.CGraphicsEnvironment
java.awt.headless	true
java.awt.printerjob	sun.awt.macosx.CPrinterJob
java.class.path	/Applications/alfresco-5.0.2-SNAPSHOT/tomcat/bin/bootstrap.jar /Applications/alfresco-5.0.2-SNAPSHOT/tomcat/bin/tomcat-juli.jar
java.class.version	52.0
java.endorsed.dirs	/Applications/alfresco-5.0.2-SNAPSHOT/tomcat/endorsed
java.ext.dirs	/Users/mgaure/Library/java/Extensions /Applications/alfresco-5.0.2-SNAPSHOT/java/lib/ext /Library/Java/Extensions /Network/Library/java/Extensions /System/Library/java/Extensions /usr/lib/java

## Thread Dump

The **Thread Dump** screen lets you inspect the currently active threads on your server.

Each thread is listed and access to the stacktraces is available where applicable. Icons to the left indicate the state of the thread. For example, threads with a green check-mark in a green circle are in a `RUNNABLE` state.

On the right of the thread name, click the down-arrow to see the stacktrace for that thread.



name	cpuTime / userTime
RMI TCP Connection(idle) (2/69)	0.8860ms 0.5460ms
java.util.concurrent.SynchronousQueue\$TransferStack@22516d50	
http-bio-8443-exec-20 (261)	888.5960ms 726.1400ms
java.util.concurrent.locks.AbstractQueuedSynchronizer\$ConditionObject@2163e7b4	
http-bio-8443-exec-21 (260)	897.1220ms 739.4760ms
http-bio-8443-exec-19 (259)	1024.3470ms 843.6570ms
java.util.concurrent.locks.AbstractQueuedSynchronizer\$ConditionObject@2163e7b4	
http-bio-8443-exec-18 (258)	1089.7150ms 903.7640ms
java.util.concurrent.locks.AbstractQueuedSynchronizer\$ConditionObject@2163e7b4	
http-bio-8443-exec-16 (238)	3086.7490ms 2545.9390ms
java.util.concurrent.locks.AbstractQueuedSynchronizer\$ConditionObject@2163e7b4	
http-bio-8080-exec-4 (180)	0.6010ms 0.3840ms
java.util.concurrent.locks.AbstractQueuedSynchronizer\$ConditionObject@5f03bfdf	
http-bio-8080-exec-3 (179)	0.9830ms 0.7510ms
java.util.concurrent.locks.AbstractQueuedSynchronizer\$ConditionObject@5f03bfdf	
http-bio-8080-exec-2 (178)	1.7560ms 1.4010ms
java.util.concurrent.locks.AbstractQueuedSynchronizer\$ConditionObject@5f03bfdf	
http-bio-8080-exec-1 (177)	4.7000ms 3.2550ms
java.util.concurrent.locks.AbstractQueuedSynchronizer\$ConditionObject@5f03bfdf	
JoblockService1 (148)	79.6230ms 58.7210ms
java.util.concurrent.locks.AbstractQueuedSynchronizer\$ConditionObject@41ebd302	
MultiThreadedHttpConnectionManager cleanup (146)	0.1180ms 0.0720ms
java.lang.ref.ReferenceQueue\$Lock@667c2d32	
http-bio-8443-exec-13 (145)	11296.9370ms

## Core-Specific Tools

Click the **Core Selector** to display a list of Solr cores, with a search box that can be used to find a specific core.

When you select a core:

- the central part of the screen shows Statistics and other information about the selected core.

- a secondary menu opens under the core name with the administration options available for that particular core. The core-specific options are:

Options	Description
Analysis	Allows data analysis according to the field, field type and dynamic rule configurations found in schema.xml.
Dataimport	Displays information about the current status of the Data Import Handler. It enables you to import commands as defined by the options selected on the screen and defined in the configuration file.
Documents	Provides a simple form allowing execution of various Solr indexing commands directly from the browser. The screen allows you to: <ul style="list-style-type: none"> <li>• Copy documents in JSON, CSV or XML and submit them to the index</li> <li>• Upload documents (in JSON, CSV or XML)</li> <li>• Construct documents by selecting fields and field values</li> </ul>
Files	Displays the current core configuration files such as solrconfig.xml and schema.xml. Configuration files cannot be edited with this screen, so a text editor must be used.
Ping	Enables you to ping a named core and determine whether the core is active. The Ping option does not open a page, but the status of the request can be seen on the core overview page shown when clicking on a collection name. The length of time the request has taken is displayed next to the Ping option, in milliseconds.
Plugins/Stats	Displays statistics for plugins and other installed components.
Query	Enables you to submit a structured query about various elements of a core.
Replication	Displays current replication status for the core and lets you enable/disable replication.
Schema Browser	Displays schema data in a browser window.

### Solr Admin UI - center panel

The center of the screen shows the detail of the Solr core selected, such as statistics, summary report, and so on.

This includes a sub-navigation for the option or text or graphical representation of the requested data.

See [Solr Admin UI - left panel](#) to know more about each screen.

### Solr certificate authentication

Alfresco uses SSL and X509 certificate authentication to secure communication between the repository server and the Solr server. In this communication, SSL not only provides encryption, it is also used for authentication. Follow these steps to turn off SSL and deactivate authentication between the Alfresco repository and the Solr server.

When you install Alfresco, port 8443 is automatically configured for SSL communication between Solr and the Alfresco repository. It is recommended that you configure the Tomcat connector to use SSL and a certificate in the <TOMCAT\_HOME>/conf/server.xml file as shown below:

```
clientAuth="want"
```

For more information, see [Secure Sockets Layer \(SSL\) and the Alfresco repository](#).

The X509 client authentication allows users to authenticate to Alfresco with certificates rather than with a username and password.

To disable Solr <--> Alfresco SSL communication, follow the steps below:

1. For Solr, set the alfresco.secureComms property in the solrcore.properties file.

You can either set this property to none or https.

- Setting the alfresco.secureComms property to none or commenting it out will turn off the SSL and X509 authentication.
- Setting the alfresco.secureComms property to https will turn on the SSL and X509 authentication.



There are multiple solrcore.properties files. Make sure that each of these files must have the same value for alfresco.secureComms property.

2. For alfresco, set the solr.secureComms property in the alfresco-global.properties file.

You can either set this property to none or https.

- Setting the alfresco.secureComms property to none or commenting it out will turn off the SSL and X509 authentication.
- Setting the alfresco.secureComms property to https will turn on the SSL and X509 authentication.

## Changes from Alfresco One 5.0

The web.xml file for both Alfresco and Solr4 now has a new servlet filter, X509AuthFilter. This filter enforces SSL and X509 authentication. When the alfresco.secureComms and solr.secureComms properties are set to https, the X509AuthFilter does the following:

- Verifies that the X509 certificate is present in the request. If the cert is not present in the request, it may be due to one of the following reasons:
  - Non-ssl port being used.
  - Client did not send a certificate, or
  - Server did not request the client certificate
- Validates that the certificate dates are valid at that time.
- The X509AuthFilter filter contains an optional init parameter called cert-contains. If present, the X509AuthFilter verifies that the X509 Subject (distinguished name) of the certificate contains that string.
  - If any of these checks fail, the X509AuthFilter filter will respond with a http 403 error message.
  - If the settings described Step1 and Step2 are set to none or commented out, the X509AuthFilter filter will not enforce X509 authentication.

The X509AuthFilter has been mapped to specific paths. For Solr, all URLs will be protected by the X509AuthFilter. For Alfresco, only specific URLs in the web.xml file are protected.

The following URLs are mapped for Alfresco:

```
<web-app xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```

xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd" version="2.4">

<display-name>Alfresco</display-name>
<description>Alfresco</description>
...
<filter>
 <filter-name>X509AuthFilter</filter-name>
 <filter-class>...</filter-class>
</filter>
...
<filter-mapping>
 <filter-name>X509AuthFilter</filter-name>
 <url-pattern>/service/api/solr/*</url-pattern>
</filter-mapping>
<filter-mapping>
 <filter-name>X509AuthFilter</filter-name>
 <url-pattern>/s/api/solr/*</url-pattern>
</filter-mapping>
<filter-mapping>
 <filter-name>X509AuthFilter</filter-name>
 <url-pattern>/wcservice/api/solr/*</url-pattern>
</filter-mapping>
<filter-mapping>
 <filter-name>X509AuthFilter</filter-name>
 <url-pattern>/wcs/api/solr/*</url-pattern>
</filter-mapping>
...
</web-app>
```

Note that the `web.xml` file no longer contain the `<security-constraint>` section.

The X509 authentication only takes place on a port that is configured for both SSL and user authentication. Different application servers will configure this port in different ways. Besides configuring SSL, user authentication must also be configured for the certificate to be made available to the `X509AuthFilter`.

If you decide to turn-off SSL and deactivate authentication between Alfresco repository and the Solr server, you need to protect your environment. For more information, see [Configuring SSL for a production environment](#).

## Solr monitoring and troubleshooting

Help for monitoring and resolving any Solr index issues that might arise as a result of a transaction.

### Performing a full reindex with Solr

This task describes how to perform a full Solr reindex.

This task assumes you are using only one Solr instance for all nodes in the Alfresco cluster. If not, then you need to repeat process on each Solr instance used in the cluster.

1. Confirm the location of the Solr 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 core location is defined in the solrcore.properties file as:

For Solr, the default data.dir.root path is:

```
data.dir.root=<ALFRESCO_HOME>/alf_data/solr4/index/
```

2. Shut down Solr (if running on a separate application server).
3. 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
4. Delete all the Alfresco models for each Solr core at \${data.dir.root}.
   
<ALFRESCO\_HOME>/alf\_data/solr4/model
5. Delete the contents of the <ALFRESCO\_HOME>/alf\_data/solr4/content directory.
6. Start up the application server that runs Solr.
7. 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...
```

8. Use the Solr administration console to check the health of the Solr index.



The process of building the Solr indexes can take some time depending on the size of the repository. To monitor reindexing progress, use the Solr administration console and check the logs for any issues during this activity.

While the reindex is taking place, some searches may not return the full set of results.

## Unindexed Solr Transactions

You can check the status of the Solr index to identify the nodes to a transaction that failed to index.

To generate a general report for Solr, 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 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, 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`: resests 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 Index

Use this information 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 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 core summary. The operations run the same consistency checks that are available by URL.

### Solr troubleshooting for SSL configurations

When you have an Alfresco installation that requires an SSL configuration, you might encounter connection issues.

If Solr search and/or the Solr 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 JSSE 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](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 backup and restore

Use this information to backup and restore the Solr 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

There are a number of ways to back up the Solr indexes.

You can set the Solr 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.

 The `\alf_data\solr4\content` directory is not backed up automatically during the back up process. If you want a back up of this directory, you will have to do it manually.

### Set up Solr 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	
<b>Main Store</b>	<b>Archive Store Properties</b>
<b>Backup Location:</b> \$(dir.root)/solr4Backup/alfresco	<b>Backup Location:</b> \$(dir.root)/solr4Backup/archive
The location where the index backup is stored on the Solr server.	The location where the index backup is stored on the Solr server.
<b>Backup Cron Expression:</b> 0 0 2 * * ?	<b>Backup Cron Expression:</b> 0 0 4 * * ?
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.	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.
<b>Backups To Keep:</b> 3	<b>Backups To Keep:</b> 3
The number of backups to keep (including the latest backup).	The number of backups to keep (including the latest backup).

Here, you can specify the backup location and edit backup properties for each core of the Solr index: **Main Store** and **Archive Store**.

- **Backup Cron Expression:** Specifies a Quartz cron expression that defines when backups occur. Solr 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 should store.
4. Click **Edit**.
  5. Specify the full location path on the Solr server file system to store the index backup in the **Backup Location** text box.
  6. Click **Save**.

## Specifying Solr backup directory by using `alfresco-global.properties` file

This task shows how to specify the Solr backup directory by using `alfresco-global.properties` file.

- To set the Solr 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:

```
solr.backup.archive.remoteBackupLocation=
solr.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 indexes using JMX client

If you have installed the Oracle Java SE Development Kit (JDK), you can use the JMX client, JConsole, to backup Solr indexes, edit Solr backup properties and setup the backup directory.

- You can set the backup of Solr 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 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 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 core summary.
- To use JMX client to setup Solr 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.

### Refreshing the backup Solr indexes (optional)

- Trigger a Solr index backup using a JMX client.  
JConsole (MBeans tab -gt Alfresco/Schedule/DEFAULT/MonitoredCronTrigger/indexBackupTrigger/Operations - **executeNow** button)
- After completing this operation, the `solr4Backup` directory contains an up-to-date cold copy of the Solr indexes, ready to be backed up.

### Restoring Solr indexes

Follow these steps to restore the Solr indexes.

During the recovery process, Solr queries Alfresco to update the indexes restored from a backup. To restore the Solr indexes, use the following steps:

- Stop the Solr server.
  - Copy a backup index to the data directory for each core.  
Remember to use a backup created from the same Alfresco instance.
  - Restart the Solr server.
- Solr will start to track the indexes based on the state of the restored index.

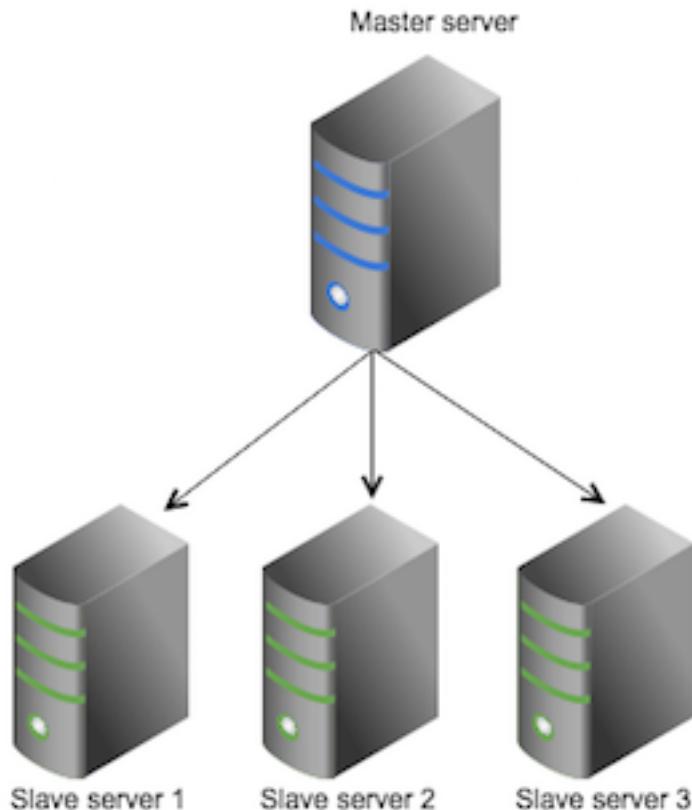
## Solr replication

Solr replication uses the master-slave model to distribute complete copies of a master index to one or more slave servers.

The master server receives all updates and all changes (such as inserts, updates, or deletes) are made against a single master server. Changes made on the master are distributed to all the slave servers which service all query requests from the clients. This division of labor enables Solr to scale to provide adequate responsiveness to queries against large search volumes.

The master server tracks the models, metadata, permissions, and content where as the slave server only tracks the models.

The figure below shows a Solr configuration using index replication. The master server's index is replicated on the slaves.



The master-slave replication requires non-SSL communication between the master server and the slave server.

### Solr replication: advantages and disadvantages

There are advantages and disadvantages of using a master-slave replication.

A system with master-slave index replication offers the following advantages and disadvantages:

#### **Advantages**

- Splits read and write load and operations
- Load distribution for search queries
- High availability for searching
- Any number of slave instances can be created to scale query performance
- Usually less frequent index updates on the slaves and better use of the cache

#### **Disadvantages**

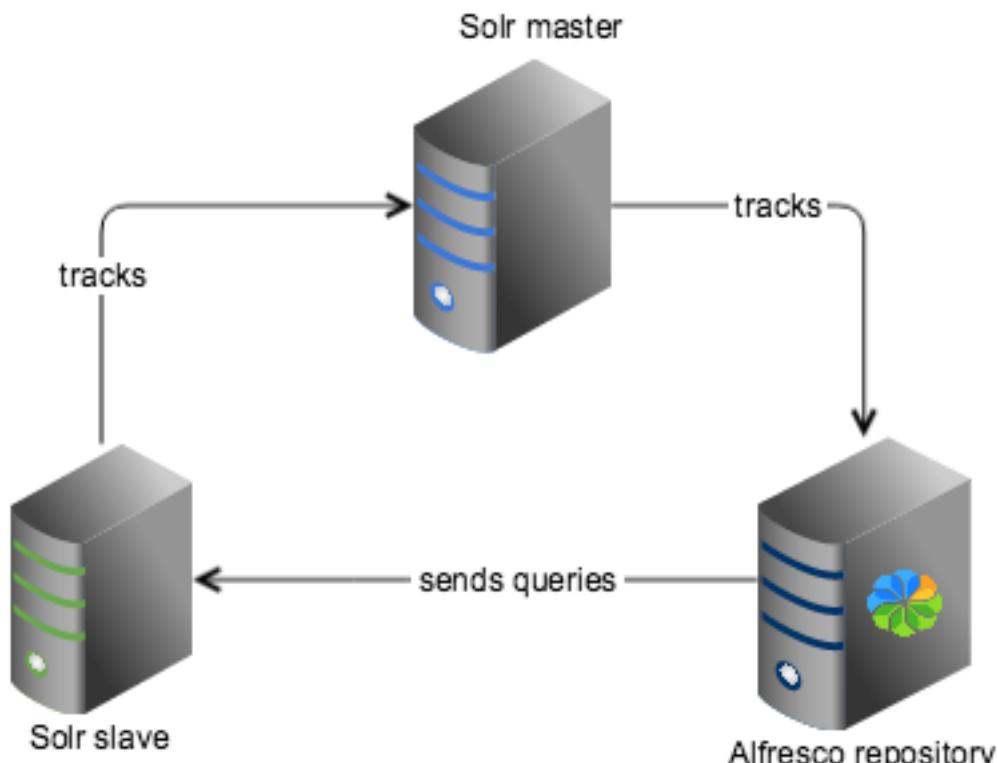
- Increased latency (sum of tracking and Solr replication latency)

- Occasional large IO load to replicate large merges
- Complicated load balance and management
- Reconfiguration if the master is lost

### Solr replication configuration

The Solr replication feature is implemented as a RequestHandler. The simplest configuration involves one Alfresco node, one Solr master, and one Solr slave.

The Solr master is configured to track the Alfresco instance while the Solr slave is configured to track the Solr master. The Alfresco instance is configured to send all the queries to the SOLR slave.



### Configuring the Alfresco instance

As usual, no SSL and queries configured to go to the slave.

### Configuring Solr master

The configuration affecting replication is controlled by a single file, `solrconfig.xml`. To configure the master server, follow the steps below:

1. Edit the `solrconfig.xml` file on the master server to change the default replication handler configuration. Remember to uncomment the `master` section.

```

<requestHandler name="/replication" class="solr.ReplicationHandler" >
 <!--
 To enable simple master/slave replication, uncomment one of the
 sections below, depending on whether this solr instance should be
 the "master" or a "slave". If this instance is a "slave" you
 will
 also need to fill in the masterUrl to point to a real machine.
 -->

 <lst name="master" >

```

```

 <str name="replicateAfter">commit</str>
 <str name="replicateAfter">startup</str>
 <str name="confFiles">schema.xml,stopwords.txt</str>
 </lst>

 <!--
 <lst name="slave">
 <str name="masterUrl">http://your-master-hostname:8983/solr</
str>
 <str name="pollInterval">00:00:60</str>
 </lst>
 -->
</requestHandler>
```

where:

Parameter name	Description
replicateAfter	<p>String specifying action after which replication should occur. Valid values are:</p> <ul style="list-style-type: none"> <li>• commit: triggers replication whenever a commit is performed on the master index.</li> <li>• optimize: triggers replication whenever the master index is optimized.</li> <li>• startup: triggers replication whenever the master index starts up.</li> </ul> <p>There can be multiple values for this parameter. If you use startup, you need to have a commit and/or optimize entry also if you want to trigger replication on future commits or optimizes.</p>
confFiles	Comma-separated list of configuration files to replicate.

2. Make sure that the `solrcore.properties` file has the following settings:

```
enable.master=true
enable.slave=false
```

## Configuring Solr slave

Here again, the `solrconfig.xml` file controls the configuration affecting replication. To configure the slave server, follow the steps below:

1. Uncomment the slave section.

```

<requestHandler name="/replication" class="solr.ReplicationHandler" >
 <!--
 To enable simple master/slave replication, uncomment one of the
 sections below, depending on whether this solr instance should be
 the "master" or a "slave". If this instance is a "slave" you
 will
 also need to fill in the masterUrl to point to a real machine.
 -->

 <!--
 <lst name="master">
 <str name="replicateAfter">commit</str>
 <str name="replicateAfter">startup</str>
 <str name="confFiles">schema.xml,stopwords.txt</str>
 </lst>
 -->

 <lst name="slave">
```

```

 <str name="masterUrl">http://your-master-hostname:8983/solr</str>
 <str name="pollInterval">00:00:60</str>
</lst>

</requestHandler>

```

where:

Parameter name	Description
pollInterval	Interval in which the slave should poll master .Format is <i>hh:mm:ss</i> . If this is missing, the slave server does not poll automatically.
masterUrl	Fully qualified URL for the replication handler of master. Make sure the <code>masterUrl</code> ends with <code>&lt;tomcat base url&gt;/solr4/alfresco</code> .

- Set the master URL to point to the Solr master. Also, set how often the slave server should poll for changes.

```

<str name="masterUrl">http://your-master-hostname:8983/solr4</str>
<str name="pollInterval">00:00:60</str>

```

- Set the following properties in the `solrcore.properties` file:

```

enable.master=false
enable.slave=true

```

In this configuration, the Solr instance will only track model changes from the Alfresco One platform.

## Additional Solr configuration

Any configuration changes related to the core schema and configuration, or any changes in `<solr_home>/conf` must be made to all Solr instances. Replication can be configured to manage the distribution of other core related configuration files.

### Solr master-slave reconfiguration

There are additional master-slave configuration requirements for Solr, such as adding a slave server and promoting a slave server.

#### Adding a slave server

To add another slave server to an existing replication configuration, see [configuring Solr slave](#).

#### Promoting a slave

In the event of a downed master in a master-slave configuration, the slave servers can continue to service queries, but will no longer be able to index until a new master is instated. The process of promoting a slave to a master is manual. The state of slave servers may differ, so choose the most up-to-date slave to promote as the master server.

To promote a slave, follow the steps below:

- Nominate the most up-to-date slave as the master.

To choose the most up-to-date slave, follow the steps below:

- Go to Solr Admin web interface using <https://localhost:8443/solr4>.
- Select the appropriate core from the **Core Selector** list.
- Select **Replication**.

The Replication screen shows the current replication status for the core, and lets you enable/disable replication. It also displays the version of the master and slave servers.

4. Identify for the slave whose index is closest to the master server or pick a slave that has the highest version.

Index	Version	Size
Master (Searching)	144105625224	160.22 KB
Master (Replicable)	144105625224	-
Slave (Searching)	144105625224	160.22 KB

2. Stop the Solr server on the new master.
3. In the `solrconfig.xml` file, replace the Solr configuration in the replication handler that defines the slave with the one that defines the master.

```
<requestHandler name="/replication" class="solr.ReplicationHandler" >
 <!--
 To enable simple master/slave replication, uncomment one of the
 sections below, depending on whether this solr instance should be
 the "master" or a "slave". If this instance is a "slave" you
 will
 also need to fill in the masterUrl to point to a real machine.
 -->

 <lst name="master">
 <str name="replicateAfter">commit</str>
 <str name="replicateAfter">startup</str>
 <str name="confFiles">schema.xml,stopwords.txt</str>
 </lst>

 <!--
 <lst name="slave">
 <str name="masterUrl">http://your-master-hostname:8983/solr</
 str>
 <str name="pollInterval">00:00:60</str>
 </lst>
 -->
</requestHandler>
```

4. Set the following properties in the `solrcore.properties` file:

```
enable.master=true
enable.slave=false
```

5. Configure all other slave servers (if any) to point to the new master server. Make sure that the state of the slave indexes is either behind or equal to the state of the master server. For more information, see [configuring Solr slave](#).

After the previously broken master server is fixed, it can either be discarded, run as a slave, or run as a second master. To run as a slave, make sure it is behind the new master. It can be restored from a back up of another slave or the current master server.

## Solr sharding

Solr sharding involves splitting a single Solr index into multiple parts, which may be on different machines. When the data is too large for one node, you can break it up and store it in sections by creating one or more shards, each containing a unique slice of the index.

Sharding is important for two primary reasons:

- It allows you to horizontally split or scale your content volume.
- It allows you to distribute operations, for example, index tracking, across shards (potentially on multiple nodes) therefore increasing performance/throughput.

Documents in the repository are distributed evenly across shards. You may have more than one shard, but a document will only be located in one shard and its instances. A conceptual shard can have any number of real instances. A shard tracks the appropriate subset of information from the repository.

 Alfresco does not support slave shards or slave replicas.

A shard can have zero or more shard instances. Multiple shard instances have the following advantages:

- It provides high availability in case a shard/node fails.
- It allows you to scale out your search throughput because searches can be executed on all the instances in parallel.
- It increases performance: search requests are handled by the multiple shard instances.

Note that if your Solr indexes are sharded, then index backup will be disabled.

### Basic Solr sharding concepts

There are a few basic concepts that are core to understanding Solr sharding. Understanding these concepts from the outset will help in learning more about sharding.

### Useful terminology

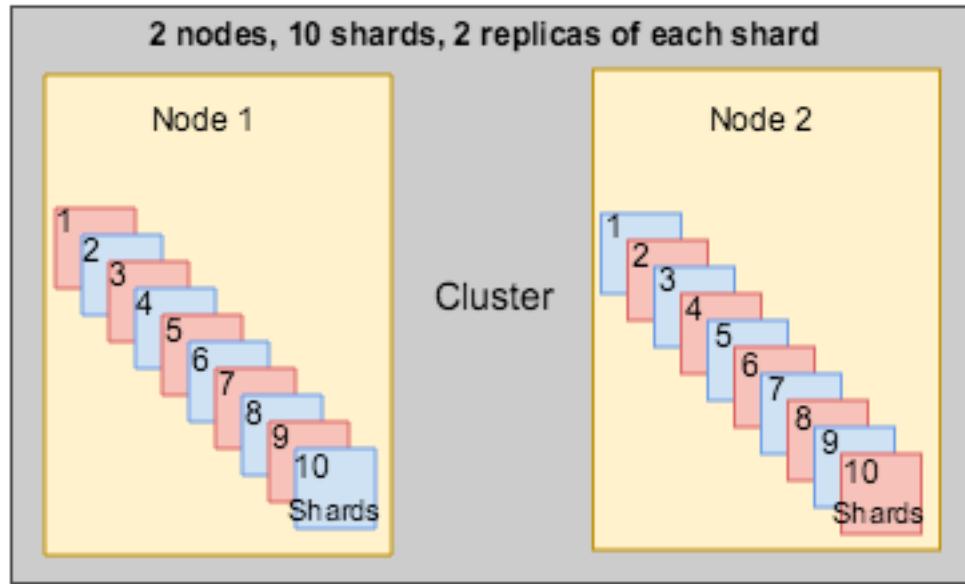
Term	Description
Node	A node represents an Alfresco instance.
Cluster	A cluster is composed of one or more Alfresco nodes.
Shard group	A shard group is a collection of documents. It is composed of one or more shards.
Shard	An index is split into chunks called shards.

### Basic concepts

A cluster is a collection of one or more nodes (servers) that provides indexing and search capabilities across all nodes. A node is a single server that is part of your cluster, stores your data, and participates in the cluster's indexing and search capabilities.

An index is a collection of documents from the same store. An index can potentially store a large amount of data that can exceed the hardware limits of a single node. To solve this problem, Alfresco provides the ability to subdivide your index into multiple pieces called shards.

When you create an index, you define the number of shards that you want. Each shard is in itself a fully-functional and independent Solr index that can be hosted on any index server. Index server includes an Alfresco node which must be in the cluster. It is recommended to have a fail over mechanism in case a shard/node fails or goes offline. As a solution, you can make one or more copies of your index's shards into shard instances.

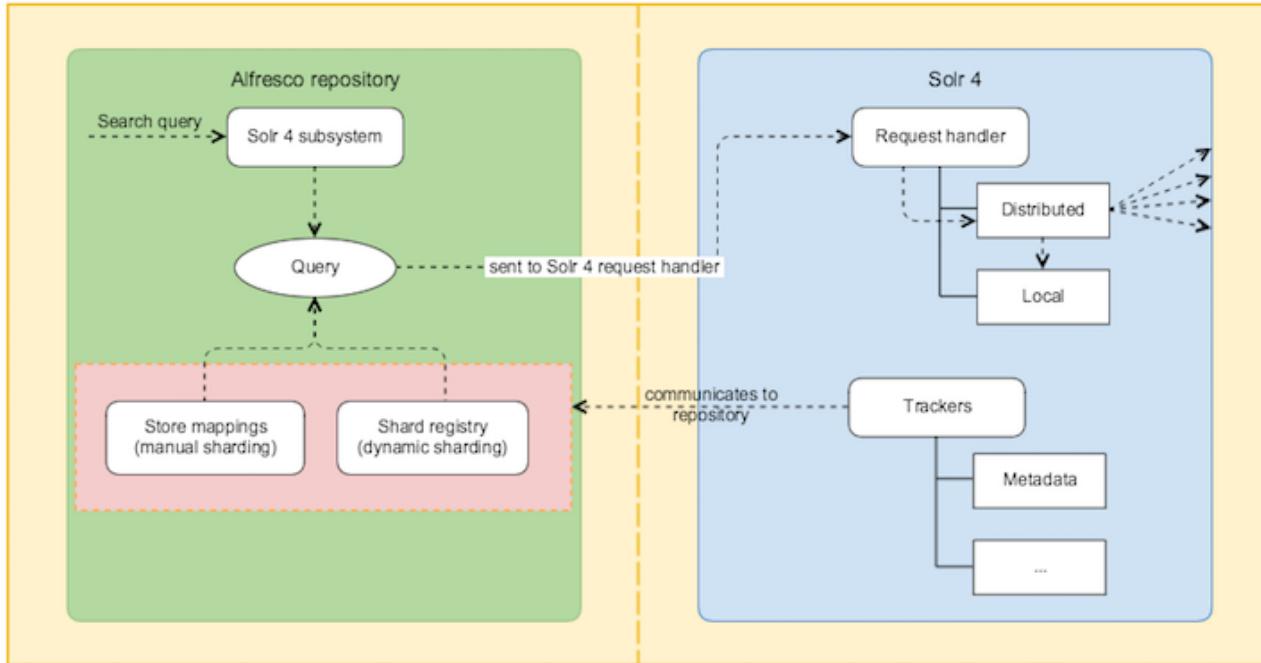


To summarize, each index can be split into multiple shards. An index can also be replicated zero (meaning no instance) or more times. A shard tracks the appropriate subset of information from the repository. The number of copies of the total index depends on the minimum number of instances for each shard.

### Setting up Solr sharding

After creating the shards manually, an Alfresco One administrator has to instruct Alfresco about how to find the indexes. This can either be done manually by configuring the indexes, or by allowing Alfresco to discover shards dynamically. This section describes how to create and configure Solr sharding.

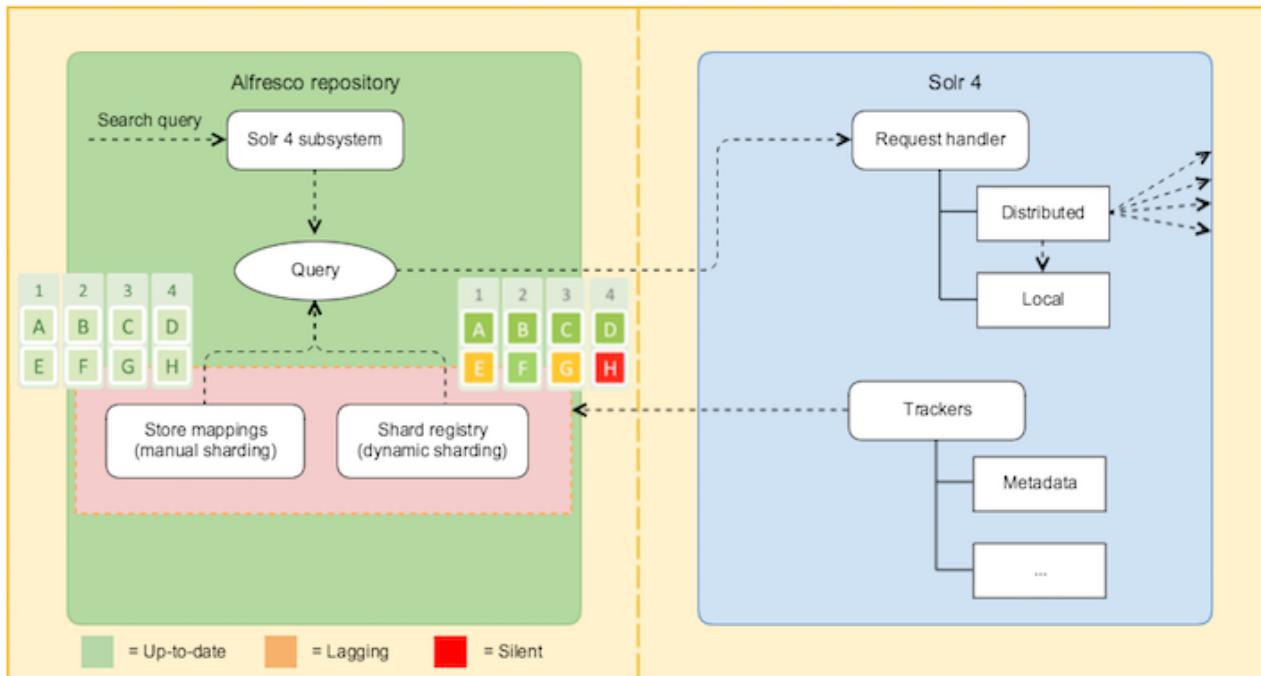
As shown in the diagram below, the trackers communicate with the Alfresco repository. When the user initiates a query, it can either be executed by manually mapping the stores (explicit configuration), or by shard registry via dynamic sharding. Dynamic sharding determines what best shards are available to answer a query. The shard registry on Alfresco stores all the information about that particular index, for example the status of the index, transactions in index, and so on.



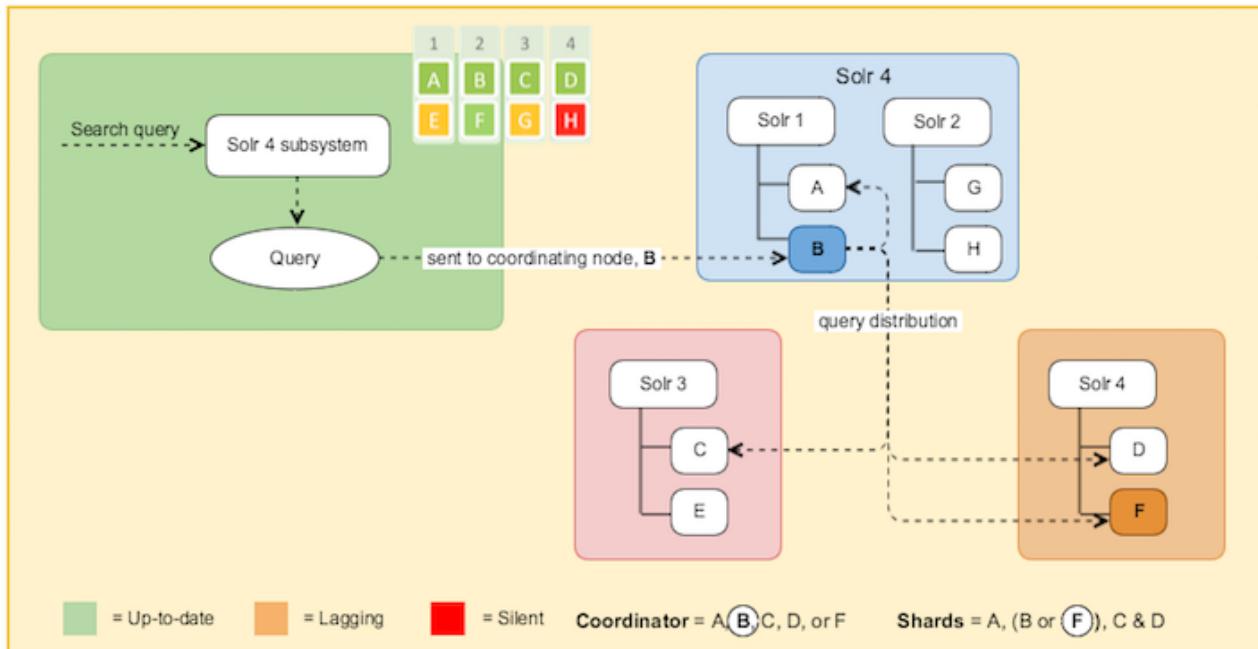
The query is sent to Solr and then to the request handler. The request handler determines if the query is local or distributed. In case of distributed query, the query is sent to other parts of the index and then combined into an overall result.

The distributed query is done in two phases. Phase 1 involves query and an initial round of faceting, and Phase 2 involves pulling back information from each relevant document and facet refinement.

The following diagram shows the difference between manual and dynamic sharding. In this example, there are 4 shards (1, 2, 3, and 4) and 2 instances for each shard (A & E, B & F, C & G, and D & H). Instances A, B, C, D, and F are up-to-date, while the instances E and G are lagging behind and can't be used. Shard instance H is silent and therefore, unavailable for querying.



In manual sharding, the user is only aware of the existence of the shards and its instances but knows nothing about the status of each shard and its instance(s). So, the query can be sent to any instance. In dynamic sharding, Alfresco will use instance A, B, C, D, or F for querying.



At query time, Solr is aware of all the available nodes and selects one node as the coordinator (one node from all the available green ones) and sends the request to it. Also, the shards (A, B, C, D or A, F, C, D) to be used for that request are selected dynamically. In this case, Solr selects F instead of B. So, if one node lags behind or stops responding, Solr stops using it.

Click each method to know more about creating shards.

- [Manual Solr sharding](#)
- [Dynamic Solr sharding](#)

### Creating Solr shards manually

You can create, configure, and register shards explicitly using ACL based hash sharding.

### How to setup sharding?

For archive and workspace stores, create shards and assign shards and instances to nodes. For example, if  $(\text{numShards} * \text{replicationFactor}) \% \text{numNodes} == 0$ , the following query will create the appropriate shards on any given node from a specified node set:

```
http://<host1>:<port1>/solr4/admin/cores?action=newCore&storeRef=archive://
SpacesStore&numShards=10
&numNodes=1&nodeInstance=1&replicationFactor=1
```

To put the index in a specified location set the properties as shown below:

```
http://<host1>:<port1>/solr4/admin/cores?action=newCore&storeRef=workspace://
SpacesStore&numShards=10
&numNodes=1&nodeInstance=1&property.data.dir.root=<
```

### Example: Creating shards

Let's consider an example for creating 12 shards, 2 instances of each shard, and 6 nodes. As shown below, each node will get 4 different shards.

	Shard 0	Shard1	Shard 2	Shard 1	Shard4	Shard 5	Shard 6	Shard 7	Shard 8	Shard 9	Shard 10	Shard 11
Node1	x						x	x				
Node2		x	x					x	x			
Node3			x	x					x	x		
Node4				x	x					x	x	
Node5					x	x					x	x
Node6	x					x	x					x

To achieve this sharding configuration, follow the steps below:

1. Setup and configure the Solr nodes.

- a. Set up the six Solr nodes.

Delete the alfresco and archive cores using the following commands:

```
https://localhost:8443/solr4/admin/cores?
action=removeCore&storeRef=workspace://SpacesStore&coreName=alfresco
https://localhost:8443/solr4/admin/cores?
action=removeCore&storeRef=workspace://SpacesStore&coreName=archive
```

- b. Configure each Solr node and start index tracking.

- For node 1:

```
http://<host1>:<port1>/<base1>/admin/cores?
action=newCore&storeRef=workspace://SpacesStore&
numShards=12&nodeInstance=1&replicationFactor=2&numNodes=6&template=reran
```

- For node 2:

```
http://<host2>:<port3>/<base3>/admin/cores?
action=newCore&storeRef=workspace://SpacesStore&
numShards=12&nodeInstance=2&replicationFactor=2&numNodes=6&template=reran
```

- For node 3:

```
http://<host3>:<port3>/<base3>/admin/cores?
action=newCore&storeRef=workspace://SpacesStore&
numShards=12&nodeInstance=3&replicationFactor=2&numNodes=6&template=reran
```

- For node 4:

```
http://<host4>:<port4>/<base4>/admin/cores?
action=newCore&storeRef=workspace://SpacesStore&
numShards=12&nodeInstance=4&replicationFactor=2&numNodes=6&template=reran
```

- For node 5:

```
http://<host5>:<port5>/<base5>/admin/cores?
action=newCore&storeRef=workspace://SpacesStore&
numShards=12&nodeInstance=5&replicationFactor=2&numNodes=6&template=reran
```

- For node 6:

```
http://<host6>:<port6>/<base6>/admin/cores?
action=newCore&storeRef=workspace://SpacesStore&
numShards=12&nodeInstance=6&replicationFactor=2&numNodes=6&template=reran
```

- c. For each core, the properties can be set at the creation time or updated later.

```
https://<host>:<port>/solr4/admin/cores?
action=updateCore&storeRef=system://system&
property.data.dir.store=<SOME VALUE>
```

You should now have six nodes with four cores, each actively tracking the repository. The following URL options are available for use:

URL option	Description	Example
numShards	Specifies the number of logical shards.	12
storeRef	Specifies reference to a node store.	workspace://SpacesStore
template	Defines the base configuration for a new Solr core with some configuration properties set using the URL as shown in Step 1(b).  http://<host6>:<port6>/<base6>/admin/cores?action=newCore&storeRef=workspace://SpacesStore&numShards=12&nodeInstance=6&replicationFactor=2&numNodes=6&temp...  For more information, see <a href="#">Core templates</a> .	template=rerank
replicationFactor	Specifies the number of copies of each document (or, the number of physical instances to be created for each logical shard.) A replicationFactor of 3 means that there will be 3 instances for each logical shard.	2
nodeInstance	Specifies the Solr node instance being configured.	6
numNodes	Returns the total number of nodes.	6
coreName	Specifies the name of the Solr core.	alfresco
property.<>	Specifies the property and its value.	property.data.dir.store=

## 2. Configure Alfresco by setting the Solr subsystem properties.

Set the three Solr subsystem properties for both the `alfresco` and `archive` cores in the `alfresco-global.properties` file:

```
solr4.store.mappings.value.solrMappingAlfresco.nodeString=
solr4.store.mappings.value.solrMappingAlfresco.numShards=
solr4.store.mappings.value.solrMappingAlfresco.replicationFactor=

solr4.store.mappings.value.solrMappingArchive.nodeString=
solr4.store.mappings.value.solrMappingArchive.numShards=
solr4.store.mappings.value.solrMappingArchive.replicationFactor=
```

For example, set the following properties as shown below:

```
solr4.store.mappings.value.solrMappingAlfresco.nodeString=<host1>:<port1>/<url1>,<host2>:<port2>/<url2>
solr4.store.mappings.value.solrMappingAlfresco.numShards=12
solr4.store.mappings.value.solrMappingAlfresco.replicationFactor=2
```

 These properties can also be configured via a JMX client or using the subsystem properties to reference the composite beans.

Some important things to know:

- If the host, port, or URL is missing, the subsystem default values (the ones set for a single index) will be used.
  - Make sure that the hosts are in the correct order. This is because Solr assumes that the shards are located on node 1, etc as defined in the above list when generating queries.
  - At query time, a Solr core is selected at random to do the distribution of all shards, again selected at random.
3. Set the configuration properties that applies to all the cores in a Solr instance in the `shared.properties` file.

For shard registration, Alfresco needs to know to which Solr port the requests should be sent. This can be configured, along with an explicit host name.

```
solr.host=localhost
solr.port=8080
```

These properties will be used when registering all cores found under the `<SOLR_HOME>` directory. For more information, see [About shared.properties file](#).

### Core templates

Core templates are used to define the base configuration for a new Solr core with some configuration properties.

The core templates are specified in the URL used for creating shards, as shown below:

```
http://<host1>:<port1>/<base1>/admin/cores?action=newCore&storeRef=workspace://
SpacesStore&numShards=12
&nodeInstance=1&replicationFactor=2&numNodes=6&template=<template>
```

The `<ALFRESCO_HOME>/solr4/templates` directory contains the following structure:

Templates	Description
rerank	This template is an enhanced core configuration for Alfresco One 5.1. To use <code>rerank</code> , you need to reindex using this template, when creating a new core. It has more appropriate settings for sharding and supports indexes containing approximately 50-80M documents per shard.
test	
vanilla	This template matches how the <code>alfresco</code> and <code>archive</code> cores were defined in Alfresco One 5.0. In addition, it supports auto-phrase and query re-ranking.
without_suggest	This template is based on the <code>vanilla</code> template but it does not support suggestion.

The core templates include `schema.xml` and `solrconfig.xml`. The main purpose is to create multiple cores on multiple machines with the same configuration.

### Comparison between the `rerank` and `vanilla` templates

No.	Rerank template	Vanilla template
1	The <code>rerank</code> template causes less duplication of the index, and therefore the index is more compact.	The <code>vanilla</code> template causes more duplication of the index, and therefore the index is large.

No.	Rerank template	Vanilla template
1	<p>In the rerank template, stop words are included and indexed as common grams. By default, majority of the 100 most frequently used words in English language text are now treated as stop words.</p> <p>For more information, see &lt;ALFRESCO_HOME&gt;/solr4/templates/rerank/conf/lang/stopwords_en.txt.</p>	<p>In the vanilla template, stop words are removed from the words that are tokenised in English language.</p> <p>For more information, see &lt;ALFRESCO_HOME&gt;/solr4/templates/vanilla/conf/lang/stopwords_en.txt.</p>
2	<p>The rerank template supports real rerank with automatic phrasing* (or auto-phrasing). Queries are run in two stages:</p> <ol style="list-style-type: none"> <li>Stage one treats phrases as conjunctions and ignores expensive positional information.</li> <li>Stage two reranks the top queries using a more expensive phrase.</li> </ol> <p>* When a user provides individual search terms in a query, the automatic phrasing feature groups those individual terms into a search phrase and returns the query results for the phrase.</p>	<p>The vanilla core performs auto-phrasing without re-ranking by The auto-phrase is added to the query.</p>

### About `shared.properties` file

The <ALFRESCO\_HOME>/solr4/conf/shared.properties file is used to set configuration that applies to all the cores in a Solr instance.

Most of these settings need to be replicated across all the Solr instances that are a part of the sharded index. However, there are some properties related to dynamic shard registration, such as host and port, which can be set for each machine.

These Solr instance specific settings can be omitted but you may have to define the correct host that the repository will use to communicate to Solr, for example, using an internal IP addresses in a cloud environment. By default, the host is detected by Java, the port will default to 8080, and the tomcat port is either determined by JMX or that explicitly defined in the shared.properties file.

The shared.properties file defines the:

- properties that are treated as identifiers
- properties that are used to generate suggestions
- data types that support cross locale/word splitting/token pattern
- properties that support cross locale/word splitting/token pattern
- solr.host property
- solr.port property

### Properties defined in the `shared.properties` file

You can define which properties are treated as identifiers, regardless of how they are defined in the model. These are properties must not be tokenised. If this list is changed, a reindex is required. You can also reindex by query. For more information, see [Reindex documents by query](#).

If you rename the `shared.properties.sample` file to `shared.properties`, it will use the same set of identifier properties that are used in Alfresco One 5.0.

```
Properties treated as identifiers when indexed

alfresco.identifier.property.0={http://www.alfresco.org/model/
content/1.0}creator
alfresco.identifier.property.1={http://www.alfresco.org/model/
content/1.0}modifier
alfresco.identifier.property.2={http://www.alfresco.org/model/
content/1.0}userName
alfresco.identifier.property.3={http://www.alfresco.org/model/
content/1.0}authorityName
```

You can define which properties are used for suggestion.

```
Suggestable Properties

#alfresco.suggestable.property.0={http://www.alfresco.org/model/
content/1.0}name
#alfresco.suggestable.property.1={http://www.alfresco.org/model/
content/1.0}title
#alfresco.suggestable.property.2={http://www.alfresco.org/model/
content/1.0}description
#alfresco.suggestable.property.3={http://www.alfresco.org/model/
content/1.0}content
```

Suggestion can also be configured for the search sub-system and for any SOLR core using properties. If the `shared.properties` file is missing in Alfresco One 5.1, suggestion will be configured as it is in Alfresco One 5.0.

You can define which properties are used for tokenisation with the Solr word delimiter factory.

```
Data types that support cross locale/word splitting/token patterns if
tokenised

alfresco.cross.locale.property.0={http://www.alfresco.org/model/
content/1.0}name
```

You can define which property types are used for tokenisation with the Solr word delimiter factory.

```
Data types that support cross locale/word splitting/token patterns if
tokenised

alfresco.cross.locale.datatype.0={http://www.alfresco.org/model/
dictionary/1.0}text
alfresco.cross.locale.datatype.1={http://www.alfresco.org/model/
dictionary/1.0}content
alfresco.cross.locale.datatype.2={http://www.alfresco.org/model/
dictionary/1.0}mltext
```

## Support for cross-language search

The cross core configuration options to use specific locales for cross-locale searches are set in the `shared.properties` file. Cross language search uses the appropriate stemmed tokens for all locales.

For backward compatibility, this file is absent in Alfresco One 5.1 to provide options equivalent to Alfresco One 5.0.

To configure cross-language search, follow the steps below:

1. Open the `<ALFRESCO_HOME>/solr4/conf/shared.properties.sample` file.

2. Set the following properties:

```
alfresco.cross.locale.property.0={http://www.alfresco.org/model/
content/1.0}name
alfresco.cross.locale.property.1=...
```

This sets the properties that should be dual tokenised.

The cross-language search in Alfresco One 5.0 is now only used to provide support to split tokens (based on case and numbers) to generate *in word* tokens. The *in word* tokenisation is mainly used for name. For example, find RedDog12 by Red, Dog, or 12, Dog12, and so on. This property must be indexed and tokenised.

3. To specify the same behaviour based on the data type, set the following properties:

```
alfresco.cross.locale.datatype.0={http://www.alfresco.org/model/
dictionary/1.0}text
alfresco.cross.locale.datatype.1=...
```

### Query time expansion of locales

Query time expansion of locales can be defined in the `solrconfig.xml` file as part of the query language definition.

Locale parameter	What is it?
<code>autoDetectQueryLocale</code>	If true, this uses the query typed in by the user to detect the locale.
<code>autoDetectQueryLocales</code>	This specifies a set of locales. One of these may be used in executing the query if <code>autoDetectQueryLocale=true</code> .
<code>fixedQueryLocales</code>	This specifies a fixed set of locales always used by the query.

What locales are used?

- The locale for the current session is always used.
- If the `autoDetectQueryLocale` parameter is used, then the best match from `autoDetectQueryLocales` is used. If no parameter is set, then all the possible locales are used.
- All `fixedQueryLocales` are used.

Here are some example entries in the `solrconfig.xml` file:

```
<queryParser name="afts"
 class="org.alfresco.solr.query.AlfrescoFTSQParserPlugin">
 <str name="rerankPhase">QUERY_PHASE</str>
 <str name="autoDetectQueryLocale ">true</str>
 <str name="autoDetectQueryLocales ">en,fr,de</str>
</queryParser>

<queryParser name="afts"
 class="org.alfresco.solr.query.AlfrescoFTSQParserPlugin">
 <str name="rerankPhase">QUERY_PHASE</str>
 <str name="fixedQueryLocales">en,fr,de</str>
</queryParser>
```

These are query time options and do not require a reindex. Currently, these values cannot be set in the `solrcore.properties` file.

### Dynamic shard registration

In dynamic shard registration, shards register as a part of the tracking process to form indexes, thereby eliminating the need to follow the manual shard distribution pattern over Solr nodes.

Unlike manual sharding, dynamic sharding does not require shards and instances to be distributed correctly over a known set of hosts. Query is resilient, with a configurable delay to instances coming and going. For manual sharding, all instances must be available on the expected host at the expected URL. While dynamic shard registration allows different numbers of instances for any shard, manual sharding does not.

To enable dynamic sharding, set the following property in the `alfresco-global.properties` file:

```
solr.useDynamicShardRegistration=true
```

The following properties govern which instances are chosen for a query:

```
search.solrShardRegistry.purgeOnInit=true
search.solrShardRegistry.shardInstanceTimeoutInSeconds=300
search.solrShardRegistry.maxAllowedReplicaTxCountDifference=1000
```

Property	Description	Example
<code>search.solrShardRegistry.purgeOnInit</code>	If <code>true</code> , this property removes persisted shard state from the database when the subsystem starts.	<code>true</code>
<code>search.solrShardRegistry.shardInstanceTimeoutInSeconds</code>	Specifies that a shard has not made a tracking request within this time, it will not be used for query.  When tracking large change sets or rebuilding your indexes, increase the shard timeout. For example, change the value of this property to 3200 or 7200 seconds.	300 seconds
<code>search.solrShardRegistry.maxAllowedReplicaTxCountDifference</code>	Specifies the count difference more than this number of transactions behind the leading instance, it will not be used.	1000 transactions

If there is more than one index for a store, the most up to date index (the one that has indexed most transactions) will be used. For each shard, an instance is chosen at random from all the shards that are actively tracking and within 1000 transactions of the lead instance.

Shards are considered to be part of the same index if they:

- track the same store
- use the same template (and therefore, Solr schema)
- have the same number of shards
- use the same partitioning method with the same configuration, if any is required
- have the same setting to transform or ignore content

In dynamic sharding, shards can be created using the same API as manual sharding or you can list the required shards as a comma-separated list of `shardIds`.

```
http://localhost:8080/solr4/admin/cores?action=newCore&storeRef=workspace:///SpacesStore&numShards=10&numNodes=1&nodeInstance=1&property.data.dir.root=<>&shardIds=0,1,2,3,4
```

The status of all the available indexes, shards, and instances can be found using a JMX client. For more information, see [Indexing information available in a JMX client](#).

Dynamic sharding will currently use partial indexes to answer queries. For example, there are two shards: Shard1 and Shard2. If there are no instances for Shard2, queries will only use Shard1.

## Installing and configuring Solr shards

Follow these steps to set up sharding of a non-sharded index or change the number of instances of an already sharded index.

Do not use SSL with sharding.

1. Create machines to host Solr shards.
  - a. These machines are basically application servers that hosts Solr webapp. If you install multiple Solr webapps on the same machine, each Solr instance must have a different configuration. In the `solr4.xml` file, edit the following parameters so that all Solr instances point to different root directories for each node:
    - `solr/home`
    - `solr/model/dir`
    - `solr/content/dir`
2. Install and start Alfresco One 5.1. For more information, see [Installing Alfresco using setup wizards](#).
3. Remove the existing Solr indexes from the installation.

Delete the alfresco and archive cores using the following commands:

```
https://localhost:8443/solr4/admin/cores?
action=removeCore&storeRef=workspace://SpacesStore&coreName=alfresco
https://localhost:8443/solr4/admin/cores?
action=removeCore&storeRef=workspace://SpacesStore&coreName=archive
```

4. Add any custom core templates. For more information, see [Core templates](#).
5. Configure the `<SOLR_HOME>/conf/shared.properties` file. For more information, see [About shared.properties file](#).
6. Start the Solr server.
7. Create your new index shards and instances by configuring the properties on the URL.

```
http://localhost:8080/solr4/admin/cores?
action=newCore&storeRef=workspace://SpacesStore&
numShards=10&numNodes=1&nodeInstance=1&template=rerank&property.data.dir.root=<>&property.alfresco.port=8090
```

where:

- `nodeInstance` is the actual Solr instance corresponding to that host:port.
- `numNodes` is the total number of Solr hosts.

When you install Alfresco One 5.1 using the installer, the templates used to create shards do not use the port specified in the installer. Here's an example to show how to set a non-SSL port manually when creating a shard.

**Example:** If you want a sharded Solr installation with a different Tomcat port (8090), set the `property.alfresco.port` property on the URL used to create the shard. The `property.alfresco.port` property specifies the port used to communicate with the repository (or repositories through a load balancer). This property can also be set if communicating through a different host or load balancer. In this example, we will set `property.alfresco.port=8090`, as shown below:

```
http://localhost:8080/solr4/admin/cores?
action=newCore&storeRef=workspace://SpacesStore&
numShards=10&numNodes=1&nodeInstance=1&template=rerank&property.data.dir.root=<>&property.alfresco.port=8090
```

8. The Solr cores will register and start tracking the indexes.

If there are two indexes for the same store, the old index will be used until both the indexes are at the same state. Thereafter, both the indexes will be used.

- Set the following properties in the `alfresco-global.properties` file.

```
solr.secureComms=none
solr.useDynamicShardRegistration=true
```

- Restart Alfresco.
- You can turn off any old indexes from tracking. To do so, wait for the instances to time out and let the new index to be up-to-date. Alternatively, navigate to the JMX sharding operations and clear out all the registered shards, and start again.

You have a new live index.

### *Configuring Solr sharding using the Admin Console*

Prerequisites for viewing the **Index Server Sharding** page:

- Check that you have a valid Alfresco One 5.1 license.
- Support for shard groups requires a clustered license. Make sure that you enable clustering on your Alfresco One license. For more information, see [Repository server clustering](#) and [Uploading a new license](#).

- Open the Admin Console. For more information, see [Launching the Admin Console](#).

- In **Repository Services**, click **Index Server Sharding**.

You see the **Index Server Sharding** page. It displays information about dynamic shard index registration, shard groups, and shard instances.

- Under **Dynamic Shard Instance Registration**, set the shard instance properties.

Shard registration property	Example setting	What is it?
<b>Dynamic Shard Instance Registration Enabled</b>	Yes	Select this property to enable dynamic shard instance registration. If disabled, manual shard registration is used.
<b>Purge at Startup</b>	No	This property purges all persisted dynamic shard instance information at startup.
<b>Instance Timeout (seconds)</b>	100	This specifies the number of seconds a shard instance can go without making a tracking call for transactions to the repository before it stops being used for queries.  When tracking large change sets or rebuilding your indexes, increase the shard timeout. For example, change the value of this property to 3200 or 7200 seconds.
<b>Max Instance Transaction Lag</b>	1000	This specifies the maximum number of transactions a instance can lag behind the lead instance of the shard before it stops being used for queries.

- Click **Refresh** to refresh this page.
- Click **Purge** to remove all registered shard instance information and start from clean.

6. Click **Clean** to remove inactive registered shard instance information.

7. Click **Manage** to create and manage shard instances.

You see the **Index Server Shard Management** window. Use this window to create individual shards or shard groups.

- a. Use **Existing Index Servers** to view a list of existing index servers and to create new index servers.

1. To add a new index server, specify the server address in **New Index Server** and click **Add**.

You can view the newly created index server under **Existing Index Servers**.

2. You can view a list of the core names already in use under **Existing Core Names**.

- b. Under **Existing Index Servers**, click **Add to Target Index Servers** next to the server you want to add to the list of target index servers.

**Target Index Servers** displays a list of index servers where you want to make the new shards.

The selected index server appears in the list under **Target Index Servers**.

- c. To create a shard group, set the following properties under **New Shard Group**:

Shard group property	Example setting	What is it?
<b>Template</b>	rerank	This specifies the template used for the shard group.
<b>Store</b>	workspace://SpacesStore	This specifies the stores that are queryable for all shards.
<b>Core</b>		This specifies the name of the Solr core.
<b>Properties</b>	solr.suggester.enabled	This specifies the properties to set on the Solr instances. These are the same properties that are set in the solrcore.properties file.
<b>Shards</b>	1	This specifies the total number of shards.
<b>Instances</b>	1	This specifies the total number of instances.

- d. Click **Create Shards Group** to create new shards based on the ordered list of target index servers.

- e. To create a single shard instance, set the following properties under **New Shard Instance**:

Shard property	Example setting	What is it?
<b>Index Server URL</b>	localhost:8080/solr4	This specifies the URL to a single index server.
<b>Nodes</b>	1	This specifies the total number of Solr nodes that have been created.

Shard property	Example setting	What is it?
<b>Target Index Server</b>	1	This specifies, out of all the solr nodes above, the number given to the target index server node for this new shard.
<b>Shards</b>		This specifies the specific shards to create, on the node given above. You can also specify a comma-separated list of shards.

See [Installing and configuring Solr shards](#) to view examples of creating shards when calling the REST URLs directly.

- f. Click **Create Shards** to create the new shard based on the specified instance properties.
  - g. Use **Report** to get detailed information on shard creation and execution.
  - h. Click **Close** to close the **Index Server Shard Management** window.
8. Under **Shard Groups**, you can view information about all the shards in the group.

Shard registration property	Example setting	What is it?
<b>Template</b>	rerank	This specifies the template used for the Solr core.
<b>Low Instance Shards</b>		This specifies a list of shards that have less than the maximum number of instances.
<b>Missing Shards</b>	100	This specifies a comma-separated list of shards with no instances.
<b>Max Repository Transaction ID</b>	14,637	This specifies the maximum number of transaction IDs in the repository.
<b>Max Live Instances</b>	1	This specifies the maximum number of instances available for any shard that can be used to answer a query.
<b>Remaining Transactions</b>	2	This specifies the maximum number of transactions remaining for all the lead instances of all the active shards.
<b>Number of Shards</b>	4	This specifies the total number of shards.
<b>Min Active Instances</b>	1	This specifies the minimum number of instances available for any shard that can be used to answer a query.
<b>Max Changeset ID</b>	104	This specifies the highest change set id in the repository.

Shard registration property	Example setting	What is it?
<b>Mode</b>	MASTER	This specifies whether the instances are SLAVE, MASTER, or MIXED.  ⚠ The SLAVE and MIXED instances are not supported for a sharded installation.
<b>Stores</b>	workspace://SpacesStore	This specifies the stores that are queryable for all instances.
<b>Has Content</b>	Enabled	This property is enabled if content is included for all instances.
<b>Shard Method</b>	MOD_ACL_ID	This specifies the method used to define shards.

9. Use the instance property table to view detailed entity information for all the shards. This is the same information that is displayed in the JMX console, for example, Base URL, Host, Last Indexed Changeset Date, and more.

For more information, see [Indexing information available in a JMX client](#).

- Click **Summary** to go to the <http://localhost:8080/solr4/admin/cores?action=SUMMARY> page on Solr for the specific core.

For more information, see [Unindexed Solr Transactions](#).

- Click **SOLR** to go to the Solr Admin screen for the specific core.

For more information, see [Connecting to the SSL-protected Solr web application](#).

10. The **Shard Group Report** section displays information about the shard groups and instances. A tabular view of this information is displayed in the shard table in Step 9. This information is read-only.

11. Click **Save** to apply the changes you have made to the index server shards.

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



Alfresco recommends that you do not use the **Solr Admin Console > Core Admin > Unload** functionality to unload indexes (either whole indexes or shards that are part of an index). Unloading an index or a shard in this way will delete it and make it unavailable for query.

If you unload or delete a shard from the Solr Admin Console, make sure you restart the Solr server and restore your indexes so that Alfresco can work properly.

#### *Indexing information available in a JMX client*

You can use a JMX client, such as JConsole, for monitoring the status of all the available indexes, shards and its instances, and other related information.

The JMX view of all the instances, shards, and indexes that stick together is displayed at the **MBeans > Alfresco > FlocAdmin > Attributes > Flocs** node. The **Flocs** node displays a tabular view of all the indexes formed by shard instances by registering with any member of the Alfresco cluster.

- Open a command console.
- Enter the following command:

```
jconsole
```

The **JConsole: New Connection** window displays.

- Double-click on the Alfresco Java process.

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

The **Java Monitoring & Management** window displays.

4. Select the **MBeans** tab.

The available managed beans display in JConsole.

5. Navigate to **Alfresco > FlocAdmin**.

The **Attributes** and **Operations** display below it in the tree.

6. Select **Attributes**.

- **Floc/Index level information**

All instances that stick together to form an index have the same value for the following settings:

Attribute name	Description	Is configurable or displays state	Example value
activeTrackingMode	Specifies if the instances for the index are all SLAVE, MASTER, or MIXED.  The SLAVE and MIXED instances are not supported for a sharded installation.	State	MASTER
hasContent	If the index contains content, the value of this attribute is true, otherwise false.	Configurable	true
lowReplicaShards	Specifies a comma separated list of shards that have less than maxReplicas.	State	
maxReplicas	Specifies the number of instances for the shard which has the maximum number of instances.	State	1
maxRepoChangesetId	Specifies the maximum changeset id in the repository.	State	5029
maxRepoTxId	Specifies the maximum transaction id in the repository.	State	16903
maxTransactions	Specifies the maximum number of transactions in any instance.	State	
minReplicas	Specifies the number of instances for the shard which has the minimum number of instances.	State	1

Attribute name	Description	Is configurable or displays state	Example value
missingShards	Specifies a comma separated list of shards with no instances.	State	
numberOfShards	Specifies the total number of shards.	Configurable	2
shardMethod	Specifies how the nodes and ACLs are split into shards.	Configurable	MOD_ACL_ID
shards	Click to displays tabular data for each shard.	Displays details	<b>shards</b>
stores	Specifies the stores that are indexed.	Configurable	workspace://SpacesStore
template	Specifies the name of the template used to create each core with common configuration.	Configurable	rerank

- **Shard level information**

You can navigate through each shard using the tabular navigation.

Attribute name	Description	Is configurable or displays state?	Example value
#	Specifies the shard number.	Configurable	0
activeCount	Specifies the number of instances that are currently able to answer queries.	State	1
activeTrackingMode	Specifies if the instances for the shard are all SLAVE, MASTER, or MIXED.  The SLAVE and MIXED instances are not supported for a sharded installation.	State	MASTER
laggingCount	Specifies the number of instances that are currently unable to answer queries because they are too far behind.	State	0
maxTransactionsRemaining	Specifies the maximum number of transactions left to index for any shard instance.	State	0

Attribute name	Description	Is configurable or displays state?	Example value
maxTxId	Specifies the maximum number of transaction id indexes by any instance.	State	16903
silentCount	Specifies the number of instances that are no longer tracking.	State	
replicas	Provide detail for each instance in the shard.	Displays details	<b>Instances</b>

- **Instance level information**

Attribute name	Description	Displays location or state?	Example value
baseUrl	Specifies the URL to access the instance.	Location	/solr4/ alfresco-0/
host	Specifies the host where the instance is located.	Location	172.31.42.83
port	Specifies the port on the host where the instance is located.	Location	
lastIndexedChangeSet	Specifies the date and time of the last indexed changeset.	State	Wed Oct 28 12:09:41 GMT 2015
lastIndexedChangeSetId	Specifies the last indexed changeset id in the repository.	State	5029
lastIndexedTxCommitTime	Specifies the date and time of the last indexed transaction.	State	Wed Oct 28 12:30:33 GMT 2015
lastIndexedTxId	Specifies the transaction id of the last indexed transaction.	State	16903
lastUpdated	Specifies when the instance was last updated.	State	Wed Oct 28 13:31:30 GMT 2015
state	Specifies if the instance state is ACTIVE, SILENT, or LAGGING.	State	ACTIVE
trackingMode	Specifies if the tracking is performed by the master.	State	MASTER
transactionsRemaining	Specifies the number of transactions remaining to be indexed.	State	5

## 7. Select Operations.

`removeAgedOutShards` removes all the shards which are too far behind and no longer tracking or are unresponsive.

`removeAll` removes all the shards that have registered and starts from clean.

- If you are using a sharded installation, go to **MBeans > Alfresco > Configuration > Search > managed > solr4 > Attributes** and set the number of filters using the `solr.defaultShardedFacetLimit` property.

```
solr.defaultShardedFacetLimit=20
```

- If you are using a non-sharded installation, go to **MBeans > Alfresco > Configuration > Search > managed > solr4 > Attributes** and set the number of filters using the `solr.defaultUnshardedFacetLimit` property.

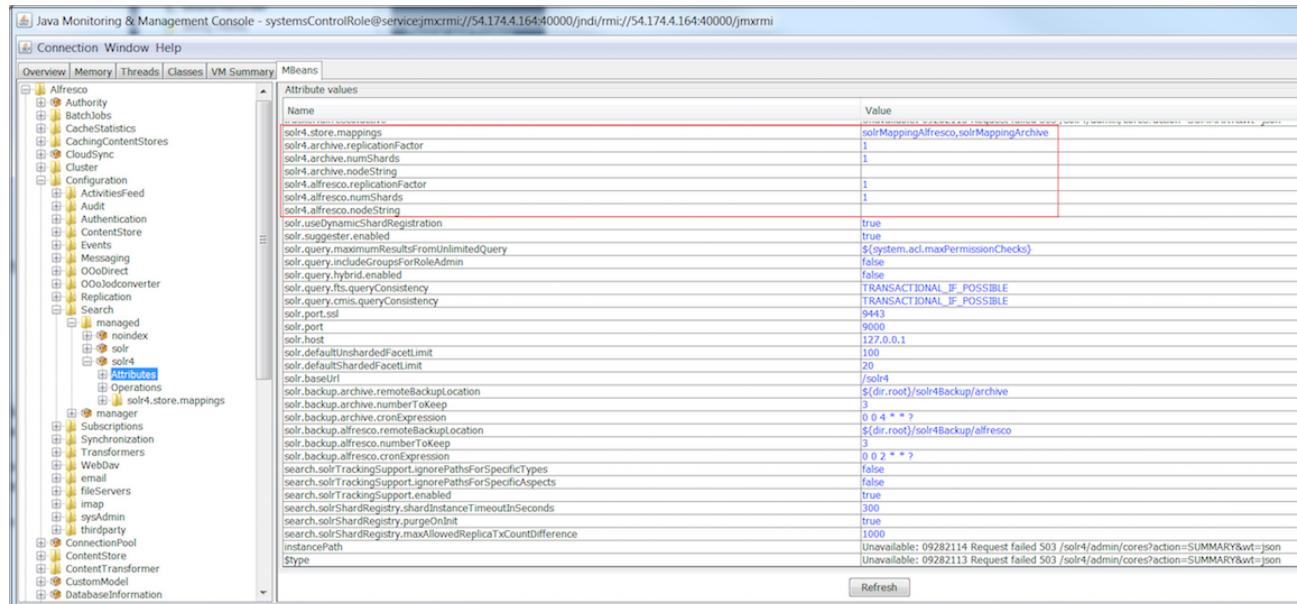
```
solr.defaultUnshardedFacetLimit=100
```

### Finding shards at query time

Use a JMX client to find shards at query time.

- In JConsole, go to **MBeans > Alfresco > Configuration > Search > managed > solr4 > Attributes**.

All the Solr attributes are listed on this page.



- Set the following properties:

```
solr4.alfresco.numShards=10
solr4.archive.numShards=10
```

- In JConsole, go to **MBeans > Alfresco > Configuration > Search > managed > solr4 > solr4.store.mappings**.
- Set numShards for `solrMappingAlfresco` and `solrMappingArchive`.

- Go to `solrMappingAlfresco > Attributes > numShards` and set the value of numShards.

```
numShards=10
```

- b. Go to `solrMappingArchive > Attributes > numShards` and set the value of `numShards`.

```
numShards=10
```

### Reindex documents by query

You can selectively reindex a small subset of the index based on a query. This enables a limited rebuild of the index.

Example 1: To reindex people after changing the first name and last name tokenisation, use the following single-threaded query:

```
http://localhost:8080/solr4/admin/cores?action=reindex&query=TYPE:person
```

Example 2: To reindex jobs that failed or threw an exception when indexing, use the following query:

```
http://localhost:8080/solr4/admin/cores?action=reindex&query=EXCEPTIONMESSAGE:*
```

You must first run the query to see how many nodes are affected. If the result is large, you can add paging as part of the query in order to reindex in smaller batches.

```
<query> AND created:"2015-08"
```

Query based reindexing is also useful when changing the property type, changing tokenisation, adding new properties to be treated as identifiers, or when reindexing synonyms.

In a sharded setup, the reindex query will have to be run on all the nodes. The query will run for all shards on any node.

### Best practices for setting up sharded Solr indexes

Use these best practices for setting up and using a sharded Alfresco installation.

- [Do I need sharding?](#)
- [Do I need dynamic shard registration?](#)
- [How many shards should I have?](#)
- [What are the reindexing recommendations for a sharded installation?](#)
- [Does sharding work with SSL enabled?](#)
- [What is the consideration for query load and number of documents?](#)
- [After upgrading, how do I use my current index while building a new sharded index?](#)
- [How do i know when this is done?](#)

#### Do I need sharding?

If you plan to store 50 million + documents in your repository, you should consider sharding to maximize indexing performance and to enable horizontal scaling to massive content repositories.

[back to top](#)

---

#### Do I need dynamic shard registration?

You can set up sharding using either manual or dynamic shard registration. Alfresco recommends that you use dynamic shard registration because it is much more easier to implement than manual sharding.

[back to top](#)

---

## How many shards should I have?

General rule of thumb is to divide the total number of documents by 50M (million). If you want to increase the query load or support more than 100 concurrent users, then check the memory specifications or the I/O specifications of the installation machine.

[back to top](#)

---

## What are the reindexing recommendations for a sharded installation?

Alfresco recommends that existing customers should reindex using the `rerank` core. This has the following benefits:

- Smaller index
- Better query performance particularly for phrases and stop words
- Improved cross-language search

This should allow the user to store anywhere between 50 million - 80 million documents in a single shard. For more information, see the [Alfresco Platform News](#) and [Alfresco 1 billion documents press release with Amazon Aurora](#).

Note that changing the number of shards requires a reindex.

[back to top](#)

---

## Does sharding work with SSL enabled?

Sharding only works if SSL is disabled. Make sure you configure the Solr and Alfresco SSL setting properly. For more information, see [Running Solr without SSL](#).

[back to top](#)

---

## Are there any considerations for query load and number of documents?

Before sharding your Solr index, it is important to consider your query load and the size of your repository. You need to create machines to host Solr. For more information, see [Installing and configuring Solr 4](#). For example, if you need 5 shards, you need to setup those 5 machines, and have Solr instances running on all the 5 machines. Once your machines are ready, you are ready to set up or register shards.

For more information, see [Dynamic shard registration](#).

[back to top](#)

---

## After upgrading, can I use my current index while building a new sharded index?

Yes. After upgrading to Alfresco One 5.1, continue to use the old search index server as before, setup a new sharded Solr server with the `rerank` template to reindex the data, and finally, switch over to the new sharded index once the indexing is done and the sharded Solr server is up-to-date.

## Upgrading from Alfresco One 5.0 and earlier versions with Solr 4 to Alfresco One 5.1 (with zero downtime)

1. Upgrade to Alfresco One 5.1 and continue to use the Solr 4 search service as before.
2. Configure a separate sharded Solr 4 index with the [rerank template](#) to track the repository. For details, see [Installing and configuring Solr shards](#).
3. While the new sharded Solr 4 builds its indexes, you can monitor the progress using the Solr Admin Web interface. For details, see [the next question](#).
4. When the sharded Solr 4 index is updated, enable the sharded Solr 4 index by setting the `solr.host` property. For more information, see [Activating Solr](#).

## Upgrading from Alfresco One 4.x and earlier versions with Lucene to Alfresco One 5.1 (search service will be unavailable while the indexes are being built)

1. Upgrade to Alfresco One 5.1 with a sharded Solr 4 installation to track the repository. Use the [rerank template](#) when configuring the new Solr core. For details, see [Installing and configuring Solr shards](#).  
 While the Solr 4 indexes are being built, you can continue to use Alfresco but the search service will not be available until the Solr 4 indexes are up-to-date.
2. Enable the sharded Solr 4 index by setting the `solr.host` property. For more information, see [Activating Solr](#).
3. While the new sharded Solr 4 builds its indexes, you can monitor the progress using the Solr Admin Web interface. For details, see [the next question](#).

## Upgrading from Alfresco One 5.0 and earlier versions with Solr 1 to Alfresco One 5.1 (with zero downtime)

1. Upgrade to Alfresco One 5.1 and continue to use the Solr 1 search service as before.
2. Configure a separate sharded Solr 4 index with the [rerank template](#) to track the repository. For details, see [Installing and configuring Solr shards](#).
3. While the new sharded Solr 4 builds its indexes, you can monitor the progress using the Solr Admin Web interface. For details, see [the next question](#).
4. When the sharded Solr 4 index is updated, enable the sharded Solr 4 index by setting the `solr.host` property. For more information, see [Activating Solr](#).

[back to top](#)

---

### How do I know the new sharded index is up-to-date?

Go to the Solr Admin Web interface at <https://localhost:8443/solr4/#/alfresco> and monitor the value of `Approx transactions remaining`. If the value is 0, it indicates that the index is up-to-date.

[back to top](#)

## Full text search configuration properties for Solr index

The Solr index's full text search properties influence the behaviour of Solr indexes.

The main index and deltas all use the same configuration. The data dictionary settings for properties determine how individual properties are indexed.

If you wish to change the default value of a property, add the relevant property to the `alfresco-global.properties` file and then make the changes.

### Solr 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 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 facetting.

**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:cmobject</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>
...
```

**Indexing options**

If you want archive or zip files to be unzipped and the files included in the index, set the following property:

```
transformer.Archive.includeContents=true
```

The default setting is false.

## Using Filtered search

Use this information for an overview of the 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 by default from Alfresco One 5.0 onwards. 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. The default filtered search properties are explained here.

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**. `displayControl` is the full module name for an Aikau widget which is used for rendering the facet filters. By default, Alfresco provides `alfresco/search/FacetFilters` which is a basic rendering of the filters available for the facet.

### **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. The `sortBy` option is passed to the `FacetFilters` widget and defines how the filters should be sorted. This property has the following options:

Option	Description
ALPHABETICALLY	Specifies the filter value A-Z.
REVERSE_ALPHABETICALLY	Specifies the filter value Z-A.
ASCENDING	Specifies the number of filter results (low to high).
DESCENDING	Specifies the number of filter results (high to low).
INDEX	This is a special value reserved for results rendered by filter queries.

**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

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 logging

You can set different debug logging levels for Alfresco-Solr components using the Solr 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 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 to re-load the properties file.

```
https://<solrHostName>:<solrPort>/solr4/admin/cores?&action=LOG4J&resource=log4j-solr.properties
```

## Calculate the memory needed for Solr nodes

Solr can have high memory requirements. You can use a formula to calculate the memory needed for the Alfresco internal data structures used in Solr for PATH queries and read permission enforcement.

By default, there are two cores in Solr: `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 100M nodes, 100M transactions, 100M ACLs and 100M ACL transactions, 20.1 GB of memory is needed. Assuming there are not many ACLs or ACL changes, for 100M 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 caches associated with one searcher

The Solr 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, 150 GB of memory is needed using the out of box configuration.

```
(512 + 1024 + 512 + 512)(500M)/8 = 150 GB
```

The default cache values needs to change to accommodate a large repository. So, for 100M documents, 100M transactions and reduced cache size, 19 GB of memory is needed.

```
(64 + 128 + 64 + 64)(500M)/8 = 19 GB
```

 The Solr memory requirement can increase exponentially depending on the search/sort being performed. The memory needed rapidly adds up in the following situations:

- Sorting a large result set
- Long running queries
- Queries with many terms
- Running multiple queries simultaneously

### Solr 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 and for capacity planning. Solr memory requirements increase with the size of the repository but also with the amount of memory you allocate to the Solr caches. Decreasing the Solr 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
```

If you are using JVM which runs Solr, you must add the following setting:

```
-XX:+UseConcMarkSweepGC -XX:+UseParNewGC
```

This includes installations where Solr is running on the same server as Alfresco and Share.

### Overall Solr memory use

This example is based on the data provided above.

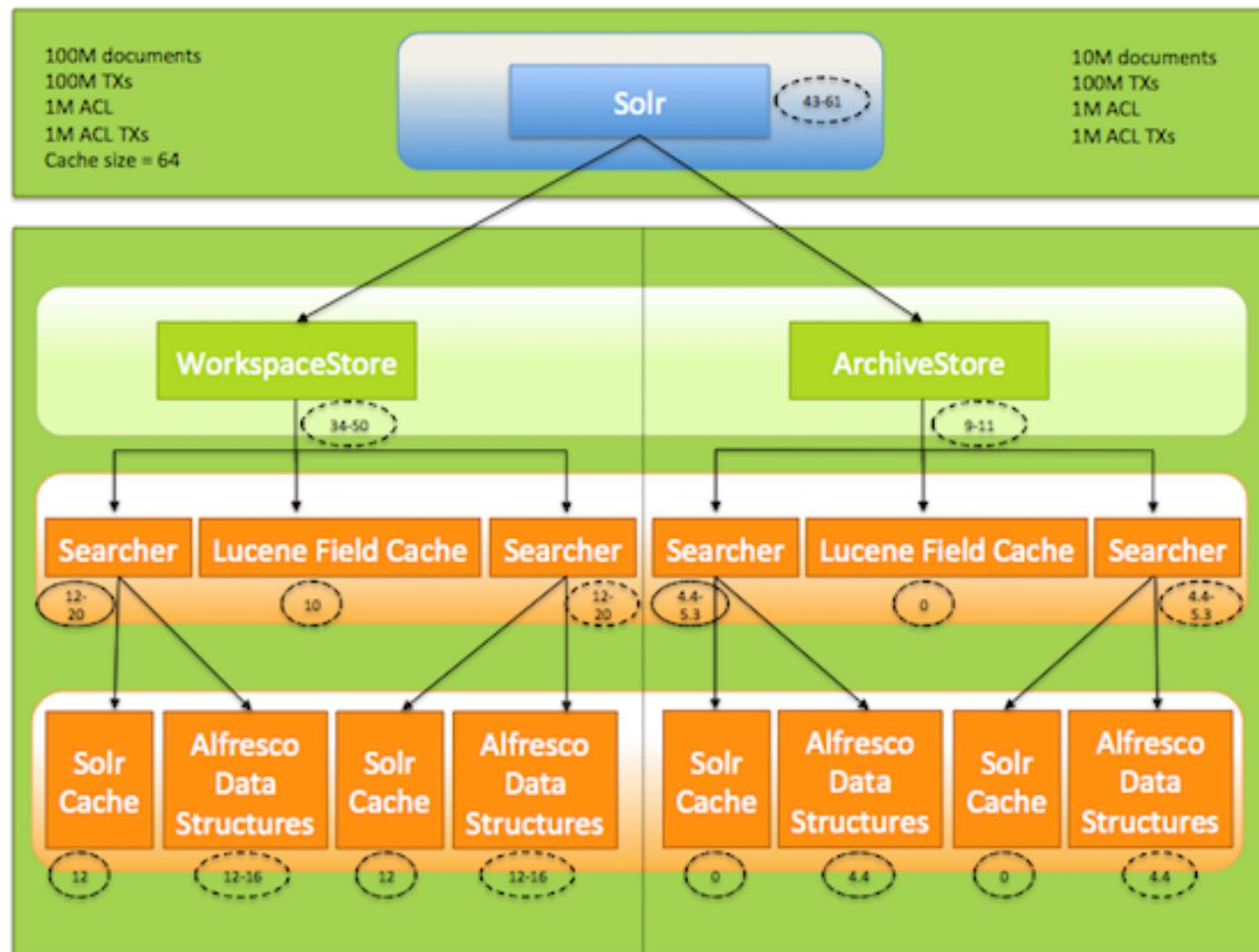
**For WorkspaceStore:** Assuming that there are 100M docs, 100M TXs, 1M ACLs and ACL TXs, cache size of 64 entries each for FilterCache, AuthorityCache and QCache, and 128 entries for

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 Archivestore:** Assuming that there are 100M transactions, 10M 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 node as explained in the example:



### Minimize the memory requirements for Solr 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

Use this information for an overview on the transactional metadata query. It 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.

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

Use this information to understand the features of the transactional metadata query.

- Transactional metadata query is supported for both Solr 4 and noindex search subsystems.
- Transactional metadata query does not support facets.
- 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.
- When queries are executed against the database:
  - Hidden nodes will be returned by the database, as they are in Alfresco One 5.0.
  - Large result sets are not supported because Alfresco does not evaluate permissions in query but as a post filter.
  - Counts will not reflect the number of nodes that match the query.
- The `SearchParameters` and `QueryOptions` objects can be used to override this behaviour per query.

### Transactional metadata queries supported by database

Use this information to understand the queries supported by the database.

The Alfresco Full Text Search (FTS) query text can be used standalone or it can be embedded in CMIS-SQL using the `contains()` predicate function. The CMIS specification supports a subset of Alfresco FTS. For more information on the Alfresco search syntax, see [Alfresco Full Text Search Reference](#) on page 676.

## 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

- double

- Supports all properties and comparisons, such as =, <>, <, <=, >=, >, IN, NOT IN
- Supports ordering for single-valued properties

- float

- Supports all properties and comparisons, such as =, <>, <, <=, >=, >, IN, NOT IN
- Supports ordering for single-valued properties

- boolean

- Supports properties and comparisons, such as = and <>
- 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 URI data type is not supported, multi-valued properties and 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



Prefix 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
- 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

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

You can configure an optional patch for upgrade.

To use or run a query against the `float`, `double`, or `boolean` property data types, you need to run an optional patch that adds the required indexes to the database. To do so, set the following property in the `alfresco-global.properties` file:

```
system.metadata-query-indexes-more.ignored=false
```

When using all other data types (such as `string`, `integer`, `id`, or `datetime`), 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 these optional patches are not 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 patches have 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.

## Configuring OpenSearch

You can configure OpenSearch to use a search engine proxy.

OpenSearch is a collection of simple formats for sharing search string results, in order to extend existing schemas such as ATOM or RSS. The list of registered search engines is in `/config/alfresco/web-scripts-config.xml`. You can configure a search engine proxy so that the OpenSearch client indirectly submits a search request through the Alfresco Web Server (the proxy), rather than directly to the search engine.

1. Create a new file called `/config/alfresco/extension/web-scripts-config-custom.xml`.

This file will contain the search engine proxy information.

2. Create a new search engine proxy, using the `proxy` attribute. For example:

```
<engine label="Alfresco Open Source Talk" proxy="opentalk">
 <url type="application/rss+xml">http://blogs.alfresco.com/opentalk/
 os-query?s={searchTerms}&itemstart={startIndex?}&itempage={startPage?}
 &itemlimit={count?}</url>
</engine>
```

The value of the `proxy` attribute must be a unique name that identifies the search engine.

3. Save `/config/alfresco/extension/web-scripts-config-custom.xml`.

## 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.getServletContext("/
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 information suggests metrics and estimates, but your system may 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).

 The disk space usage calculation above is only for content storing. It does not take into account any indexes (Lucene or Solr).

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	4.0 GB JVM RAM 4x server CPU (or 2xDual-core)
For 200 concurrent users or up to 2000 casual users	8.0 GB JVM RAM 8x server CPU (or 4xDual-core)

 For full performance tuning, contact Alfresco Support or Alfresco Consulting.

## 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
-Xms1G
-Xmx2G
-Dcom.sun.management.jmxremote
```

If you are using JVM which runs Solr 4, you must add the following setting:

```
-XX:+UseConcMarkSweepGC -XX:+UseParNewGC
```

This includes installations where Solr 4 is running on the same server as Alfresco and Share.

 You do not need to set the `-XX:MaxPermSize` parameter if using Java 8. If you are using Java 7, this parameter **must** be set.

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 LibreOffice 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 here.**

## Permanent Generation (PermGen) Size

The default PermGen size in Oracle 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 Oracle 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
```

## Setting debug mode for troubleshooting

To debug your JVM server:

- If you are a Linux user, edit the JVM options used to start the Alfresco Tomcat instance, set by the `tomcat/scripts/ctl.sh` script. See [Controlling JVM system properties](#) on page 345 for detailed information.

For example, set the following:

```
JAVA_OPTS=%JAVA_OPTS% -server -Xdebug -
Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=8082
```

where `address` is a port for your system.

- If you are a Windows user, register Tomcat as a Windows service:
  - In the installation directory, locate the `properties.ini` file and copy the value of the `tomcat_unique_service_name` parameter (for example, `alfrescoTomcatnum1`).
  - From the `/tomcat/bin` directory, run the following command at a command prompt:
 

```
tomcat7w.exe //ES//<alfrescoTomcatnum1>
```

 where `<alfrescoTomcatnum1>` is the value from your `tomcat_unique_service_name` parameter.
  - Open the `alfrescoTomcatnum1` Properties window, select the **Java** tab, and the **Java Options** field and add the following lines of code on two separate lines:

```
-Xdebug
-Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=8000
```

where `address` is a port for your system.

### 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

There are 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
http.proxyUser	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.
http.proxyPassword	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.
https.proxyUser	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.
https.proxyPassword	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.

## Configuring server administration properties

The sysAdmin subsystem allows real time control across some of the general repository properties. The sysAdmin subsystem replaces the `RepoServerMgmt` management bean.

### Configuring server administration settings

1. Open the Admin Console.
  2. In the **General** section, click **System Settings**.
- You see the **System Settings** page showing the details of your Alfresco installation.
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 344.

Alfresco Repository Settings property	Example setting	What is it?
<b>Repository Context</b>	alfresco	This property specifies the context path of the Alfresco web application URL. The default value is <code>alfresco</code> . 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 <code>alfresco</code> is the name of the proxy server or specific server that you are using.

Alfresco Repository Settings property	Example setting	What is it?
<b>Repository Hostname</b>	<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.
<b>Server Allow Writes</b>	true	Write access is permitted to the repository, as long as the Alfresco license is valid. When this property is set to false, the repository is in read-only mode.
<b>Protocol</b>	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.
<b>Port</b>	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?
<b>Allowed Users</b>		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.
<b>Maximum Users</b>	-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?
<b>Share Context</b>	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.
<b>Protocol</b>	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.
<b>Share Hostname</b>	127.0.0.1	This property sets the externally resolvable host name of the Share web application URL. The default value is \${localname}.
<b>Port</b>	8080	This property sets the externally resolvable port number of the Alfresco web application URL. The default is 8080.
<b>Site Public Group</b>	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**.

### 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.allowWrite

A Boolean property that when true indicates that the repository will allow write operations (provided that the license is valid). Set this property to false to put the repository in to 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 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.

## Controlling JVM system properties

Use these techniques 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](#) on page 79.

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, 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 security](#) on page 281.

### Configuring SSL for a production environment

Several proxy application servers are available to configure for SSL communications; for example, Apache HTTP Server, HAProxy or Nginx. Using a proxy server means that you can do not have to edit your Alfresco configuration files directly.

A client machine connects to the proxy server instead of the application server directly. As a result, Share and Alfresco must be configured to use an externally available URL, rather than an internal machine name. This external URL can then be passed to other parts of the Share application; for example, when Share creates a link to Alfresco Office Services for online editing.

For security reasons, configure your proxy to forward only requests to the resources that you want to expose to the outside world. In this scenario, the applications need to use the internal machine name when talking to each other but use the external name when creating links for the user.

1. Set your proxy to forward the following URL extensions to Alfresco:

```
/share
/share/*
/alfresco/api/*public/cmis/versions/*
/alfresco/api/*public/alfresco/versions/*
/alfresco/api/cmis/versions/*
/alfresco/service/api/server
/alfresco/cmisatom/*
/alfresco/service/cmis/*
```

2. If you are using WebDAV, add these URL extensions to your proxy:

```
/alfresco/webdav
/alfresco/webdav/*
```

3. For Alfresco Office Services, add these URL extensions to your proxy:

```
/_vti_inf.html
/_vti_bin/*
/alfresco/aos
/alfresco/aos/*
```

and for OPTIONS and PROPFIND requests:

```
/
```

4. Block requests with these URL patterns:

```
/share/*proxy/alfresco/api/solr/*
/share/-default-/proxy/alfresco/api/*
```

The communication between Solr and Alfresco is, by default, protected by SSL. These patterns need to be explicitly blocked to protect the API endpoints.

5. Edit the `alfresco-global.properties` file with these values:

```
alfresco.context=alfresco
alfresco.host=<external-proxy-host-name>
alfresco.port=443
alfresco.protocol=https
share.context=share
share.host=<external-proxy-host-name>
share.port=443
share.protocol=https
opencmis.context.override=true
opencmis.context.value=
opencmis.servletpath.override=true
opencmis.servletpath.value=
opencmis.server.override=true
opencmis.server.value=https://<external-proxy-host-name>
```

6. If you are using a proxy server other than Apache with AJP, follow these steps:

- Add this line to your `alfresco-global.properties` file:

```
aos.baseUrlOverwrite=https://<external-proxy-host-name>/alfresco/-aos
```

- Configure proxy redirect responses sent by the application server.

The server behind the proxy uses the `http` schema because it is not aware of SSL.

Here is an example of an nginx configuration:

```
proxy_redirect http://example.com/alfresco/ https://example.com/
alfresco/;
```

 The `proxy_redirect` configuration shows the change from `http` to `https`.

7. Use the following sample `httpd.conf` configuration file for Apache HTTP Server:

```

General

ServerName yourserver.example.com
PidFile /path/to/your/http.pid
ErrorLog /path/to/your/apache/log/error_log
LogLevel info

LoadModule unixd_module /path/to/your/apache/modules/mod_unixd.so
LoadModule authn_core_module /path/to/your/apache/modules/
mod_authn_core.so
LoadModule authz_host_module /path/to/your/apache/modules/
mod_authz_host.so
LoadModule authz_core_module /path/to/your/apache/modules/
mod_authz_core.so
LoadModule rewrite_module /path/to/your/apache/modules/
mod_rewrite.so

<IfModule unixd_module>
 User _www
 Group _www
</IfModule>

Block API requests

LoadModule rewrite_module /path/to/your/apache/modules/
mod_rewrite.so
RewriteEngine on
RewriteBase /
RewriteRule ^/share/(.*)/proxy/alfresco/api/solr/(.*)$ - [F]
RewriteRule ^/share/-default-/proxy/alfresco/api/(.*)$ - [F]
```

```

Proxy

LoadModule jk_module /path/to/your/apache/modules/mod_jk.so
JkWorkersFile /path/to/your/workers.properties
JkLogFile /path/to/your/apache/log/mod_jk.log
JkLogLevel info
JkShmFile /path/to/your/apache/log/jk-runtime-status

SSL

LoadModule ssl_module /path/to/your/apache/modules/mod_ssl.so
Listen 443
<VirtualHost *:443>
 SSLEngine on
 SSLProtocol all -SSLv2
 SSLCipherSuite HIGH:!aNULL:!MD5
 SSLVerifyClient none
 SSLCertificateFile /path/to/your/certificate.pem
 ErrorLog /path/to/your/apache/log/ssl_error_log
 LogLevel warn
 JkMount /share alfresco-worker
 JkMount /share/* alfresco-worker
 JkMount /alfresco/webdav alfresco-worker
 JkMount /alfresco/webdav/* alfresco-worker
 JkMount / alfresco-worker
 JkMount /_vti_inf.html alfresco-worker
 JkMount /_vti_bin/* alfresco-worker
 JkMount /alfresco alfresco-worker
 JkMount /alfresco/ alfresco-worker
 JkMount /alfresco-aos alfresco-worker
 JkMount /alfresco-aos/* alfresco-worker
 # Un-comment these lines for public API access
 # JkMount /alfresco/api/*/public/cmis/versions/* alfresco-worker
 # JkMount /alfresco/api/*/public/alfresco/versions/* alfresco-worker
 # JkMount /alfresco/service/api/server alfresco-worker
 # JkMount /alfresco/cmismat/* alfresco-worker
 # JkMount /alfresco/service/cmis/* alfresco-worker
 # JkMount /alfresco/api/cmis/versions/* alfresco-worker
 # Un-comment these lines for Desktop Sync
 # JkMount /alfresco/api/*/private/alfresco/versions/* alfresco-worker
</VirtualHost>
```

This configuration file has been tested with Apache httpd 2.4. Replace the values in **bold** font with the file names and directories that are relevant to your system.

8. Use the following sample `worker.properties` configuration file for Apache HTTP Server:

```

worker.list=alfresco-worker
worker.alfresco-worker.port=8009
worker.alfresco-worker.host=your-internal-alfresco-host-name
worker.alfresco-worker.type=ajp13
worker.alfresco-worker.lbfactor=1
```

In this example, Apache is configured to accept strong encryption only. Adapt `SSLCipherSuite` if this causes you problems.

## Configuring SSL for a test environment



These instructions should only be used for configuring a test environment. If you are configuring a production environment you should use a proxy server to handle all SSL communication. See [Configuring SSL for a production environment](#) on page 346 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"
sslEnabledProtocols="TLSv1.2,TLSv1.1,SSLv1,SSLv2Hello" />
```

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"
sslEnabledProtocols="TLSv1.2,TLSv1.1,SSLv1,SSLv2Hello" />
```

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"
sslEnabledProtocols="TLSv1.2,TLSv1.1,TLSv1,SSLv2Hello" />

```

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, `hz.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.

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>

```

An example file is provided in `<extension>/mimetype/mimetypes-extension-map.xml.sample`. You can include multiple files and each one is loaded automatically.

3. Save the file.
4. Restart Alfresco.

The MIME type is available in the repository.

## 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 can have properties that, when added, can enhance the content types. You can also 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 <b>Categories</b> property in the document properties.
Complianceable	This aspect is no longer valid. For compliance-related behavior, use the Record Management module.	

Aspects	Description	Changes in Behavior/Share Interface
Dublin Core	Enables metadata (such as publisher, contributor, identifier) to be added to a content item.	<p>Adding Dublin Core aspect displays the following additional metadata properties in the document properties:</p> <ul style="list-style-type: none"> <li>• Publisher</li> <li>• Contributor</li> <li>• Type</li> <li>• Identifier</li> <li>• Source</li> <li>• Coverage</li> <li>• Rights</li> <li>• Subject</li> </ul>
Effectivity	This aspect is no longer valid. For compliance-related behavior, use the Record Management module.	
Summarizable	Enables addition of a brief description about the content item.	Adding Summarizable aspect displays additional <b>Summary</b> property in the document properties.
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 <b>Version History</b> .
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) in the document properties.
Inline Editable	Enables content items to be edited directly in Alfresco.	Adding Inline Editable aspect displays the <b>Edit in Alfresco</b> link in the document properties.
Taggable	<p>Enables tagging of content items using keywords. In AlfrescoShare, content items are taggable by default.</p>	Adding Taggable aspect displays the tagged keywords in the <b>Tags</b> 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 <b>Latitude</b> and <b>Longitude</b> properties on the <b>Edit Properties</b> page. Also, the <b>View on Google Maps</b> link is displayed in the <b>Document Actions</b> .
EXIF	<p>Enables capturing and viewing of additional image-related metadata of a content item.</p> <p> This aspect is automatically applied to an image content item.</p>	Adding EXIF aspect displays additional information (such as Camera Model, Camera Software, Resolution Unit) about the image in the document properties.

Aspects	Description	Changes in Behavior/Share Interface
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 document properties.
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 document properties.

## 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.

💡 Any properties that you set on a file are saved with the current version of a file, and written to the Version History after a major update; for example, when a new file is uploaded. This means that if you save properties in version 1.0, they are saved in the Version History of version 1.1.

To change this behavior, you can set `cm:autoVersionOnUpdateProps` to `true`. See [VersionHistoryNode API](#) for more information.

### Making all content versionable

1. Download the [contentModel.xml](#) file.
2. Create a `$TOMCAT_HOME/shared/classes/alfresco/extension/models` directory.
3. In the `contentModel.xml` file, search for `<type name="cm:content">`, and immediately after the closing `</properties>` tag, insert the following lines to make the content versionable:

```
<mandatory-aspects>
 <aspect>cm:versionable</aspect>
</mandatory-aspects>
```

4. Copy the edited `contentModel.xml` file to the `$TOMCAT_HOME/shared/classes/alfresco/extension/models` directory.
5. Add a Spring context file to `$TOMCAT_HOME/shared/classes/alfresco/extension` with the following lines:

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE beans PUBLIC '-//SPRING//DTD BEAN//EN' 'http://www.springframework.org/dtd/spring-beans.dtd'>

<beans>
 <bean parent="dictionaryModelBootstrap" depends-on="dictionaryBootstrap">
```

```

<property name="models">
 <list>
 <value>alfresco/extension/models/contentModel.xml</value>
 </list>
</property>
</bean>
</beans>

```

6. Save the file.
7. Restart the Alfresco server.

Uploaded content will then have the `cm:versionable` aspect.

### Disabling the auto-versioning feature

1. Open the `alfresco-global.properties` file.
2. Add the following property:

```
version.store.enableAutoVersioning=false
```

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.
2. Grant the slave permission to replicate:

```
GRANT REPLICATION SLAVE ON *.* TO <slave_user> IDENTIFIED BY
'<slave_password>'
```

3. 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.

4. 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.

5. 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.

1. Download the [content-services-context.xml](#) file:
2. Paste this file into the `<extension>` directory.
3. Open the file.

Transformers start below the comment:

```
<!-- Content Transformations -->
```

4. 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.)
5. Delete every pair of `<bean> </bean>` tags except the pair containing the similar transformer.
6. Rename and modify the bean.
7. 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 134.
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 Share

---

You can use configuration files to customize Alfresco Share, in order to meet your organization's requirements.

For more information on customizing Share, see [Configuring Alfresco Share](#).

## 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.

### 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?
<b>File System Name</b>	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?
<b>CIFS Enabled</b>	Yes	This enables or disables the CIFS server.
<b>Server Name</b>	<code> \${localname}A</code>	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 <code> \${localname}</code> in place of the local server's host name and generate a unique name by prepending/appending to it.
<b>Host Announce</b>	Yes	Enables the announcement of the CIFS server to the local domain/workgroup so that it shows up in the Network Places/Network Neighborhood.

CIFS property	Example setting	What is it?
<b>Session Timeout (seconds)</b>	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.
<b>Domain</b>		The domain or workgroup to which the server belongs. If not specified then the domain/workgroup of the server is used.

5. Set the FTP properties:

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

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

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

## 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. If Alfresco is installed on a Windows server, the name of the machine must not exceed 14 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. The combined \${localname} value must not exceed 15 characters.

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. If you are running on Mac OS X 10.10 (Yosemite) or later, set up the `pf` firewall to forward to the non-privileged TCP 1139/1445 ports. You will need admin rights to perform these actions:

- a. In the `/etc` directory, locate the `pf.conf` file and the `pf.anchors` folder.

Take a copy of the `pf.conf` file and rename it as `pf-alfresco-cifs.conf`.

- b. Add the following code to the `pf-alfresco-cifs.conf` file:

```
rdr-anchor "alfresco-forwarding"
load anchor "alfresco-forwarding" from "/etc/pf.anchors/
alfresco.cifs.forwarding"
```

- c. Create a new file in the `/etc/pf.anchors` folder called `alfresco.cifs.forwarding` and add the following code:

```
rdr pass on en0 inet proto tcp from any to any port 445 -> 127.0.0.1
port 1445
rdr pass on en0 inet proto tcp from any to any port 139 -> 127.0.0.1
port 1139
```

- d. Add the following code to the end of the `pf.conf` file:

```
rdr-anchor "alfresco-forwarding"
load anchor "alfresco-forwarding" from "/etc/pf.anchors/
alfresco.cifs.forwarding"
```

- e. Enable port forwarding using this command:

```
pfctl -ef /etc/pf-alfresco-cifs.conf
```

2. For other platforms, 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.

Set up the firewall rules to forward requests:

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

3. On Mac OS X 10.9 and earlier, use these commands:

```
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.

4. On Linux, you can use the following commands 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` 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
Pkttype	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.fileservice=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 **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](#)

Use this 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](#)

Use this information to configure the FTP file server.

For more information about configuring the FTP file server using the Admin Console, see [Enabling file servers](#) on page 358.

### [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.fileservice=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, PKTTYPE, 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.

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 email

Use this information to configure email services, including inbound and outbound email, subscriptions, and email clients.

### Configuring inbound and outbound 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)

#### Managing inbound emails

Set these inbound email properties in the Admin Console 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?
<b>Enabled</b>	No	Use check box to enable or disable the inbound email service. By default, it is not enabled.
<b>Unknown User</b>	anonymous	This is the user name to authenticate as when the sender address is not recognized.
<b>Allowed Senders</b>	.*	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.
<b>Overwrite Duplicates</b>	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.
<b>Maximum Server Connections</b>	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.
<b>SMTP Authentication Enabled</b>	No	Use this check box to enable or disable the authentication of inbound email against the repository.
<b>Email Server Port</b>	25	This is the default port number for the email server.
<b>Email Server Domain</b>	alfresco.com	This is the default domain for the email server.
<b>Blocked Senders</b>		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.
<b>Email Authentication Group</b>	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?
<b>Transport Layer Security (TLS)</b>	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"> <li>• <b>Disabled:</b> TLS support is disabled</li> <li>• <b>Hidden:</b> On the EHLO command, server support for TLS is hidden, though TLS will still be accepted if the client uses it</li> <li>• <b>Enabled:</b> On the EHLO command, server support for TLS is announced</li> <li>• <b>Required:</b> TLS authentication is required</li> </ul>

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

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

### Inbound SMTP configuration properties

The Inbound SMTP 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.`

**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?
<b>Hostname</b>	smtp.example.com	This is the name of the SMTP(S) host server.
<b>Encoding</b>	UTF-8	This is the email encoding type. The default is UTF-8.
<b>Editable Sender Address</b>		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.
<b>Email Server Port</b>	25	This is the default port number for the email server.
<b>Default Sender's Address</b>	alfresco@demo.alfresco.org	The default address that is used in the From field of outbound emails if no alternative is available.
<b>Email Protocol</b>	SMTP	Select a protocol from the list. This is the protocol that will be used when sending email.
<b>Username</b>	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.
<b>Password</b>		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**.

## Outbound SMTP configuration properties

The following properties can be configured for the Outbound SMTP 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 in the Outbound SMTP 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 Outbound SMTP 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 `smtpls`.

### **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 SMTPLS 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=smtpls
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
```

## Handling messages by target node type

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 the 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?
<b>Activity Feed Enabled</b>	Yes	This enables or disables activity notifications to users using email.
<b>Frequency CRON Expression</b>	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.
<b>Maximum Number</b>	200	The maximum number of activities that are reported on in the Activities dashlets and Activities Feed emails.
<b>Maximum Age (mins)</b>	44640	This is the maximum age of the activities shown in the Activities Feed emails. Activities that are older than the maximum age are not shown in the Activities dashlet. 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**.

## Enabling the Subscription Service

When users are being followed, the person(s) following them receive activity notifications. The Subscription Service is the underlying service used to manage and generate activity notifications. You can use this page to enable/disable the Subscription Service and the follow feature on a system wide basis.

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**.

## Configuring the email client with IMAP

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 in Alfresco, for example, they have their own INBOX. Users can manage emails in 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.

 Be careful when deciding what mount points you provide. When an IMAP client mounts a mount point, it issues a `LSUB " " *` command. This retrieves the entire tree of folders below the mount point.

For information about working with Alfresco and Microsoft Outlook, see [Installing and configuring Alfresco Outlook Integration](#) on page 87.

### Enabling the IMAP protocol using `alfresco-global.properties`

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.

1. Open the `alfresco-global.properties` file.
2. Enable the IMAP server by setting the following property to `true`:

```
imap.server.enabled=true
```

3. 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.

4. 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.

## Enabling the IMAP Service using the Admin Console

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

IMAP Service property	Example setting	What is it?
<b>IMAP Server Enabled</b>	No	This enables or disables the IMAP server.
<b>Hostname</b>	0.0.0.0	This specifies the host or IP address to which the IMAP service will bind.
<b>Mail TO Default</b>	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.
<b>Mail FROM Default</b>	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.

4. Set the IMAP Protocol properties:

IMAP Protocol property	Example setting	What is it?
<b>Enable IMAP</b>	Yes	This enables or disables the IMAP service.
<b>Port</b>	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?
<b>Enable IMAP</b>	Yes	This enables or disables the IMAPS service.
<b>Port</b>	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**.

## 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](#) on page 625 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.

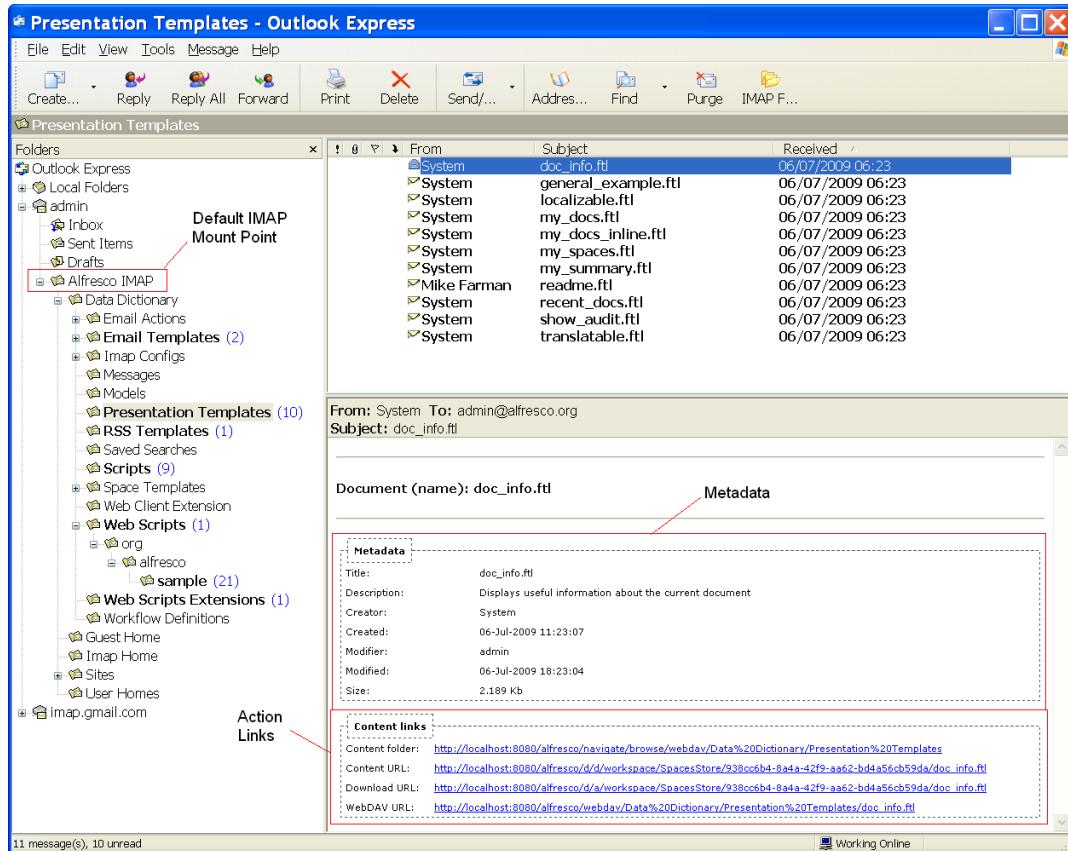
By default, a single mount point called **AlfrescoIMAP** is defined for **Company Home** and you can change it or add more mount points.

-  Be careful when deciding what mount points you provide. When an IMAP client mounts a mount point, it issues a `LSUB " " *` command. This retrieves the entire tree of folders below the mount point.

### 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

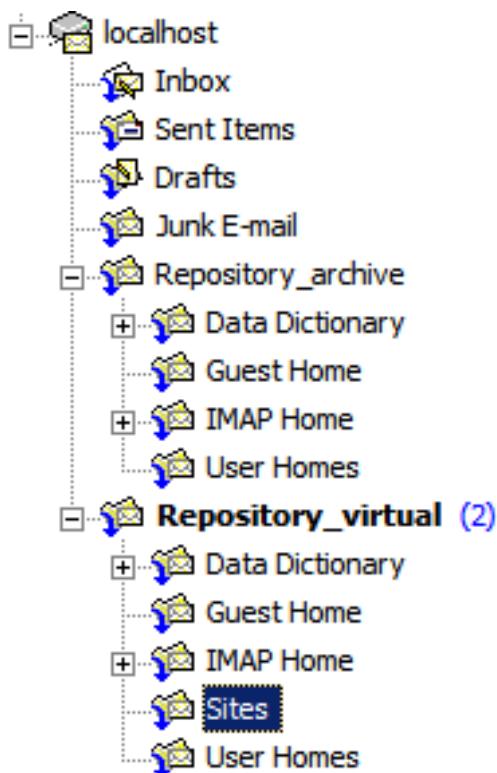
To have access to Alfresco Share sites using IMAP, the site(s) need to be added to your list of sites using Share IMAP Favorites.

1. Select **IMAP Favorites** in the Share **My Sites** dashlet on your **My Dashboard** page:

The screenshot shows the 'My Sites' dashboard. It features a 'Create Site' section with a 'Content Community' site listed. The site has a yellow star icon next to its name. A red box highlights this star icon. Below the site list, a callout box with a red border and arrow points to the star icon with the text 'Click the IMAP Favorite to make the site available via IMAP'.



2. Refresh your IMAP view to see the new sites.



You can see the site added to the IMAP Sites folder.

If the folders do not appear in your email client, you should confirm that:

- The folder is in a site marked as an **IMAP favorite**.
- Your email client is showing all folders, and not just the folders that you have subscribed to.
- In your email client, look for a property like **Reload IMAP folders**.
- In your email client, clear your IMAP cache.

## Configuring LibreOffice

You can transform a document from one format to another using the LibreOffice subsystem. This feature requires you to install LibreOffice.

### OOoJODconverter

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

- 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 (this is a workaround for LibreOffice memory leaks)

### OOoDirect

If you are using OpenOffice in place of LibreOffice, use the OOoDirect subsystem for OpenOffice integration. 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 LibreOffice transformations is OOoJodconverter. Alfresco also supports the OOoDirect subsystem for OpenOffice.

You can change the OOoJodConverter and OOoDirect subsystems using the following ways:

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

### Alfresco 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?
<b>JODConverter Enabled</b>	No	This enables or disables the JODConverter for transformations.
<b>Max Tasks per Process</b>	200	This is the maximum number of tasks that can be performed concurrently.

Property	Example setting	What is it?
<b>Office Suite Location</b>	/Applications/alfresco-5.0.0/libreoffice.app/Contents	This shows the directory path locations of OpenOffice.org or LibreOffice.
<b>Port Numbers</b>	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.
<b>Task Execution Timeout</b>	120000	This is the duration in milliseconds after which a task will timeout.
<b>Task Queue Timeout</b>	30000	This is the duration in milliseconds after which 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**.

### 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.

## LibreOffice configuration properties

LibreOffice uses the OOoJodConverter subsystem. Configure the following properties for the OOoJodconverter subsystem.

#### **jodconverter.connectTimeout**

Specifies the maximum number of milliseconds before a connection times out. The default is 10000 milliseconds (10 seconds).

#### **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 OpenOffice transformations in place of LibreOffice

LibreOffice is used in preference to OpenOffice in Alfresco. Use this information if you need to configure OpenOffice transformations specifically.

1. Open the `alfresco-global.properties` file.
2. Set the `ooo.exe` property to the path of the OpenOffice 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 OpenOffice installation path.

**ooo.enabled**

Enables or disables the OOoDirect subsystem.

## Configuring Smart Folders

Stored searches are shown in a folder tree, so that when a user opens a folder, a query is run and the results are displayed in a list. Files are also automatically classified when they are uploaded.

Smart Folders are installed as a core part of Alfresco, so there is no separate AMP file to install or upgrade. The Smart Folders function is disabled by default, and can be enabled in your `alfresco-global.properties` file by specifying `smart.folders.enabled=true`.

Folders are differentiated by icon:

- Physical folder: 
- Smart folder: 

Using Smart Folders in this way helps you to manage your information; for example, where you have a number of sources of information, in a variety of folders. Content that might be related to, but not directly involved in your work is also retrieved, depending on the search criteria.

The Smart Folder structure is created by associating a Smart Folder Template with an Alfresco physical folder. Multiple Smart Folder structures can be defined in a single template. For every Smart Folder, the template defines a folder name, search, and filing criteria, along with other properties. New templates are typically defined and added by business analysts, and created by administrators.

The folder structure can be personalised by user, for example, if you create a folder called `My Files`, you can populate it with files relevant to each user.

Take a look at the videos to learn more: [Smart Folders videos](#)

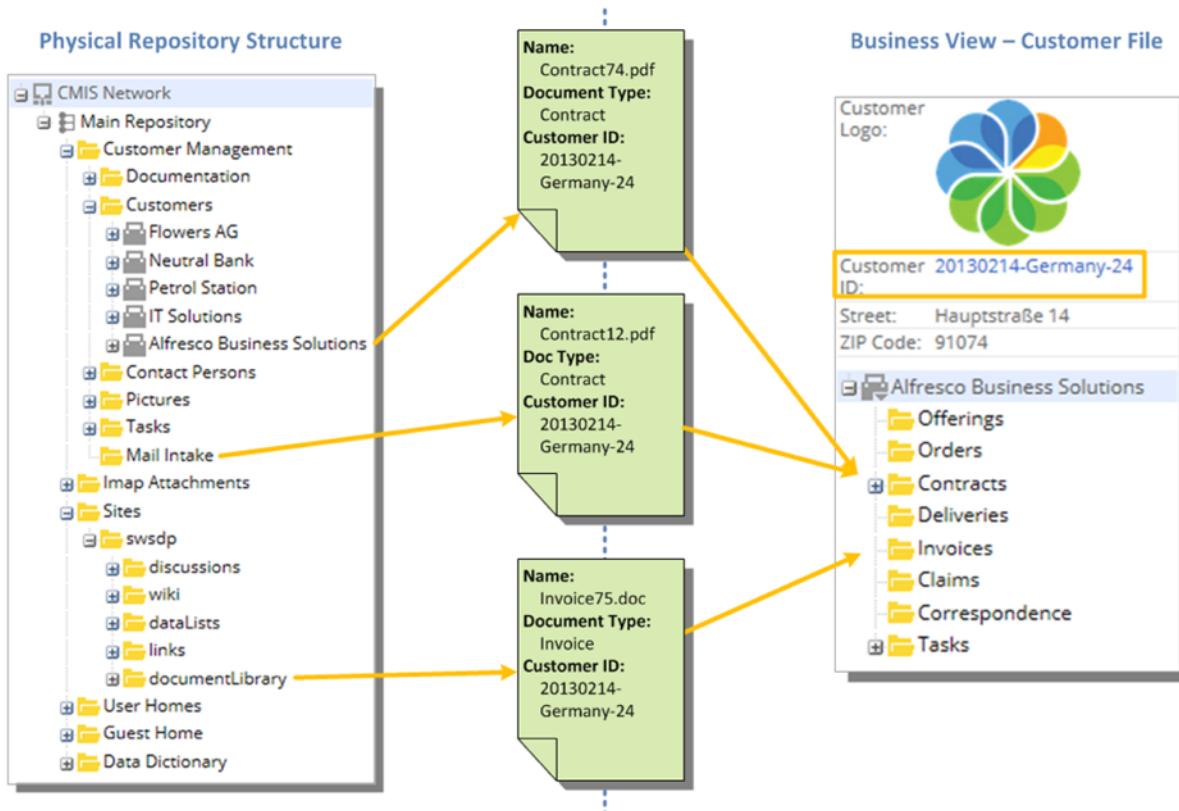
## What is a Smart Folder?

Use this information to understand the structure of Smart Folders.

This information is primarily aimed at business analysts, and system administrators.

A Smart Folder displays the results of a query in a folder format. It is “smart”, because there is no physical folder to represent it in the repository and the results are created dynamically. For example, a Smart Folder called `My video files` might be created to contain all files that I created that have a video format. Every time I open the `My video files` folder, the search query is run, and all my video files are available in that folder, wherever in the repository I have created them.

The diagram shows a physical file system, and how a Smart Folder structure is created to contain files relevant to a particular customer:



Smart Folders are created when a Smart Folder Template is run. The Smart Folder Template contains:

- A folder name
- The query to be executed, when the folder is accessed by a user

- An optional filing rule, so that a user can add a file to the Smart Folder (and the file is filed according to the query for that folder)
- An optional list of properties that can be inherited by files or used for value propagation

Smart Folders have a limited set of actions:

- Add/ Create: You can add files to a Smart Folder. The file is put into a physical folder, as specified by the filing rule.
- Update: You can update files in a Smart Folder. Updating a property might result in a file being removed from the current Smart Folder (because it no longer meets the query criteria).
- Delete, Edit Properties, Unzip To, Sync, Locate To, Move, and Copy actions for files are not supported.

The Smart Folder itself can't be edited in Alfresco, except through the Smart Folder Template. For more information about Smart Folder Templates, see [Applying a Smart Folder Template](#).

Physical folders can be displayed inside Smart Folders as long as the physical folder matches the query criteria.

## Smart Folders terminology

Special terms used to describe Smart Folders.

### Filing rule

A filing rule is specified in a Smart Folder Template and defines where a new file is stored in the repository, when it is uploaded to a Smart Folder. The filing rule also specifies the type and aspects that are applied to the new file, along with its property values.

### Smart Folder

A Smart Folder displays the results of a query in a folder format. It is “smart”, because there is no physical folder to represent it in the repository and the results are created dynamically. A Smart Folder can also contain a hierarchy of Smart Folders.

### Smart Folder Template

A Smart Folder Template is a JSON file that is stored in Alfresco in `Repository/Data Dictionary/Smart Folder Templates`. When the template is run in a physical folder, a Smart Folder structure is created.

## Prerequisites for using Smart Folders

There are a number of prerequisites for using Smart Folders in Alfresco.

### Alfresco requirements

- Smart Folders are provided as part of the standard installation with Alfresco One 5.1 (and later versions), and Alfresco Community Edition. Smart Folders are not available with Alfresco in the Cloud.
- You must enable Smart Folders in your `<tomcat>/shared/classes/alfresco-global.properties` file:

```
smart.folders.enabled=true
```

- To define a query for a Smart Folder, Alfresco Full Text Search (AFTS) must be used.
- Ensure that your system administrator has configured Alfresco to use Solr 4 as a search service and that Alfresco Full Text Search is configured to either Always use Database or Use Database if possible.

Ensure that your business analyst has considered the business case for enabling Smart Folders. See [Planning and implementing Smart Folders](#) on page 386 for more information.

## Planning and implementing Smart Folders

Consider the business requirements for creating Smart Folders in Alfresco.

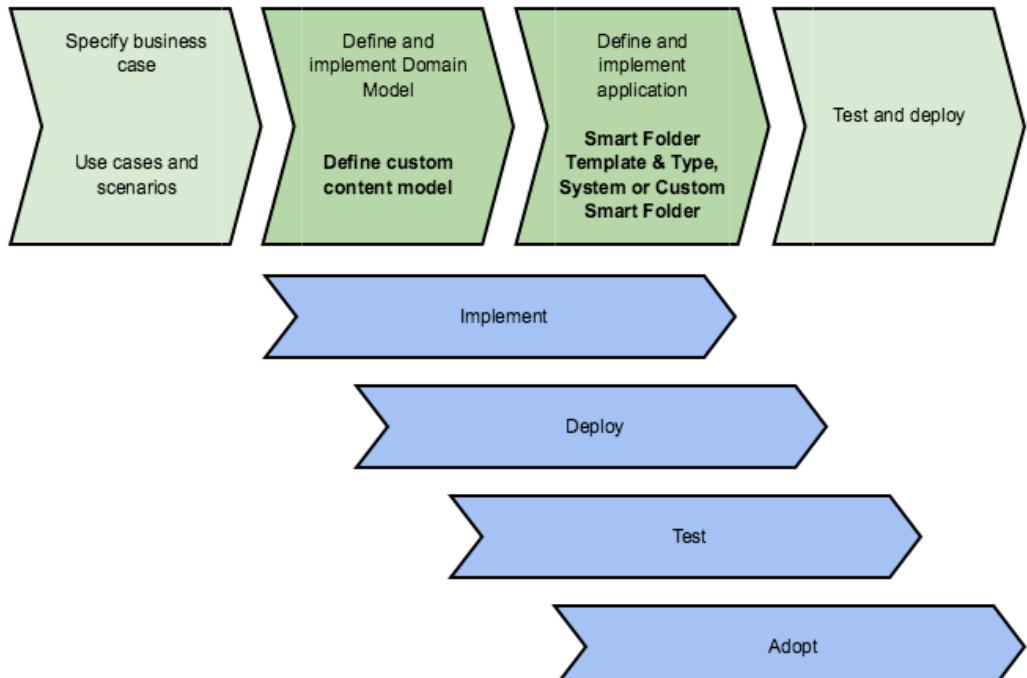
This information is primarily for business analysts, who are responsible for creating and defining the business scenario that requires Smart Folders.

Before you use Smart Folders, consider the use cases and scenarios that are appropriate to your business problem. Then you can:

- Define a custom content model. To get you started, you can use the example model that is provided with the Smart Folders tutorial. See [Smart Folders tutorial](#) on page 387 for more information. For more information on content models in Alfresco, see [Content modeling with Model Manager](#) on page 406.
- Create a Smart Folder Template, which defines the queries and filing rule for your Smart Folder structure, and property propagation rules for file uploads. The Smart Folder Template is a JSON file. See [Smart Folder Template syntax](#) on page 397 for more information.
- Choose Type-based, System, or Custom Smart Folders to associate a Smart Folder Template with a physical repository folder. See [Type-based, System, and Custom Smart Folders](#) on page 395 for more information.
- In an advanced setup, you might need to:
  - Enable Share actions in the `share-config-custom.xml` file. See [Configuring Share Actions with Smart Folders](#) for more information.
  - Configure other Smart Folders properties in the `alfresco-global.properties` file. See [Smart Folders global properties settings](#) on page 401 for more information.

You can then test and deploy your solution. Use the [Smart Folders tutorial](#) to understand more about the basic Smart Folders setup.

This diagram shows the recommended



workflow:

## Enabling Smart Folders

As an admin user, you must enable Smart Folders, and specify a Smart Folder Template for use.

A predefined template is available by selecting the **System Smart Folder** aspect in Alfresco. You can add other customized templates, and if they are uploaded to Repository/Data Dictionary/Smart Folder Templates, they are then available by selecting the **System Smart Folder** aspect. If you store templates anywhere else in your repository, you can use them by selecting the **Custom Smart Folder** aspect.

1. Stop Alfresco, and edit your <tomcat>/shared/classes/alfresco-global.properties file to enable Smart Folders:

```
smart.folders.enabled=true
```

Advanced Smart Folders settings are provided in the <tomcat>/shared/classes/alfresco-global.properties.sample file.

2. Restart Alfresco.
3. In Alfresco, select the Repository/Data Dictionary/Smart Folder Templates directory.

Select the Smart Folders Template that is provided: smartFoldersExample.json or your own customized template.

If you use your own template, make sure that you change the type to **Smart Folder Template**. See [Applying multiple templates](#) for more information.

If you store templates anywhere else in your repository, navigate to the template and select it. You can use them later by selecting the **Custom Smart Folder** aspect.

There is no need to restart Alfresco. When you edit properties on nodes that have the **Custom Smart Folder** aspect applied, the new Smart Folder is included in the **Smart Folder Template** menu. See [Applying a Smart Folder Template](#) for more information.

If you need to customize the template, see [Applying a Smart Folder Template](#) for information on the sample file structure, and [Smart Folder Template syntax](#) on page 397 for guidance on the Smart Folder Template JSON format.

## Smart Folders tutorial

In this seven-step tutorial you will create a simple claims management solution.

You can use Smart Folders for any purpose where you want to bring together files from across an organization, and apply metadata across a set of files. A good case study is an insurance claim, where you might want to bring together information for one customer, that relates to a claim and a specific policy.

 You will need system administrator rights to perform the activities in this tutorial.

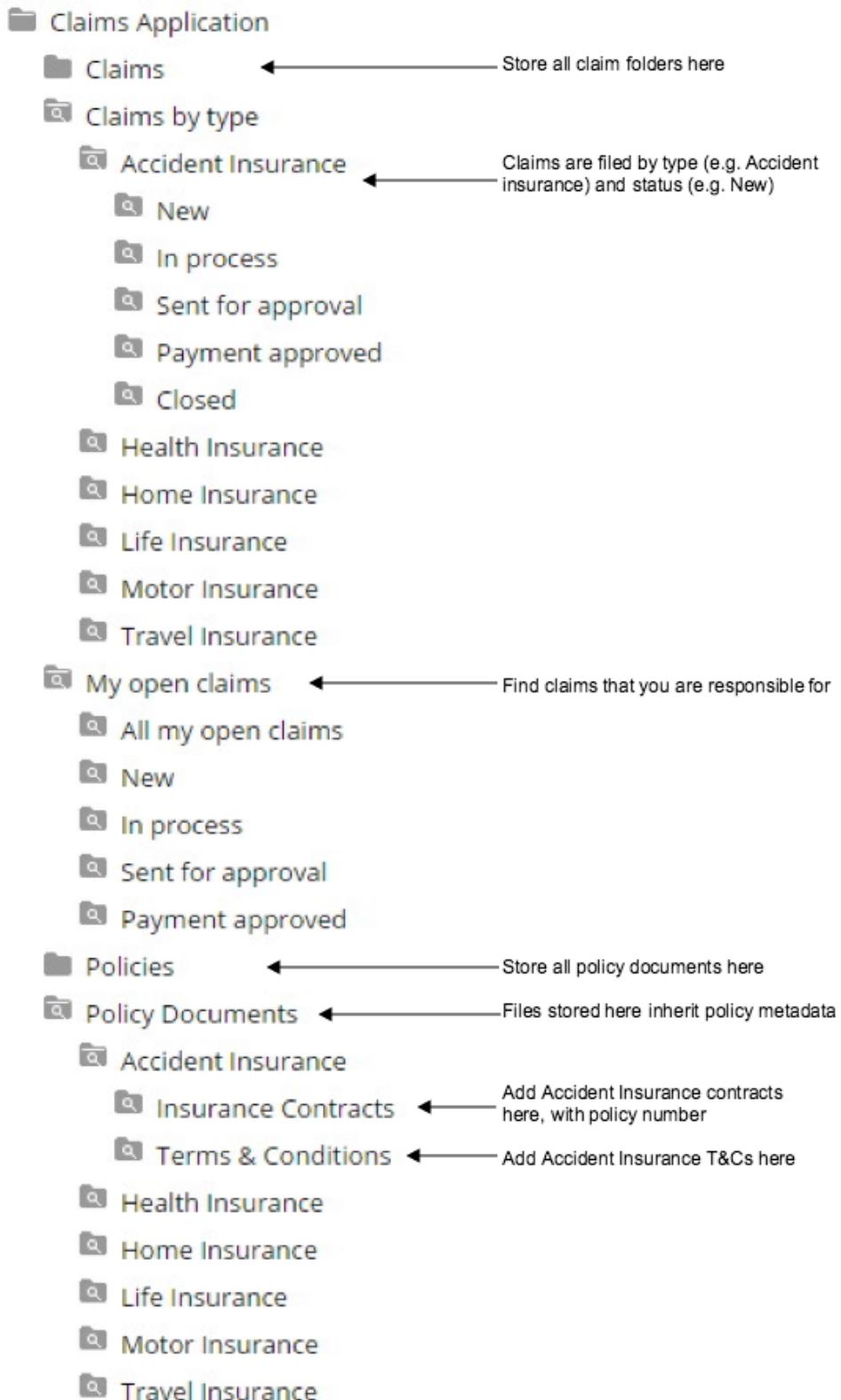
In the tutorial, you will:

1. Import a custom content model and create a `Claims Application` folder where your Smart Folder structure will live
2. Enable Smart Folders and the Type-based Smart Folder, and import the `clex_claimFolder.json` Smart Folders Template
3. Create a rule to automatically apply the aspects for your Smart Folder structure
4. Create a new claim
5. Add some supporting files to your claim
6. Apply a System Smart Folder to your `Claims Application` folder, to see how you can use different Smart Folder Templates together

7. Link your claim to related policy files

For more information about the Smart Folders workflow, see [Planning and implementing Smart Folders](#) on page 386. For information about Type-based, System, and Custom Smart Folders, see [Type-based, System, and Custom Smart Folders](#) on page 395.

The diagram shows the final folder structure that you will create during this tutorial:



For background information on Smart Folder Templates, see [What is a Smart Folder?](#) on page 384

## 1. Setting up claims management

To set up the claims framework, you need to create a custom content model, then create a folder structure for your content.

You need a custom content model to specify the metadata that the claims solution requires. You need to be in the ALFRESCO\_MODEL\_ADMINISTRATORS group to create a content model. For detailed information about Alfresco Model Manager, see [Content modeling with Model Manager](#) on page 406.

1. Download the [Smart Folders tutorial files](#) from the `smartfolders-master/tutorial` directory.  
You can download a zip of the Smart Folders master directory [here](#).
2. In Alfresco, select **Admin Tools** and **Model Manager**.  
The **Model Manager** page is displayed.
3. Click **Import Model** and browse to `smartfolders-master/tutorials` in your Downloads directory to import `claims_example.zip`, and click **Import**.  
You'll see the `claims_example` model and namespace, with a status of **Inactive**.
4. Select **Actions** and **Activate** to set the status to **Active**.  
Click `claims_example` to see the Custom Types and Aspects that are defined for the model.
5. Click **Sites** and **Create Site**. Create a new site called `Smart Folders`, and **Save**.
6. Select **Document Library** and create a new folder for the tutorial called `Smart Folders Tutorial`.
7. In the `Smart Folders Tutorial` folder, create a folder called `Claims Application`, and sub folders called `Claims` and `Policies`. You should see this structure:
  - `Smart Folders Tutorial/Claims Application`
  - `Smart Folders Tutorial/Claims Application/Claims`
  - `Smart Folders Tutorial/Claims Application/Policies`

You are now ready to configure your claim.

## 2. Configuring claims management

To configure the claims framework, add the sample template to the Data Dictionary and enable the claims aspect.

The example data model that you imported in the previous task contains the `clex:claimFolder` aspect. This aspect defines the metadata for a claim, and also marks a folder as being used to contain claim information. Make sure that you have downloaded the [Smart Folders tutorial files](#) before proceeding with this task.

1. Stop Alfresco, and edit your `alfresco-global.properties` file to specify the following settings:
 

```
smart.folders.enabled=true
smart.folders.config.type.templates.qname.filter=clex:claimFolder
```

The `smart.folders.config.type.templates.qname.filter` property specifies the custom type or aspect of the contents of the Smart Folder Template.
2. Restart Alfresco.
3. Browse to `smartfolders-master/tutorials` in your Downloads directory, and locate the `clex_claimFolder.json` file.

This is the Smart Folder Template.

This file matches the `cxlex:claimFolder` aspect, so that any folder type with the `cxlex:claimFolder` aspect applied to it should use the `cxlex_claimFolder.json` Smart Folder Template for its folder structure.

Adding this aspect and Smart Folder Template means that you are using Type-based Smart Folders.

4. In Alfresco, click **Repository** then **Data Dictionary**, and copy `cxlex_claimFolder.json` into the **Smart Folder Templates** folder.

You'll see the default `smartFoldersExample.json` Smart Folder Template is already in this folder.

 You need system administrator rights to upload this file.

You are now ready to create a new claim.

### 3. (Optional) Creating a rule to define your Smart Folder structure

You can create a simple folder rule to add an aspect automatically to your folder structure.

To simplify the creation of a claim folder, you can create a folder rule to add the `cxlex:claimFolder` aspect automatically to any new claim folder.

1. Upload the `addAspect_claimsFolder.js` file from `smartfolders-master/tutorials` in your Downloads directory to the `Repository/Data Dictionary/Scripts` directory in Alfresco.  
This file provides additional function that is not available in the standard aspect and property settings.
2. Click the site **Document Library** and drill down to the `Smart Folders Tutorial/Claims Application` folder.  
It's important that you create the rule for the `Claims` folder so that all sub folders will have the aspect that marks them as a claim.
3. Click the `Claims` folder and from the menu, click **More** then **Manage Rules**, and **Create Rules**.
4. Give the rule a name (Add Claims Folder aspect) and a description (Adds `cxlex:claimFolder` aspect and converts folder to a claim structure). Use the following options for the remaining fields:
  - Select **Define Rule When:** and **Items are created or enter this folder**
  - Check **If all criteria are met:** and select **Content of type or sub-type is folder**
  - Select **Perform Action:** and **Execute script**, and select the `addAspect_claimsFolder.js` file

 Make sure that these options are not selected:

- Rule applies to subfolders
- Run rule in background

5. When you are done, click **Create**.

### 4. Creating a new claim

You can create a new claim structure using the Smart Folder Template, and edit a new claim.

1. Click the site **Document Library** and drill down to the `Smart Folders Tutorial/Claims Application/Claims` folder.

2. Create a new folder called `Insurance Claim`.
  3. If you did not set up a [rule](#) to add the aspect:
    - a. Hover over the `Insurance Claim` folder and select **More** then **Manage Aspects**.
    - b. In the **Select Aspects** window, add the **Claim Folder (clex:claimFolder)** aspect, and **Save**.
  4. Hover over the `Insurance Claim` folder and select **Edit Properties**, and **All Properties**. You'll see a new section called **Claim Details** that has been populated from the Claim Folder aspect.
- Claim Number, Policy Number, Claim Type, Claim Status, Handling Administrator, and Claim Date fields are visible. Specify a unique claim number (and other data) in these fields, which are propagated to any file in this folder. A list of numbers is provided for testing.
5. Select the default numbers, a claim type of **Accident Insurance** and note the claim number. Click **Save**.
  6. Return to **Document Library**. You'll see the new Smart Folders are shown as sub folders of the Insurance Claim folder.

Smart Folders are identified by this icon: 

You can now add some files to your claim.

## 5. Adding new claim files

Add some files for the new claim.

You can add any files you like to the site Document Library for your claim. For convenience, an image (in JPG format) and a claim form (in PDF format) are provided in: [Smart Folders tutorial files](#).

1. Click the site **Document Library** and drill down to the `Smart Folders Tutorial/Claims Application/Claims/Insurance Claim` folder.
  2. Drag and drop any image (or the image from [Smart Folders tutorial files](#)) into the `Assessment` folder.
- Look in the `Assessments/Images` folder. The image you added is shown there. The image property defines that it must be filed in the `Images` folder.
3. Hover over the image and click **Edit Properties** then **All Properties** to view the Claim Details.

The image has inherited the Claim Number that you set up when you create the folder, and it has inherited the Assessment file type, because this is the folder where the file was dragged to. The file status is set to Draft by default.

- a. Change the **Document Type** to **Correspondence**. The image is viewable in the `Correspondence` Smart Folder.

- b. Change the **Document Status** to **In Review**. The image is viewable in the `Review processes/2_In Review` Smart Folder.

You can look at the `clex_claimFolder.json` file contents to understand the search criteria being applied to each folder. See [Smart Folder Template syntax](#) on page 397 for more guidance on understanding and creating your own templates.

4. Add a claim form (you can use the form from [Smart Folders tutorial files](#)) to the `Forms` Smart Folder.
5. Hover over the claim form and click **Edit Properties** then **All Properties** to view the Claim Details.

The form has inherited the Claim Number that you set up when you create the folder, and it has inherited the Claim Form file type, because this is the folder where the file was dragged to. The file status is set to none by default.

You have set up a claim structure, and learned how to configure it with a template, create a new claim folder, and populate it with content.

## 6. Applying multiple templates

You can use multiple Smart Folder Templates at the same time, to help you find your content more easily.

You can add System or Custom Smart Folders to your structure to use alongside the Type-based Smart Folders that you have already applied.

1. In Alfresco, click **Repository** and **Data Dictionary**, and copy `claimsApplication.json` from [Smart Folders tutorial files](#) into the **Data Dictionary/Smart Folder Templates** folder.  
You'll see the `smartFoldersExample.json` sample file (and any other templates you have added) already in this folder.
2. Select the `claimsApplication.json` file. In **Document Actions** select **Change Type** and select **Smart Folder Template** as the new type, and **OK**.
3. Click the site **Document Library** and drill down to the **Smart Folders Tutorial** folder.
4. Hover over the `Claims Application` folder and from the menu select **More** then **Manage Aspects**. Add the **System Smart Folder (smf:systemConfigSmartFolder)** aspect, and **Save**.

Adding this aspect allows you to select a Smart Folder Template that is in the **Data Dictionary/Smart Folder Templates** directory.

Alternatively, select the **Custom Smart Folder (smf:customConfigSmartFolder)** and select a template from anywhere in your repository.

 You can add a single template only to a folder. If you select both the **System Smart Folder (smf:systemConfigSmartFolder)** and **Custom Smart Folder (smf:customConfigSmartFolder)** aspects, the system aspect overrides the custom aspect.

5. Hover over the `Claims Application` folder and from the menu select **Edit Properties** and **All Properties**.
6. In the **Smart Folder Template** field, select the `claimsApplication.json` Smart Folder Template.  
If you need to navigate to the template, it lives in `Repository/Data Dictionary/Smart Folder Templates`.
7. In the site **Document Library**, click the `Claims Application` folder.  
You'll see the new folder hierarchy displayed, showing `Claims by type`, `My open claims`, and `Policy documents`. These Smart Folders are displayed in addition to the Smart Folders we set up under the `Claims Application` folder.

## 7. Adding policy files and reviewing the final claim structure

You can add files relating to the policy and review the Smart Folder structure that you have created for managing claims.

We have already created a new claim in previous steps, and built up the Smart Folder structure as part of the tutorial. In a real life scenario, the policy files would be created first, and the claim files linked to them afterwards. You can add any files you like to the Document Library for

your policy. For convenience, a Terms and Conditions file and a policy file (in PDF format) are provided in the zip package: [Smart Folders tutorial files](#).

1. Click the site **Document Library** and drill down to the `Smart Folders Tutorial/Claims Application/Policies` folder.  
You created a new Accident Insurance claim in the previous steps, and now you are going to add some policy files to the Accident Insurance type.
2. Drill down to the `Policy Documents/Accident Insurance` folder.
  - a. Drag and drop a Terms and Conditions file into the `Terms & Conditions` Smart Folder.  
See [Smart Folders tutorial files](#) for an example file.
  - b. Drag and drop a Policy file into the `Insurance Contracts` Smart Folder.  
See [Smart Folders tutorial files](#) for an example file.
  - c. Edit the properties of the Policy file.  
Hover over the policy file, and from the menu click **Edit Properties** and **All Properties**. You will see a **Policy Details** section, where you can set the Document Type, Policy Number and Insurance Class. If you specify a certain policy number, this must be specified in any subsequent claims that you create in the `Claims` folder.
3. Try creating a new folder in `Claims`, with a unique claim number, unique policy number (relating to your new policy file) and insurance class (Accident Insurance, in this case).

The diagram shows the final folder structure that you have set up for your claims management:



## Type-based, System, and Custom Smart Folders

There are three ways to attach Smart Folders to physical folders.

Each method assigns a Smart Folder Template to a physical folder, which is then immediately available to all users.

## Type-based Smart Folders

Type-based Smart Folders replicate a Smart Folder structure and apply it to many folders of a specific type, or carrying a specific aspect.

These are the key elements of Type-based Smart Folders:

- Best used to replicate a Smart Folder structure on multiple objects
- Allows you to configure new sections that are automatically embedded into folder and file properties
- Allows you to associate a Smart Folder Template with a specific type or an object that has a specific aspect
- New templates can be added in Repository/Data Dictionary/Smart Folder Templates
- Names must match. For example, in our tutorial the Smart Folder Template **clex\_claimFolder.json** matches the **Claim Folder (clex:claimFolder)** type
- Additional alfresco-global.properties settings required to enable this method. You could use any of these examples:

```
smart.folders.config.type.templates.qname.filter=*
smart.folders.config.type.templates.qname.filter=clex:claimFolder,dam:*
smart.folders.config.type.templates.qname.filter=none
```

The `smart.folders.config.type.templates.qname.filter` property can be set to one of the following:

- `none` for no types or aspects
- `*` for all types and aspects
- `<prefix>:*` for all types and aspects that are defined within a specified namespace
- `<prefix>:<name>` for a type or aspect with the specified name

Advanced Smart Folders settings are provided in the `<tomcat>/shared/classes/alfresco-global.properties.sample` file.

The setup of Type-based Smart Folders is somewhat complex, but is explained in detail in the tutorial. See [Configuring claims management](#) and [Creating a new claim](#) for more information.

## System Smart Folders

System Smart Folders are best used when you want to see content (that is distributed across the repository) in context; for example, all my files, or all files that are tagged as confidential.

These are the key elements of System Smart Folders:

- Best used to apply multiple taxonomies to find content in context
- Loaded using the **System Smart Folder (smf:systemConfigSmartFolder)** aspect
- Default template selected using the **Smart Folder Template** called `smartFoldersExample.json`
- New templates can be added in Repository/Data Dictionary/Smart Folder Templates.

 When you add a template to Repository/Data Dictionary/Smart Folder Templates, select **Change Type** and choose the **Smart Folder Template** type, to ensure that the new template is displayed in the list in Repository/Data Dictionary/Smart Folder Templates.

Advanced Smart Folders settings are provided in the <tomcat>/shared/classes/alfresco-global.properties.sample file.

See [Applying a Smart Folder Template](#) for more information.

### Custom Smart Folders

Custom Smart Folders are similar to System Smart Folders, except that you select the template from anywhere in your repository.

These are the key elements of the Custom Smart Folder:

- Best used to apply multiple taxonomies to find content in context
- Allows use of Smart Folder Templates that are located anywhere in the repository
- Loaded using the **Custom Smart Folder (smf:customConfigSmartFolder)** aspect
- Selected using the **Smart Folder Template** specific to your custom template

Advanced Smart Folders settings are provided in the <tomcat>/shared/classes/alfresco-global.properties.sample file.

See [Enabling Smart Folders](#) on page 387 and [Applying multiple templates](#) for more information.

## Metadata inheritance

You can set files and folders to inherit metadata using Smart Folders in Alfresco.

One of the most useful features of Smart Folders is the ability to automatically classify new files and inherit or map metadata to the file itself. This is possible by using Type-based Smart Folders, so that when you drag and drop files into your Smart Folder structure, they inherit any properties that you have set up in the Smart Folder Template.

Use the tutorial, [Smart Folders tutorial](#) on page 387 to set up a Smart Folder framework using Type-based Smart Folders, and in [Adding new claim files](#) you will see metadata inheritance in action.

Take a look at the Metadata Inheritance video to learn more: [Metadata inheritance with Smart Folders](#)

For more information on Type-based Smart Folders, see [Type-based Smart Folders](#) on page 396.

## Smart Folder Template syntax

You can build your own Smart Folder Template using these guidelines.

A Smart Folder Template is a configuration file that contains one or more queries to define the nodes of a hierarchical tree of “smart” folders. It is a JSON (Java Script Object Notation) file that defines one node object for every Smart Folder.

You can customize a copy of the `smartFoldersExample.json` template, which is available from `Repository/Data Dictionary/Smart Folder Templates` in Alfresco. The tutorial also provides links to a variety of examples; see [Smart Folders tutorial](#) on page 387 for more information.

For more information about Alfresco Full Text Search (AFTS), see [Alfresco Full Text Search Reference](#) on page 676.

A node is defined by the following properties:

Node property	Property description
<code>name</code>	Mandatory folder name

Node property	Property description
id	Optional ID or number that is unique for the node in the template. This property is optional, however it is recommended as specifying an ID generates a much shorter (and permanent) noderef for the Smart Folder
description	Optional description, displayed in the detailed view
nodes	Optional collection of sub nodes (sub folders)
search	Mandatory query defined using Alfresco FTS (full text search) language. The search is run when a Smart Folder is accessed by a user.
language	Mandatory property, set to <code>fts-alfresco</code>
query	Mandatory FTS query that defines the folder content
filing	Optional rule that defines the filing action for a new file when it is uploaded to the Smart Folder. If no filing rule is defined, files are not permitted to be uploaded to that folder. Parameters include: <ul style="list-style-type: none"> <li>• <code>path</code>: path where a document is physically stored</li> <li>• <code>classification</code>: type and aspects assigned to the new file</li> <li>• <code>properties</code>: property values attributed to the new file</li> </ul>
path	Mandatory property in a filing rule. Path to store new documents. This is the <a href="#">ISO9075</a> encoded QName.
classification	Mandatory property in a filing rule. Type and aspects of the new object.
properties	Optional property. Defines property values and inheritance.

The following code fragments give more information about these properties.

Here are some additional tips on notation:

- Use percent (%) signs to use predefined placeholders in queries and filing rules
- For repository path expressions use Alfresco QNames, for example; `/app:company_home/st:sites/cm:swsdp/cm:documentLibrary`.
- Special characters and whitespace are [ISO9075](#) encoded. Use this notation to encode special characters in repository path names. For example, use `_x0020_` for the whitespace character.

Placeholder	Description
<code>%ACTUAL_PATH%</code>	<a href="#">ISO9075</a> encoded repository path of the physical parent folder. Only the physical parent folder (or next physical folder up the folder tree) can use <code>%ACTUAL_PATH%</code> .
<code>%CURRENT_USER%</code>	Account name of the user
<code>_x0020_</code>	<a href="#">ISO9075</a> encoded whitespace character

Placeholder	Description
<>	Use angle brackets, for example, <cm:name>, to inherit property values from the physical parent folder. Used for inheritance in a filing rule and in a query.

1. **Nested nodes:** define a Smart Folder structure inside another Smart Folder structure, for example:

```
{
 "id": "1",
 "name": "Documents",
 "nodes": [
 {
 "name": "Correspondence",
 "description": "Smart Folder - documents from type 'Correspondence'",
 "nodes": [
 {
 "name": "High Prio",
 ...
 },
 {
 ...
 }
],
 "name": "Assessment",
 "description": "Smart Folder - documents from type 'Assessment'"
 },
 {
 "name": "Pending approvals",
 "description": "Smart folder - pending approvals documents"
 }
]
}
```

2. **Search queries:** information is populated by running a search query:

```
{
 "id": "1",
 "name": "Documents",
 "nodes": [
 {
 "name": "Correspondence",
 ...

 "search": {
 "language": "fts-alfresco",
 "query": "=cmg:claimDocumentType:Correspondence and cmg:claimDocumentId:<cmg:claimId>"
 }
 },
 {
 ...
 }
]
}
```

The query is run when the Smart Folder is opened in Alfresco, and the results displayed as the folder contents.

You can limit the query to specific types or aspects, for example:

```
"query": "+ASPECT:'ins:claimFolder'"
"query": "+TYPE:'cm:folder'"
```

You can use %CURRENT\_USER% to limit the search to documents relevant to the logged in user, for example:

```
"query" : "cm:modifier:%CURRENT_USER% or cm:creator:%CURRENT_USER%"
```

3. **Filing rules:** define the path where a document uploaded to a Smart Folder should be created, as well as the type and aspects of the new file, and its property values:

```
{
 "id": "1",
 "name": "Documents",
 "nodes": [
 {
 "name": "Correspondence",
 ...
 "filing": {
 "path": "%ACTUAL_PATH%",
 "classification": {
 "type": "cm:content",
 "aspects": [
 "cmg:claim-document"
]
 },
 "properties": {
 "cmg:claimDocumentType": "Correspondence",
 "cmg:claimDocumentId": "<cmg:claimId>"
 }
 }
 },
 ...
]
}
```

- **"path":** The path can be an existing folder location, for example:

- ```
"path": "/app:company_home/cm:Claims_x0020_Pool"
```

using an XPath expression, and ensuring the expression is ISO9075 encoded

- or the parent folder, by specifying the placeholder %ACTUAL_PATH%.

The path variable can also be used in a query to restrict the search to a certain folder:

```
"query": "PATH: '/app:company_home/st:sites/cm:legal-documents/'"
```

or in a filing rule to store new objects:

```
"path": "/app:company_home/cm:Insurance/*"
```

- **"classification":** You can define the type for content that populates a Smart Folder, and which aspects should be associated to them. In the [code example](#), each new document shown in the "Correspondence" folder is of type "cm:content" with aspect "cmg:claim-document".
- **"properties":** You can assign property values. These can be fixed, or a placeholder "<[property_name]>" that uses the value of the parent folder property.

In a Smart Folder, you can map the value of the parent folder or object to that of a new object as variables:

```
"[new_obj_prop_name]": "<[existing_obj_prop_name]>"
```

For example, "cmg:claimDocumentId": "<cmg:claimId>"

or as names:

```
"[new_obj_prop_name]": "[literal]"
```

For example, "cmg:claimDocumentType": "Correspondence"

You can also use the value of the parent folder or object in a search query, for example:

```
"query": "=cmg:claimDocumentType:Correspondence and  
cmg:claimDocumentId:<cmg:claimId>"
```

Smart Folders global properties settings

Use this information to understand the full list of `alfresco-global.properties` settings available for Smart Folders.

Settings for Smart Folders are listed in the `<tomcat>/shared/classes/alfresco-global.properties.sample` file:

```
#Smart Folders Config Properties

smart.folders.enabled=true
smart.folders.model=alfresco/model/smarterFolder-model.xml
smart.folders.model.labels=alfresco/messages/smarterFolder-model

#Smart reference config

#smart.reference.classpath.hash=
${smart.folders.config.vanilla.processor.classpath}->1,
${smart.folders.config.system.templates.classpath}->2

#Smart store config

#Company home relative download associations of smart entries
#smart.download.associations.folder=${spaces.dictionary.childname}/
${spaces.smartdownloads.childname}

#Generic virtualization methods config

#Vanilla JSON templates javascript processor classpath. A java script processor
used to
#covert JSON templates to internal smart folder definitions.

#smart.folders.config.vanilla.processor.classpath=/org/alfresco/repo/virtual/
node/vanilla.js

#System virtualization method config

#System virtualization method aspect.
#smart.folders.config.system.aspect=smf:systemConfigSmartFolder
#System virtualization method aspect defined template location property.
#smart.folders.config.system.aspect.template.location.property=smf:system-
template-location
#Classpath to be explored for *.json entries defining system templates.
#smart.folders.config.system.templates.classpath=/org/alfresco/repo/virtual/
node
#A company home relative name or qname path location of repository system
templates.
#smart.folders.config.system.templates.path=${spaces.dictionary.childname}/
${spaces.smartfolders.childname}
#Content sub type of repository system templates.
#smart.folders.config.system.templates.template.type=smf:smartFolderTemplate

#Custom virtualization method config

#Custom virtualization method aspect.
#smart.folders.config.custom.aspect=smf:customConfigSmartFolder
```

```
#Custom virtualization method aspect template content association.
#smart.folders.config.custom.aspect.template.association=smf:custom-template-
association

#Type virtualization method config

#A company home relative name or qname path location of the type mapped
templates.
#smart.folders.config.type.templates.path=${spaces.dictionary.childname} /
${spaces.smartfolders.childname}
#Type and aspect qname regular expression filter.
#smart.folders.config.type.templates.qname.filter=none
```

The different sections are used in the following ways:

1. Smart Folders config properties: these are the basic mandatory settings for Smart Folders.
`smart.folders.enabled=false`
is the default, and must be set to `true` to enable Smart Folders.
2. Smart reference config: reduces the length of NodeRefs
3. Smart store config: If you use the **Download as Zip** function in Share for a folder that contains Smart Folders, a temporary file is created in the `Data Dictionary/Smart Folder Downloads` folder that contains information about the Smart Folder contents (an association folder). Use this variable to change where the association folder lives.
4. Generic virtualization methods config: defines overall Smart Folder Template classpath. By default, templates live in `<configRootShare>\classes\org\alfresco\repo\virtual\node`
5. System virtualization method config: defines the configuration for System Smart Folders. See [System Smart Folders](#) on page 396 for more information.
6. Custom virtualization method config: defines the configuration for Custom Smart Folders. See [Custom Smart Folders](#) on page 397 for more information.
7. Type virtualization method config: defines the configuration for Type-based Smart Folders. See [Type-based Smart Folders](#) on page 396 for more information.

Best practices when using Smart Folders

There are a number of best practices when using Smart Folders in Alfresco.

Server Configuration and Alfresco Search Service:

- Configure the Alfresco Search Service to use Solr4
- Configure transactional queries in the Search Service to use the database always, or if possible.
- When you define a search query, restrict the query to certain types or aspects (using `+TYPE` or `+ASPECT`), otherwise the query will search for all content.
- When defining a filing rule for a Smart Folder, use a transactional query for that folder where possible, otherwise uploaded files will not appear immediately. See [Transactional metadata queries supported by database](#) on page 331 for more information.

Smart Folder Templates:

- Use the FTS query language (this is mandatory for Smart Folders). All other languages are experimental and do not allow creation or upload of new objects into a Smart Folder.
- If you are using CIFS or WebDAV, only `cm:folder` types are supported for a folder. Do not use a sub type of `cm:folder`; instead use aspects to apply properties to a folder.
- Don't create filing rules that don't match the query criteria for the folder.

- Don't use folder types in a filing rule (creating physical folders in Smart Folders is not supported).

General guidance:

- Use the optional `id` property for every folder node to shorten the `noderef` for a Smart Folder (the length of `noderefs` can become critical). The ID must be unique in a template.
- Don't use file system protocols such as WebDAV, CIFS, or IMAP for uploading content (these are not supported).
- When you create a model, don't use the – (dash) character in a type, aspect, or property name. A better method is to use mixed case in your names. If you have used the – character in a property name, you must escape the property name in a Smart Folder Template, using `\`; for example, `mod:first-name` must be escaped to `mod:first\\-name`.

Smart Folders technical FAQs

If you have any technical problems with Smart Folders, try these suggestions to resolve your issue.

- [How do I enable Smart Folders?](#)
- [How do Smart Folders affect Alfresco if the function is enabled, but not used?](#)
- [Are there best practices to avoid performance problems?](#)
- [Where can I find technical documentation?](#)
- [Which components or subsystems do Smart Folders provide?](#)
- [Which services do Smart Folders extend?](#)
- [How can I extend Smart Folders? Which interfaces, APIs, and extension points exist?](#)
- [As a developer, I want to create a custom application. How can I use Smart Folders in my code?](#)
- [As a developer, creating my own custom application \(separate from the Share evaluators\), how do I distinguish between a Smart Folder and an object that is displayed in a Smart Folder?](#)
- [I have created a new Share module. How can I enable Share actions for Smart Folders?](#)
- [Which Share actions are enabled as standard for Smart Folders?](#)
- [Are Smart Folders supported in a multi-tenant production environment?](#)
- [Can I use multi-value properties with Smart Folders?](#)
- [What is the Data Dictionary/Smart Downloads folder used for?](#)

How do I enable Smart Folders?

Smart Folders can be enabled (and disabled) by your system administrator in the `alfresco-global.properties` file:

```
smart.folders.enabled=true
```

[back to top](#)

How does Smart Folders affect Alfresco if the function is enabled, but not used?

Smart Folders is part of the standard Alfresco repository, and there might be unexpected interactions if Smart Folders are enabled but not used. The most important impact is on performance. Alfresco performance might degrade based on the complexity and the number of Smart Folders used. See [Best practices when using Smart Folders](#) on page 402 for more information about best practices with Smart Folders, and [Type-based, System, and Custom](#)

[Smart Folders](#) on page 395 for information about Type-based, System, and Custom Smart Folders.

[back to top](#)

Are there best practices to avoid performance problems?

Yes there are. See [Best practices when using Smart Folders](#) on page 402 for more information.

[back to top](#)

Where can I find technical documentation?

Use the Smart Folder tutorial, [Smart Folders tutorial](#) on page 387, to set up a working Smart Folder configuration. General configuration information is here: [Configuring Smart Folders](#) on page 383.

[back to top](#)

Which components or subsystems do Smart Folders provide?

Smart Folders introduce a private AspectJ-based, non-disruptive extension point implementation mechanism called TraitExtender.

The Smart Folders implementation extension bundle, which is a set of extensions that act as service interceptors for several services, is classified as a module because it can be added to and removed from the Alfresco repository. They are not publicly exposed currently.

[back to top](#)

Which services do Smart Folders extend?

These services are extended:

- DbNodeServiceImpl basic node handling
- FileFolderServiceImpl basic file and folder modelling
- PreferenceServiceImpl node based preferences information handling (for example; favourites)
- RatingServiceImpl node based ratings
- LockServiceImpl node locking
- PermissionServiceImpl node permissions handling
- CheckOutCheckInServiceImpl node check out / check in
- Version2ServiceImpl node versioning

The LockableAspectInterceptor Spring interceptor is also extended.

[back to top](#)

How can I extend Smart Folders? Which interfaces, APIs, and extension points exist?

There are currently no publicly exposed extension points for Smart Folders.

[back to top](#)

As a developer, I want to create a custom application. How can I use Smart Folders in my code?

Use standard Alfresco APIs. All custom uses of nodes, files, and folders apply.

[back to top](#)

As a developer, creating my own custom application (separate from the Share evaluators), how do I distinguish between a Smart Folder and an object that is displayed in a Smart Folder?

Use this guidance to differentiate between a Smart Folder and an object that is displayed in a Smart Folder:

- **Root folder, with Smart Folder sub folders:** This is a standard folder and is referenced using normal NodeRefs. The only difference is the sub-folder content.
- **Smart Folder structure, as defined by a Smart Folder Template:** This structure has the `smf:smart` aspect, and is referenced using a special NodeRef. Repository clients must not use the NodeRef format to detect these nodes.
- **Smart Folder structure, as defined by a query:** This structure has the `smf:smartFolderChild` aspect, and is referenced using a special NodeRef. Repository clients must not use the NodeRef format to detect these nodes.

[back to top](#)

I have created a new Share module. How can I enable Share actions for Smart Folders?

Any new Share action, by default, is not enabled, because not all actions are supported with Smart Folders (for example, data can't be persisted with a Smart Folder).

Ensure that you conduct adequate testing before enabling an action for Smart Folders.

To enable the new action, add an evaluator in the appropriate action group definition in your module, or add it to the Tomcat `shared/classes/alfresco/web-extension/smartsfolders-amp-actions-config.xml` file.

[back to top](#)

Which Share actions are enabled as standard for Smart Folders?

Smart Folders have a limited set of actions:

- Add/ Create: You can add files to a Smart Folder. The file is put into a physical folder, as specified by the filing rule.
- Update: You can update files in a Smart Folder. Updating a property might result in a file being removed from the current Smart Folder (because it no longer meets the query criteria).
- Delete, Edit Properties, Unzip To, Sync, Locate To, Move, and Copy actions for files are not supported.

The Smart Folder itself can't be edited in Alfresco, except through the Smart Folder Template. For more information about Smart Folder Templates, see [Applying a Smart Folder Template](#).

[back to top](#)

Are Smart Folders supported in a multi-tenant production environment?

No, Smart Folders are not supported in a multi-tenant production environment.

[back to top](#)

Can I use multi-value properties with Smart Folders?

Properties that can have multiple values are supported in a query. However, you can't upload new content and set a value for a multi-value property.

[back to top](#)

What is the Data Dictionary/Smart Folder Downloads folder used for?

The Data Dictionary/Smart Folder Downloads folder is used only when you use the **Download as Zip** function in Share, for a folder that contains Smart Folders. **Download as Zip** creates a temporary file in the Smart Folder Downloads folder that contains information about the Smart Folder contents.

[back to top](#)

Content modeling with Model Manager

Model Manager allows you to create and manage your own custom models in Alfresco Share. This is a user-friendly tool that enables you to add custom types, aspects, and properties to your models. Alfresco provides several out-of-the-box content models for specifying the core content types in the repository.

The screenshot shows the Alfresco Share interface with the 'Model Manager' page selected. At the top, there is a navigation bar with links for Shared Files, Sites, Tasks, People, Repository, Admin Tools, Administrator, and a search bar. Below the navigation bar, the page title 'Manager' is displayed. The main content area is titled 'Models'. It features two buttons: 'Create Model' (highlighted in blue) and 'Import Model'. A table lists existing models with columns for Name, Namespace, and Status. The table shows two entries: 'Finance' with Namespace 'http://www.democo.com/model/finance/1.0' and Status 'Inactive', and 'Document' with Namespace 'http://www.democo.com/model/document/1.0' and Status 'Active'.

| Name | Namespace | Status |
|----------|--|----------|
| Finance | http://www.democo.com/model/finance/1.0 | Inactive |
| Document | http://www.democo.com/model/document/1.0 | Active |

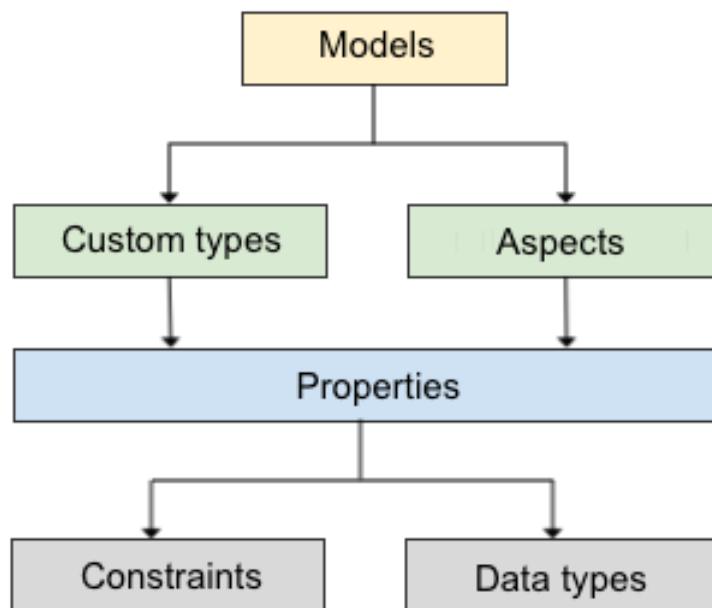
The Model Manager is available to users in the `ALFRESCO_MODEL_ADMINISTRATORS` permission group. To create or edit a model you must be a member of this group. By default, the System Administrator is a member of the `ALFRESCO_MODEL_ADMINISTRATORS` group and can create models.

This information is primarily aimed at business analysts who need to define and manage content models in Alfresco.

What is a model?

A model is a fundamental building block of the Alfresco content repository that provides a foundation for structuring content and working with content. A model has the following characteristics:

- It describes the data being stored in Alfresco.
- It allows the management of content metadata by applying custom types or aspects to the content and folders.
- It is uniquely identified by its defined Namespace, Prefix, and Name.
- It is defined using a small set of building blocks: custom types, aspects, properties, and constraints.



Custom types

A custom type enumerates the properties and relationships that a file of that type can support. Typically, types represents nodes in Alfresco with support for properties and the ability to inherit the definition of a parent type. Content and Folder are the two important types defined out-of-the-box.

Aspects

An aspect is a collection of properties that can encapsulate both data and behaviour, providing a flexible tool for modeling content. Aspects add extra functionality and properties to the models by attaching them to custom types. A file in Alfresco must be of a single type, but may have one or

more aspects attached to it. By default, the content repository comprises of some out-of-the-box aspects, such as Classifiable, Versionable, and so on. To know more about aspects, see [About aspects](#).

Properties

Properties are metadata which describes the content. For example, **Author** is a property which specifies the person who wrote the content.

Constraints

Constraints control the input property values. For example, you can specify that the author name must not be more than 40 characters.

For more information on content modeling, see [Content metamodel](#).

Managing models

You can create and manage your own business-specific models using the Model Manager tool.

Creating a new model

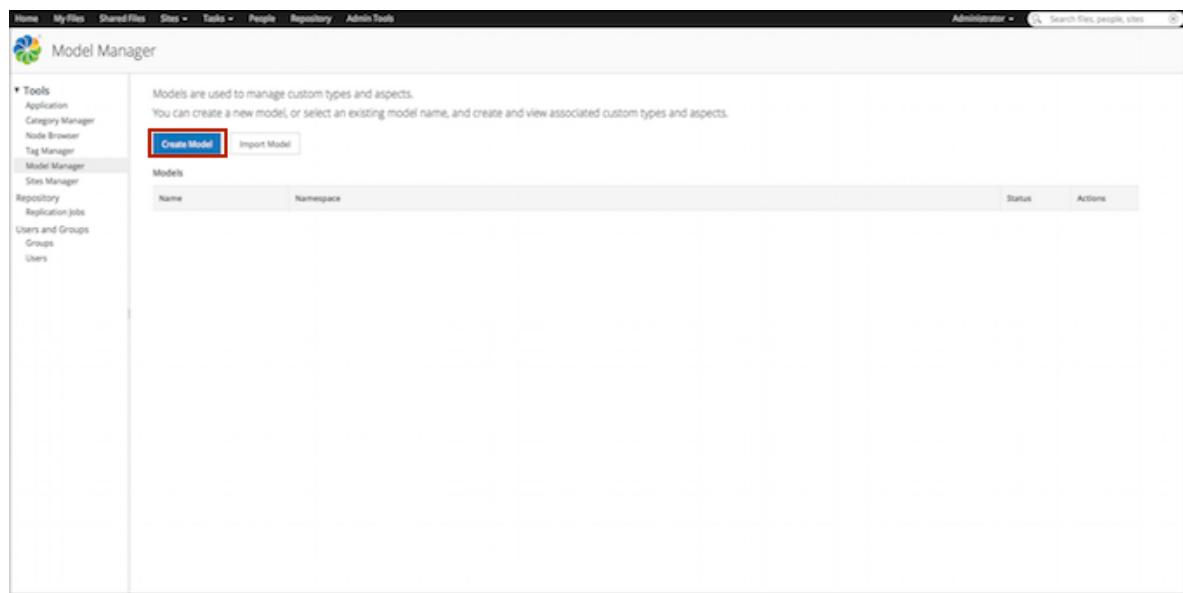
You can create new models using the **Model Manager** page.

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

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_MODEL_ADMINISTRATORS permission group.

The **Model Manager** page is displayed.

2. Click **Create Model**.



The **Create Model** window appears. Fields marked with an asterisk (*) are required.

3. Enter the details for the new model.

Namespace: *

Enter a unique model namespace. For example, if you're creating a model for financial documents you might use a URI similar to <http://www.mycompany.com/model/finance/1.0>.

Prefix: *

Enter a namespace prefix for the model. This is shorthand for the namespace URI and is used as the namespace prefix and for advanced search property syntax.

Name: *

Enter a name for the model.

Creator:

Enter an optional creator of the model.

Description:

Create Cancel

- Enter the model namespace.

Namespaces provide a way to specify globally unique names for definitions within content models. All custom types, aspects, and properties have names which are made unique across the repository by using a namespace specific to the model. Using namespaces prevents name collisions when the models are shared across repositories. A namespace is composed of a URI and a prefix.

Only alphanumeric characters or a URI, for example, `http://www.mycompany.com/model/mynamespace/1.0`, are allowed. Do not use spaces or special characters.

The namespace value **must** be unique within the Alfresco repository.

- Enter a short prefix for the model.

A prefix is just an abbreviation for the namespace identifier (URI), which is typically quite long. For example, if the namespace URI is `http://example.org/contentmodels` and associated prefix is `ex`, then the name `ex:customtype` means that `customtype` is a name defined within the namespace `http://example.org/contentmodels`.

Only alphanumeric characters, hyphens (-), and underscores (_) are allowed. Do not use spaces. For example, `finance`.

The prefix value **must** be unique within the repository.

- Enter a name for the model.

For example, `Finance`.

Only alphanumeric characters, hyphens (-), and underscores (_) are allowed. Do not use spaces or special characters.

- d. Specify an optional author for the model.
If you leave this field blank, Alfresco Share will auto-fill it based on the current logged in user.
- e. Enter an optional description for the model.

4. Click **Create**.

The new model appears in the **Custom Models** table on the **Model Manager** page.

Additional information displayed in the table includes the model's name, namespace, status, and the actions that can be performed on the type.

The status of a model can either be **Active** or **Inactive**.

- Inactive models: A new model is initially inactive. This means that the model is work in progress and will not be visible in Alfresco Share. A model can be moved from an inactive status to an active status.
- Active models: Active models can be used by the end users of Alfresco Share and any custom types or aspects defined within the model can be applied to files. Active models can be moved back to the inactive status if there are no instances of the model's types or aspects in use within Alfresco Share.

Viewing existing models

Use the **Model Manager** page to create a new model, select an existing model to view and create the associated custom types and aspects.

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

The **Model Manager** page displays all the models created using the Model Manager module. The list displays the model name, its namespace, status, and the available actions.

2. Under **Custom Models**, click a model name to view the custom type(s) and aspect(s) associated with it.
3. To perform an action on a model, click the **Actions** drop-down list against the relevant model name.

The options available in the **Actions** drop-down list will depend on the status of the model.

| Option | Available when content model status is.. | Description |
|-------------------|--|---|
| Activate | Inactive | Activates the model so that the custom type or aspect is available to the end users of Alfresco Share. |
| Edit | Inactive | Enables you to update information of the model using the Edit Model window. |
| Delete | Inactive | Deletes the model. You can only delete an inactive model. To delete an active model, you need to deactivate it first. |
| Deactivate | Active | Deactivates the model so that the custom type or aspect is no longer available to the end users of Alfresco Share. |
| Export | Active/Inactive | Saves the model on your local machine for future use or use in other Alfresco repositories. |

Inactive model:

| Name | Namespace | Status | Actions |
|------------------|---|--------|--|
| Finance Document | http://www.democo.com/model/finance/1.0
http://www.democo.com/model/document/1.0 | Active | Actions Activate Edit Delete Export |

Active model:

| Name | Namespace | Status | Actions |
|------------------|---|--------|--|
| Finance Document | http://www.democo.com/model/finance/1.0
http://www.democo.com/model/document/1.0 | Active | Actions Deactivate Edit Delete Export |

Activating a model

Activate a model to make its custom type or aspect available to other users in Alfresco Share.

To activate a model, follow the instructions below:

1. Click **Admin Tools**, and then click **Model Manager**.
The **Model Manager** page is displayed. The status of the relevant model is **Inactive**.
2. Click the **Actions** drop-down list for the model you want to activate.
3. Click **Activate**.

| Name | Namespace | Status | Actions |
|------------------|---|----------|--|
| Finance Document | http://www.democo.com/model/finance/1.0
http://www.democo.com/model/document/1.0 | Inactive | Actions Activate Edit Delete Export |

The status of the relevant model changes to **Active**. The types and aspects associated with this model are now available in Alfresco Share.

Deactivating a model

Deactivate a model to make its associated custom type and aspect unavailable in Alfresco Share.

If you have applied a model's custom type or aspect to a file in Alfresco Share, you cannot deactivate the model directly from the **Model Manager**. Use the **Find Where Used** option to

search and locate all the nodes in Alfresco to which the relevant type or aspect has been applied. For more information, see [Creating new custom types and aspects](#).

- If you have applied a model's custom type to any files and folders in Alfresco Share, delete the files or folders that use the type from Alfresco Share and your trashcan, and then deactivate the model using **Model Manager**.
- If you have applied a model's aspect to a file in Alfresco Share, remove the aspect from the node, then deactivate the model using **Model Manager**.
- If you have applied a model's custom type and aspect to a file in Alfresco Share, delete the file from Alfresco Share and your trashcan, and then deactivate the model using **Model Manager**.

To deactivate a model, follow the instructions below:

1. Click **Admin Tools**, and then click **Model Manager**.
The **Model Manager** page is displayed. The status of the relevant model is **Active**.
2. Click the **Actions** drop-down list for the model you want to deactivate.
3. Click **Deactivate**.

| Models | | Status | Actions |
|----------|--|----------|---------|
| Name | Namespace | | |
| Finance | http://www.democo.com/model/finance/1.0 | Inactive | Actions |
| Document | http://www.democo.com/model/document/1.0 | Active | Actions |

The status of the relevant model changes to **Inactive**. The types and aspects associated with this model are no longer available in Alfresco Share.

4. Once you have deactivated the model, you can delete it by clicking **Delete** from the **Actions** drop-down list.
- If you have created a filter for a custom model, remember to remove the filter from the [Search Manager](#) when you delete the model.

The deleted model is removed from the Model Manager.

Exporting/importing models

Model Manager provides a feature for exporting and importing models across different repositories in Alfresco.

You can export and import only those models which are created using the Model Manager.

Exporting a model

Model Manager allows you to export layouts, custom types, aspects, and their associated properties for use in other repositories within Alfresco.

To export a model, follow the steps below:

1. Click **Admin Tools**, and then click **Model Manager**.
The **Model Manager** page is displayed.
2. Click the **Actions** drop-down list for the model you want to export.
3. Click **Export**.

| Name | Namespace | Status | Actions |
|----------|--|----------|-----------|
| Finance | http://www.democo.com/model/finance/1.0 | Inactive | Actions ▾ |
| Document | http://www.democo.com/model/document/1.0 | Active | Actions ▾ |

Export

This will save the model locally on your computer as a ZIP package.

Importing a model

You can import a model in Model Manager only as a ZIP package.

- You can't import models with the same name as the existing models in Model Manager. Model names must be unique.

To import a model, follow the steps below:

1. Click **Admin Tools**, and then click **Model Manager**.
The **Model Manager** page is displayed.
2. Click the **Import Model** and select a ZIP file to import.

| Name | Namespace | Status | Actions |
|----------|--|----------|-----------|
| Finance | http://www.democo.com/model/finance/1.0 | Inactive | Actions ▾ |
| Document | http://www.democo.com/model/document/1.0 | Active | Actions ▾ |

- You can only import ZIP files that have been exported from Model Manager.

The imported model appears in the **Custom Models** table on the **Model Manager** page.

Managing custom types, aspects, and properties

You can create and manage custom types, aspects, and properties for your models using the Model Manager tool.

Creating new custom types and aspects

A model can have one or more custom types and aspects. You can create new custom types and aspects using the Model Manager.

- Within a model, the custom types and aspects **must** have unique names.
1. Click **Admin Tools**, and then click **Model Manager**.
The **Model Manager** page is displayed.
 2. Click the model name for which you want to add the custom type and / or the aspect.

Configuring

The screenshot shows the Alfresco Model Manager interface. The left sidebar has a 'Tools' section with 'Model Manager' selected. The main area displays a table of models:

| Name | Namespace | Status | Actions |
|----------|--|----------|---------|
| Finance | http://www.democo.com/model/finance/1.0 | Inactive | Actions |
| Document | http://www.democo.com/model/document/1.0 | Active | Actions |

A red arrow points to the 'Document' row.

The relevant model page appears. This page shows the existing custom types and aspects associated with the selected model.

3. To create a new custom type or aspect, click on the relevant tab.

- To create a type, click **Create Custom Type**. The **Create Custom Type** window appears.

The screenshot shows the 'Document' model page. The left sidebar has 'Model Manager' selected. The main area shows:

Document

Create Custom Type (highlighted with a red arrow)

Create Aspect

Custom Types

| Name | Display Label | Parent |
|--|---------------|--------|
| No types found. Click Create Custom Type to get started. | | |

Aspects

| Name | Display Label | Parent |
|---|---------------|--------|
| No aspects found. Click Create Aspect to get started. | | |

- To create an aspect, click **Create Aspect**. The **Create Aspect** window appears.

The screenshot shows the 'Model Manager' page. On the left, there's a sidebar with a 'Tools' section containing links like Application, Category Manager, Node Browser, Tag Manager, Model Manager (which is selected and highlighted in grey), Sites Manager, Repository, and Replication Jobs. Below that is a 'Users and Groups' section with Links for Groups and Users. At the top, there's a navigation bar with Home, My Files, Shared Files, Sites, Tasks, People, Repository, and Admin Tools. The main content area has a 'Document' header. It includes a 'Create Custom Type' button and a 'Create Aspect' button (which is highlighted with a red arrow). Below these are sections for 'Custom Types' and 'Aspects', each with a table-like structure for managing them.

4. Enter the details for the new custom type and / or the aspect. Fields marked with an asterisk (*) are required.

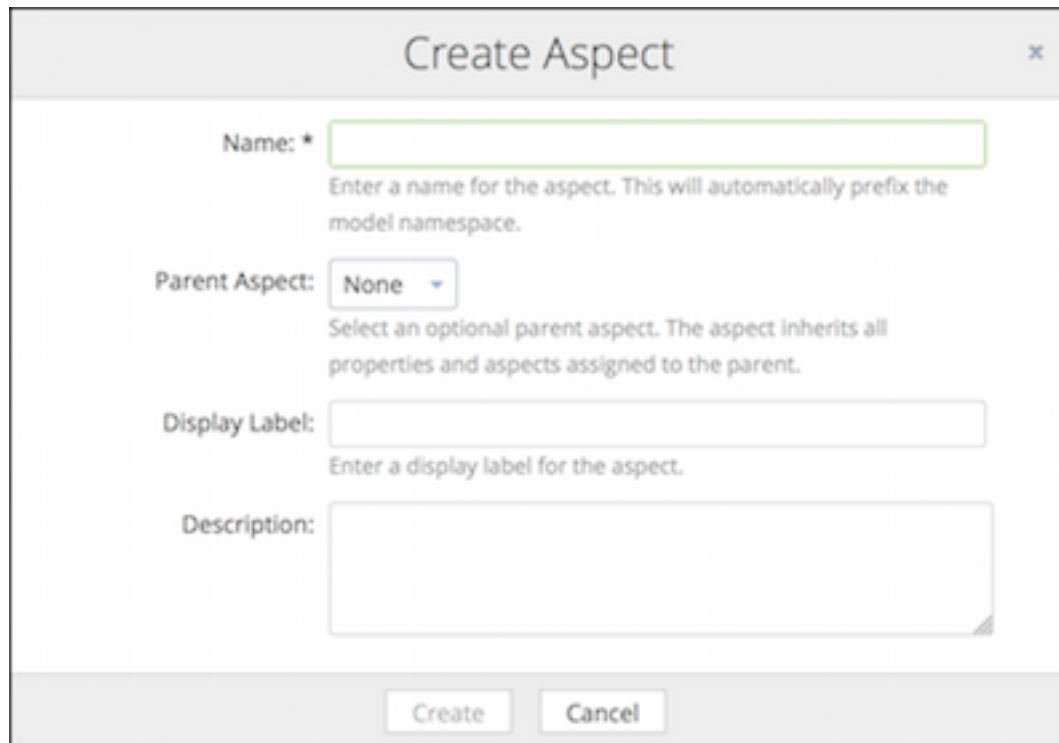
For custom type:

The dialog box is titled 'Create Custom Type'. It contains the following fields:

- Name:** * (The input field is highlighted with a green border.)
Enter a name for the type. This will automatically prefix the model namespace.
- Parent Type:** (The dropdown menu is open.)
Select the parent type. The custom type inherits all properties and aspects assigned to the parent.
- Display Label:**
Enter a display label for the type.
- Description:**

At the bottom, there are 'Create' and 'Cancel' buttons.

For aspect:



- a. Enter a name for the type and / or aspect.

Only alphanumeric characters, hyphens (-), and underscores (_) are allowed. The model name will automatically prefix the model namespace.

- b. Select the parent type for the type and / or aspect.

- The parent type must either be a sub-type of the `cm:content` type or `cm:folder` type. The custom type will inherit all the properties and aspects assigned to the parent.
- The aspect will inherit all the properties and aspects assigned to the parent. The default parent type is none.

- c. Enter an optional display label for the type and / or aspect.

The display label is shown to the users as the model name in the **New Type** drop down in Alfresco Share.

For example, Invoice.

- d. Enter an optional description of the type and / or aspect.

5. Click **Create**.

For custom type: The new custom type appears in the **Custom Types** table. The name of the custom type is of the format, `Prefix:Custom type name`. Additional information displayed in the table includes the type name, display label, parent, and the actions ([Layout Designer](#), [Edit](#), [Delete](#), and [Find Where Used*](#)) that can be performed on the type.

| Name | Display Label | Parent | Layout | Actions |
|---------------|---------------|------------|--------|--|
| dc:whitePaper | White paper | cm:content | No | <input type="button" value="Actions"/> |

| Name | Display Label | Parent | Layout | Actions |
|------------|---------------|--------|--------|---|
| dc:webable | Webable | | No | <input type="button" value="Actions"/> <input checked="" type="checkbox"/> Layout Designer <input type="checkbox"/> Edit <input type="checkbox"/> Delete <input type="checkbox"/> Find Where Used |

For aspects: The new aspect appears in the **Aspects** table. The name of the custom type is of the format, `Prefix:Aspect name`. Additional information displayed in the table includes the aspect name, display label, parent, and the actions ([Layout Designer](#), [Edit](#), [Delete](#), and [Find Where Used*](#)) that can be performed on the aspect.

- ✎ * The **Find Where Used** option searches and locates all the nodes in Alfresco to which the relevant type or aspect has been applied, and displays the result on the [Search](#) result page.

Editing custom types and aspects

Use this information to edit a custom type and an aspect.

1. Click **Admin Tools**, and then click **Model Manager**.
The **Model Manager** page is displayed.
2. Click the relevant model from the **Custom Models** list.
The selected model page appears. This page shows the existing custom types and aspects associated with the selected model.
3. To edit a custom type or aspect, perform the following action:
 - For the type you want to edit, click **Edit** from the **Actions** drop-down list. The **Edit Custom Type** window appears.

| Name | Display Label | Parent | Layout | Actions |
|---------------|---------------|------------|--------|---|
| dc:whitePaper | White paper | cm:content | No | <input type="button" value="Actions"/> <input checked="" type="checkbox"/> Layout Designer <input type="checkbox"/> Edit <input type="checkbox"/> Delete <input type="checkbox"/> Find Where Used |

- For the property type you want to edit, click **Edit** from the **Actions** drop-down list. The **Edit Aspect** window appears.

| Custom Types | | | | | | | | | | | | | | |
|---|---------------|------------|--------|-----------|------|---------------|--------|--------|---------|------------|---------|--|----|-----------|
| Name | Display Label | Parent | Layout | Actions | | | | | | | | | | |
| dc:whitePaper | White paper | cm:content | No | Actions ▾ | | | | | | | | | | |
| Aspects | | | | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th>Name</th><th>Display Label</th><th>Parent</th><th>Layout</th><th>Actions</th></tr> </thead> <tbody> <tr> <td>dc:webable</td><td>Webable</td><td></td><td>No</td><td>Actions ▾</td></tr> </tbody> </table> | | | | | Name | Display Label | Parent | Layout | Actions | dc:webable | Webable | | No | Actions ▾ |
| Name | Display Label | Parent | Layout | Actions | | | | | | | | | | |
| dc:webable | Webable | | No | Actions ▾ | | | | | | | | | | |

4. Edit the relevant properties.

- **For custom type:** For an inactive model, you can edit all the fields except **Name**. If a model is active, you can edit all the fields except for **Name** and **Parent Type**.
- **For aspects:** For an inactive model, you can edit all the fields except **Name**. If a model is active, you can edit all the fields except for **Name** and **Parent Aspect**.

5. Click **Save**.

Deleting custom types and aspects

The delete action on a custom type and aspect depends on the status of the model.

Prerequisites for deleting custom types and aspects:

| | |
|-------------------------|---|
| For aspects | <ul style="list-style-type: none"> If the model is inactive, you can simply delete the aspect by clicking Delete from the Actions drop-down list for the relevant aspect. If the model is active and the aspect has been applied to a file in Alfresco, you can't delete the aspect. In this case, you need to remove the aspect by navigating to Manage Aspects and removing the aspect from the Currently Selected list. Now, you can deactivate the model. |
| For custom types | <ul style="list-style-type: none"> If the model is inactive, you can simply delete the custom type by clicking Delete from the Actions drop-down list for the relevant type. If the model is active, you can't delete the custom type using the Model Manager. In this case, you have to delete the file from Alfresco Share and then permanently delete the file from Alfresco's trashcan. For more information, see Emptying your trashcan. |

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

The **Model Manager** page is displayed.

2. Click the relevant model from the **Custom Models** list.

The selected model page appears. This page shows the existing custom types and aspects associated with the selected model.

3. For the type or aspect you want to delete, click **Delete** from the **Actions** drop-down list.

| Custom Types | | | | | | | | | | | | | | |
|---|---------------|------------|--------|-----------|------|---------------|--------|--------|---------|------------|---------|--|----|-----------|
| Name | Display Label | Parent | Layout | Actions | | | | | | | | | | |
| dc:whitePaper | White paper | cm:content | No | Actions ▾ | | | | | | | | | | |
| Aspects | | | | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th>Name</th><th>Display Label</th><th>Parent</th><th>Layout</th><th>Actions</th></tr> </thead> <tbody> <tr> <td>dc:webable</td><td>Webable</td><td></td><td>No</td><td>Actions ▾</td></tr> </tbody> </table> | | | | | Name | Display Label | Parent | Layout | Actions | dc:webable | Webable | | No | Actions ▾ |
| Name | Display Label | Parent | Layout | Actions | | | | | | | | | | |
| dc:webable | Webable | | No | Actions ▾ | | | | | | | | | | |

If deleting a custom type, the **Confirm Custom Type Deletion** window appears.

If deleting an aspect, the **Confirm Aspect Deletion** window appears.

4. Click **Delete**.

For an inactive model, if a type (or aspect) refers to another type (or aspect) within the same model, then you can't delete the referenced type (or aspect).

For example, Model 1 comprises of Type 1 and Type 2. Within Model 1, Type 1 refers to Type 2, so you cannot delete Type 2 as it is being referenced by Type 1. But you can delete Type 1 as it is neither used nor referenced by another type within the same model.

Creating a new property for custom types and aspects

Properties are pieces of metadata associated with a particular custom type and / or aspect. Both types and aspects can have one or more properties.

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

The **Model Manager** page is displayed.

2. Click the relevant model from the **Custom Models** list.

The selected model page appears. This page shows the existing custom types and aspects associated with the selected model.

3. To create a new property, perform the following action:

- Under the **Custom Types** list, click the type for which you want to create the new property.

The property page relevant to the selected type is displayed. This is of the format
model name:custom type name.

- Under the **Aspects** list, click the aspect for which you want to create the new property.

The property page relevant to the selected aspect is displayed. This is of the format
model name:aspect name.

| Custom Types | | |
|---------------|---------------|------------|
| Name | Display Label | Parent |
| dc:whitePaper | White paper | cm:content |
| Aspects | | |
| Name | Display Label | Parent |
| dc:webable | Webable | |

4. Click **Create Property**.

The screenshot shows the Alfresco Model Manager interface. In the top navigation bar, the 'Model Manager' icon is selected. On the left, a sidebar lists various tools under 'Tools', with 'Model Manager' currently selected. The main content area displays a document titled 'Document - whitePaper'. At the top right of this area, there is a red box around the 'Create Property' button. Below it, a table titled 'Properties' is shown with one row: 'No properties found. Click Create Property to get started.' The top right corner of the main area has a status message 'Inactive Parent Model'.

The **Create Property** window appears.

5. Enter the details for the new property. Fields marked with an asterisk (*) are required.

The 'Create Property' dialog box is open. It contains the following fields:

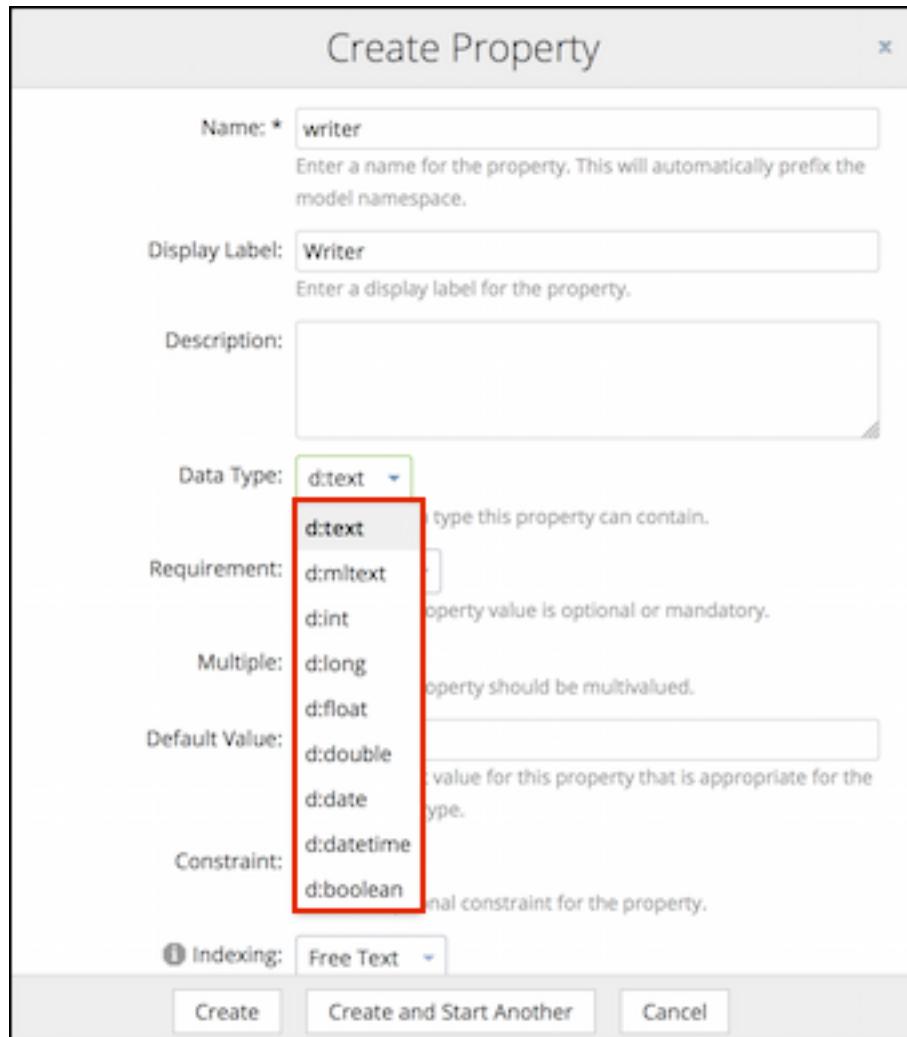
- Name:** * (highlighted with a green border)
- Display Label:** (empty input field)
- Description:** (empty text area)
- Data Type:** d:text (dropdown menu)
- Requirement:** Optional (dropdown menu)
- Multiple:** (checkbox)
- Default Value:** (empty input field)
- Constraint:** None (dropdown menu)
- Indexing:** Free Text (dropdown menu)

At the bottom are three buttons: 'Create' (highlighted with a red border), 'Create and Start Another', and 'Cancel'.

- a. Enter a name for the property.

Only alphanumeric characters, hyphens (-) and underscores (_) are allowed. The property name will automatically prefix the model namespace.

- b. Select an optional display label for the property.
The display label is shown as the property name to the end users in Alfresco Share.
- c. Enter an optional description of the property.
- d. Select the data type this property can contain.



Data types describe the fundamental types of data the repository will use to store properties. Alfresco supports a wide variety of data types. The available out-of-the-box data types are:

| Option | Description |
|----------|---|
| d:text | Specifies a text value or a character string. |
| d:mltext | Specifies a multilingual text value where many localized representations of the text value may be held. |
| d:int | Specifies an integer value. |
| d:long | Specifies a long value. This type is used when a wider range than int is needed. |

| Option | Description |
|------------|---|
| d:float | Specifies a float value. This data type is mainly used to save memory in large arrays of floating point numbers. |
| d:double | Specifies a double value. This data type is generally used as the default data type for decimal values. |
| d:date | Specifies a date value in the format dd/mm/yyyy. |
| d:datetime | Specifies a date and time value. |
| d:boolean | Specifies a boolean value which can either be true and false. This data type represents one bit of information and is used for simple flags that track true/false conditions. |

6. Select if the property value is optional or mandatory, and if this is enforced by Alfresco Share.

The available options are:

| Option | Description |
|------------------|--|
| Optional | Specifies that the property value is not required and the property can be left blank. |
| | |
| Mandatory | Specifies that the property value is required and must be filled if Edit Properties in Alfresco Share is to be completed. The property will be marked with an asterisk. |

7. Select if the property should be multi-valued.

The following table shows each data type's support for single or multiple values, with or without constraints.

| Data type | Supports single/multi-value properties | Constraints supported | Additional information |
|------------|--|--|--|
| d:text | Multi-value properties | <ul style="list-style-type: none"> • List of Values • Minimum / Maximum length • Regular expression | |
| d:mltext | Multi-value properties | <ul style="list-style-type: none"> • List of Values • Minimum / Maximum length • Regular expression | |
| d:int | Single-value properties in Model Manager
Multi-value properties in Alfresco Share | <ul style="list-style-type: none"> • List of Values • Minimum / Maximum value | In Alfresco Share, d:int does not support multiple values with constraint. |
| d:long | Single-value properties in Model Manager
Multi-value properties in Alfresco Share | <ul style="list-style-type: none"> • List of Values • Minimum / Maximum value | In Alfresco Share, d:long does not support multiple values with constraint. |
| d:float | Single-value properties in Model Manager
Multi-value properties in Alfresco Share | <ul style="list-style-type: none"> • List of Values • Minimum / Maximum value | In Alfresco Share, d:float does not support multiple values with constraint.

Also, d:float does not support the List of Values constraint in Alfresco Share. |
| d:double | Single-value properties in Model Manager
Multi-value properties in Alfresco Share | <ul style="list-style-type: none"> • List of Values • Minimum / Maximum value | In Alfresco Share, d:double does not support multiple values with constraint.

Also, d:double does not support the List of Values constraint in Alfresco Share. |
| d:date | Single-value properties | Certain data types should not be used with the List of Values constraints. | d:date does not support the List of Values constraint. |
| d:datetime | Single-value properties | Certain data types should not be used with the List of Values constraints. | d:datetime does not support the List of Values constraint. |
| d:boolean | Single-value properties | Certain data types should not be used with the List of Values constraints. | d:boolean does not support the List of Values constraint. |

If you want the end user to specify multiple values for the property in **Edit Properties** in Alfresco Share, then select the **Multiple** check box. For example, if you have created a property, **Languages known** and you want the end user to be able to specify multiple

answers for this property in Alfresco Share, then select the **Multiple** check box in the **Create Property** window in Model Manager.

- Specify an optional default value for this property.

The value specified in **Default Value** in Model Manager is displayed as the default property value in Alfresco Share. Also, the default value should be appropriate for the selected data type. For example, if the data type is `d:int`, the default value must be an integer.

The control/layout of **Default Value** depends on the selected data type. For example, if you select `d:text` as the data type, then the Default Value layout is a text box, whereas if you select `d:int` as the data type, then the Default Value layout is a spin control.

- Select the optional constraint for the property.

Depending on what constraint option you select, some additional fields are displayed.

| Option | Additional fields | Condition/Example |
|-------------|------------------------|------------------------|
| None | No constraint applied. | No constraint applied. |

| Option | Additional fields | Condition/Example |
|---------------------------------|---|---|
| Regular expression | Regular Expression: Specify a regular expression for the constraint. This field is mandatory. | Use this constraint if you want the property value specified in Alfresco Share to match the expression specified in the Model Manager.

For example, if the constraint expression specified in the Model Manager is <code>[a-z]*</code> , then Alfresco Share will accept any value in the range of lowercase a to z. |
| Minimum / Maximum length | <ul style="list-style-type: none"> Minimum Length: Specify the minimum length allowed for the property. Maximum Length: Specify the maximum length allowed for the property. Both the fields are mandatory. | For example, if the minimum length and maximum length are 1 and 5, respectively, then Alfresco Share will accept any text or integer value in the range of one to five characters or numbers. |
| Minimum / maximum value | <ul style="list-style-type: none"> Minimum Value: Specify the minimum value allowed for the property. Maximum Value: Specify the maximum value allowed for the property. Both the fields are mandatory. | For example, if the Minimum value and maximum value are 1 and 3, respectively, then Alfresco Share will accept either 1, 2, or 3 as the property value. |
| List of Values | <ul style="list-style-type: none"> List of Values: Enter the list of allowed values, with each new item on a new line. This field is mandatory. | For example, if you specify A, B, and C in List of Values , then Alfresco Share will display these values in a drop-down list. |
| Java class | <ul style="list-style-type: none"> Class name: Enter the fully qualified name of the class to use as a constraint. | Used for custom constraints implemented in Java. |

10. To enable the property to be used for searching, select the appropriate searching option from the **Indexing** drop-down list.

Depending on the data type selected, the following searching options are supported:

| Searching option | Description | Searchable | Supported data types |
|---------------------------------------|---|------------|---|
| None | Search is not supported. Use this option if you do not want to use a property for searching. | No | Non-text data types: <ul style="list-style-type: none"> • int • long • float • double • date • datetime • boolean Text date types: <ul style="list-style-type: none"> • text • mltext • content |
| Basic | Property is searchable but the values will not be available in the search result filters. | Yes | Non-text data types: <ul style="list-style-type: none"> • int • long • float • double • date • datetime • boolean |
| Enhanced search | Use enhanced search if you want to use the property in faceting, stats, sort, and range queries. While this indexing option improves query performance and reduces memory usage, it also requires more disk space for the search index. | Yes | Non-text data types: <ul style="list-style-type: none"> • int • long • float • double • date • datetime |
| Free text | Property is searchable but the values will not be available in the search result filters. | Yes | Text date types: <ul style="list-style-type: none"> • text • mltext • content |
| List of values - whole match | This indexing option enables you to filter on a property in the search results while searching for the whole term. | Yes | Text date types: <ul style="list-style-type: none"> • text • mltext • content |
| List of values - partial match | This indexing option enables you to filter on a property in the search results while searching the property using wildcard characters. | Yes | Text date types: <ul style="list-style-type: none"> • text • mltext • content |

| Searching option | Description | Searchable | Supported data types |
|---------------------------------|--|------------|--|
| Pattern - unique matches | This indexing option enables you to use unique identifiers which are searched on the basis of the full value of the property. The property itself will not be shown as a filter in the search results. | Yes | Text date types: <ul style="list-style-type: none">• text• mltext• content |
| Pattern - many matches | This indexing option enables you to use identifiers which could be searched on the basis of the full value or via the wild card characters. The property itself will not be shown as a filter in the search results. | Yes | Text date types: <ul style="list-style-type: none">• text• mltext• content |

-  Different values can have an impact on the search performance, memory requirement, and disk storage requirement for your installation. For very large repositories, the values can have a significant impact; however, for most installations the default settings are fine. For properties that will be used for search filters, it is important to use the correct value as shown in the above table.
-  For more information on setting up search filters using the Search Manager, see [Search Manager](#).

11. To create a new property, click **Create**. To save and create another property, click **Create and Start Another** or click **Cancel** if you do not want to save the changes.

The property created appears in the **Properties** table for the selected type or aspect. The property name is of the format, prefix:property name.

Additional information displayed in the table includes the property name, display label, data type, requirement, default value, multivalued, and the actions (Edit and Delete) that can be performed on the property.

Editing a property for custom types and aspects

Use this information to edit a type property.

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

The **Model Manager** page is displayed.

2. Click the relevant model from the **Custom Models** list.

The selected model page appears. This page shows the existing custom types and aspects associated with the selected model.

3. To edit a property, perform the following action:

- Under the **Custom Types** list, click the property you want to edit.

The property page relevant to the selected type is displayed. This is of the format model name:custom type name.

- Under the **Aspects** list, click the property you want to delete.

The property page relevant to the selected aspect is displayed. This is of the format model name:aspect name.

4. For the property you want to edit, click **Edit** from the **Actions** drop-down list.

The **Edit Property** window appears.

5. Edit the relevant properties.

For a model with **Inactive** status, you can edit all the properties except **Name**. If a model is **Active**, you can edit all the properties except for **Name**, **Data Type**, **Requirement**, and **Multiple**.

6. Click **Save**.

Deleting a property for custom types or aspects

The delete action on a property depends on the status of the model and whether or not the property has been applied to a file in Alfresco Share.

Prerequisites for deleting a property for custom types and aspects:

| | |
|-------------------------|--|
| For aspects | <ul style="list-style-type: none"> • If the model is inactive, you can simply delete the property by clicking Delete from the Actions drop-down list for the relevant property. • If the model is active and the aspect has been applied to a file in Share, you can delete the property by clicking Delete from the Actions drop-down list for the relevant property. The aspect will still be applied to the file but the deleted property will no longer be visible on the Edit Properties page in Alfresco Share. <p> When deleting a property, remember to update the Layout Designer and delete any search filter that you may have created for that property.</p> |
| For custom types | <ul style="list-style-type: none"> • If the model is inactive, you can simply delete the property by clicking Delete from the Actions drop-down list for the relevant property. • If the model is active, you can delete the property if it was created after the type was applied to a file in Alfresco. Once the property is applied to a file, it cannot be deleted using the Model Manager. In this case, you need to delete the file from Alfresco Share and then permanently delete the file from Alfresco's trashcan. • If a model's custom type is applied to a file in Alfresco Share, then the associated properties: <ul style="list-style-type: none"> • can be deleted, if the user does not edit or save the properties via the Edit Properties option in Alfresco Share. • cannot be deleted, if the property is created with the default value and then the type is applied to a file on Alfresco Share. • can be deleted, if the property is created for a type which is already applied to the file in Alfresco Share. |

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

The **Model Manager** page is displayed.

2. Click the relevant model from the **Custom Models** list.

The selected model page appears. This page shows the existing custom types and aspects associated with the selected model.

3. To delete a property, perform the following action:

- Under the **Custom Types** list, click the type whose property you want to delete.

The property page relevant to the selected type is displayed. This is of the format
model name:custom type name.

- Under the **Aspects** list, click the aspect whose property you want to delete.

The property page relevant to the selected aspect is displayed. This is of the format
model name:aspect name.

4. For the property you want to delete, click **Delete** from the **Actions** drop-down list.

The **Delete Property** window appears.

5. Click **Delete**.

Configuring models - custom types, aspects, and properties

Depending on the status of the model, the Model Manager tool limits which fields you can edit while configuring or editing models.

For an Inactive Model, you can edit all the fields except for `Name`. For an Active Model, the following table shows the list of fields that you can or can't edit for a given custom type, aspect, and property.

| | Active Model | | |
|-----------------------|--------------|---------|------------|
| | Custom types | Aspects | Properties |
| Constraint | | | ✓ |
| Data Type | | | ✗ |
| Default Value | | | ✓ |
| Description | ✓ | ✓ | ✓ |
| Display Label | ✓ | ✓ | ✓ |
| Indexing | | | ✗ |
| Multiple | | | ✗ |
| Name | ✗ | ✗ | ✗ |
| Parent Property Group | | ✗ | |
| Parent Type | ✗ | | |
| Requirement | | | ✗ |

✓ = can edit

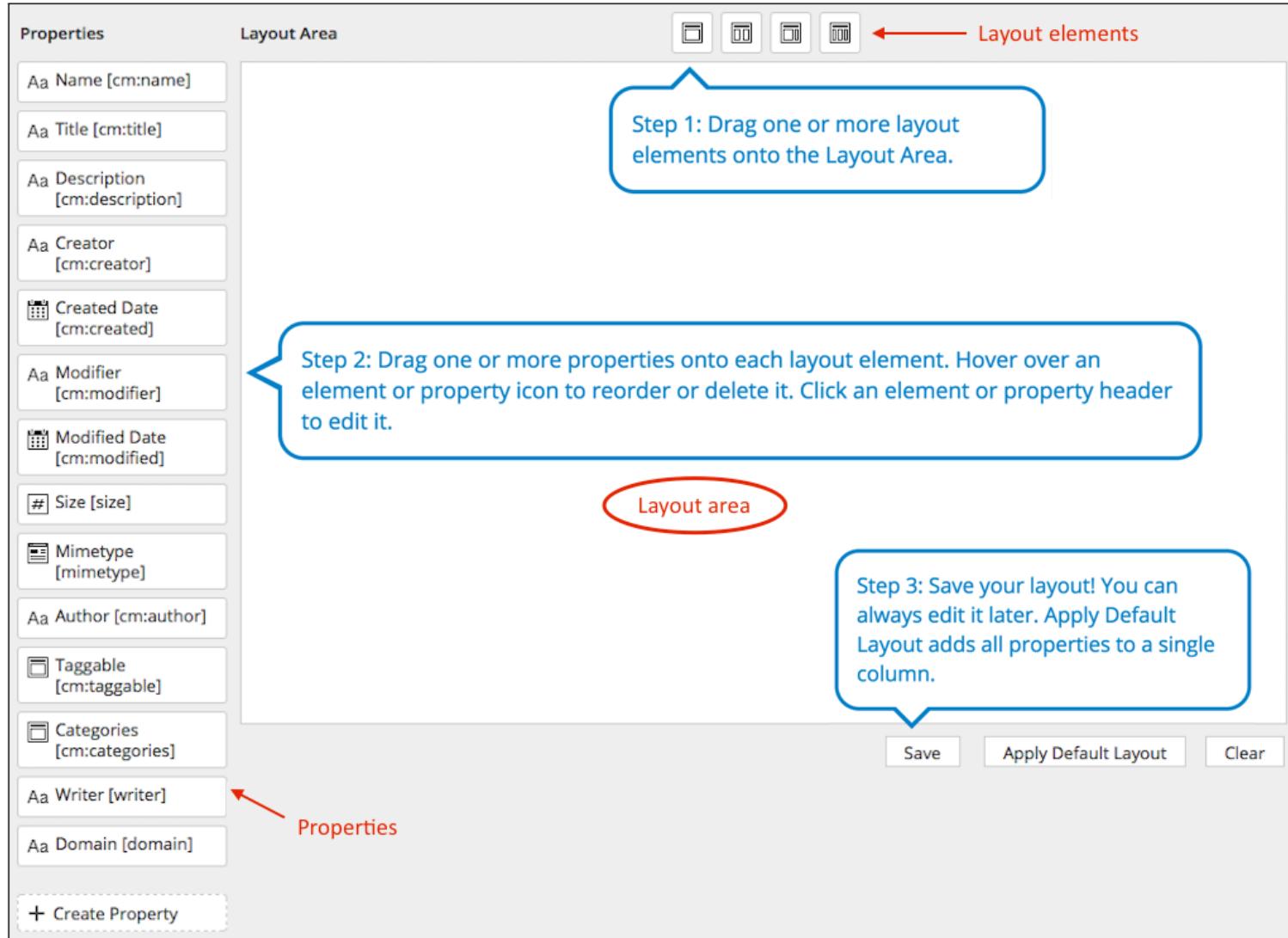
✗ = can't edit after the value is saved

Using the Layout Designer

Use the Layout Designer to define how the properties you create using the Model Manager are displayed on the **Edit Properties** page in Alfresco Share.

The Layout Designer provides a simple, visual representation of your page layout in the editor using horizontal sections and vertical columns. By adding multiple sections with different column configurations you can build quite complex layouts very easily. The Layout Designer consists of a layout area in the center, the layout elements on the top, and the properties arranged vertically on the left of the layout area.

Components of the Layout Designer



Layout element

The layout elements render heading text and panel arrangement with one, two or three rows. These elements are reusable and can be used multiple times on the layout area. There are four different options in the layout elements:

- Single column panel
- Double column panel
- Wide left double column panel
- Triple column panel

Properties

The properties are intended for single use only. To use the properties, drag them from the side onto the elements. Once the is used, it is automatically removed from the left panel.

For information on how to create properties, see [creating a new property for custom type or aspect](#).

Create Property

Enables you to create a new property from inside the Layout Designer using the **Create Property** window.

To use the Property Layout Designer, follow the steps below:

1. Click **Admin Tools**, and then click **Model Manager**.
The **Model Manager** page is displayed.
 2. Click the relevant model from the **Custom Models** list.
The selected model page appears. This page shows the existing custom types and aspects associated with the selected model.
 3. To define how properties appear on the **Edit Properties** page in Alfresco Share, click **Layout Designer** from the **Actions** drop-down list for the relevant type or aspect.
-  If you do not apply the newly created property to the Layout Designer, then neither the type nor its property will be available for use in Alfresco Share.
- The Layout Designer page is displayed.
4. Drag the required layout elements from top onto the layout area.
After adding the layout elements onto the layout area, you can either edit, reorder, or delete the elements.
 - Edit: To edit the layout elements, click anywhere on the element's top panel showing the element name. This displays the **Edit Properties** window, which enables you to change the column configuration, specify the panel label, and select the panel appearance.
 - Reorder: To reorder the layout elements on the layout area, hover over  or  on the element's top right corner and select  or  to move the element up or down a place.
 - Delete: To remove the layout element from the layout area, hover over  on the element's top right corner and select .
 5. Drag the required properties from the side onto the elements.

After adding the properties onto the layout elements, you can either edit, reorder, or delete the elements.

- Edit: To edit the property, click anywhere on the property. This displays the **Edit Properties** page, which displays the property name, label, and type information. It also enables you to manage the following attributes:
 - *Form control*: Select the type of form control shown to the end user in Alfresco Share. The form controls displayed in the Layout Designer depend on the data type of the property.

| Form control | Description | Supported data type |
|--------------|--|---|
| Default | Allows the user to enter a value based on the selected data type. For example, if the selected data type is <code>d:int</code> , the user will have to specify an integer value. | <ul style="list-style-type: none"> • int • text |
| Number | Allows the user to enter a <code>int</code> number. | |

| Form control | Description | Supported data type |
|----------------|---|---------------------|
| Text field | Allows the user to type a small amount of text. If you need to obtain more than one line of input from the user, use a text area. | text |
| Text area | Allows multi-line text input and can hold an unlimited number of characters. | text |
| Rich text | Allows text to be formatted with common formatting options, such as bold and italics. | text |
| Password field | Displays characters as masked, such as asterisks or circles. | text |
| Mimetype | Enables you to identify files mimetype based on their nature and format. | |
| Categories | Enables you to organize and categorize your content into related groups to form a hierarchy. | cm:categories |
| Taggable | Enables tagging of content items using keywords. | cm:taggable |

- **View mode:** Specifies how the property should be displayed in Alfresco Share.

| View mode options | Description |
|-------------------|--|
| Any | Displays the property on the details page under Properties and also on the Edit Properties page. |
| View | Displays the property only on the details page under Properties . |
| Edit | Displays the property only on the Edit Properties page. |

- **Style:** Enables you to impart text formatting options to the property when it is displayed in Alfresco Share. The following options are available: bold, underline, italics, font colour, and background colour.
- **Style class:** Enables you to add your own css class.
- **Read only:** Select this checkbox to make the property read-only in Alfresco Share. The **Read only** option overwrites the **View mode** option.
- **Force display:** For the 'standard' content type properties, **Force display** ensures that the property is visible on view/edit forms even if the type does not actually have that property applied from its aspect. An example of this is **titled**, which is a part of the `cm:titled` aspect that is not applied to a file until the property is first set.
- **Hidden:** Enables you to set the value of the property but it will not visible to the user in Alfresco Share. For example, you need a property with a default value that must always be applied. In this case, you can create a property with some default value and mark it as **Hidden** so it is not visible to the user but its value is still set by the system.

- Reorder: To reorder the property on the layout element, hover over  on the property and select  or  to move the property up or down a place.
 - Delete: To remove the property from the layout element, hover over  on the property and select . The deleted property reappears in the **Properties** list on the Layout Designer.
6. To save your design, click **Save** or click **Clear** to clear the layout area. To apply the default layout design, click **Apply Default Layout**.

To enable the properties layout design in Alfresco Share, you must activate and apply the relevant model to file(s) in Alfresco Share.

Tutorial: creating and using models in Alfresco Share

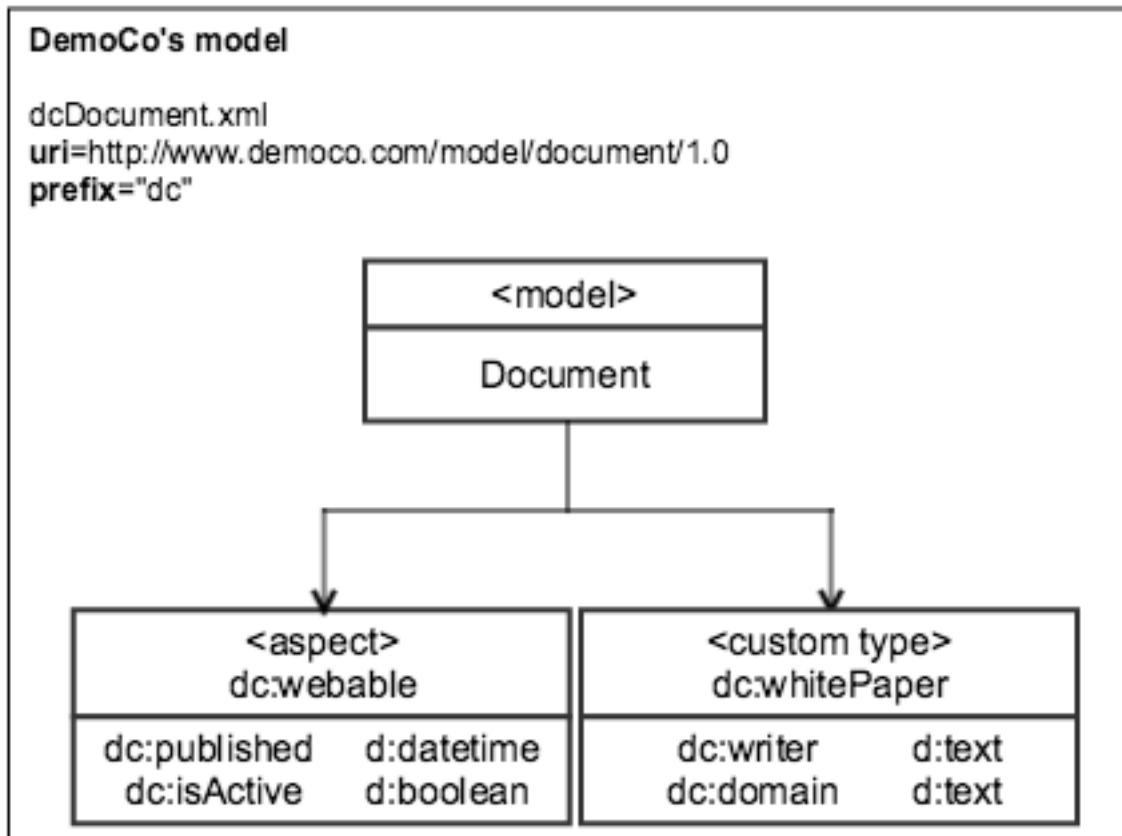
In this tutorial you will learn how to create custom models and implement content modeling in Alfresco using the Model Manager.

You will perform a step-by-step walk-through of creating and using models, custom types, aspects, and their properties in Alfresco Share.

In this tutorial we assume that Alfresco is implemented in a fictitious company called DemoCo as their ECM solution for managing content on their new website. DemoCo wants to add new types of content (for example, white papers) to their website and track metadata for all their files in the Alfresco repository.

For this purpose, DemoCo needs to create a new model, *Document* with a custom type called *whitePaper*. If a document is a white paper, then DemoCo would want to capture its writer's name and domain area. Also, for each document, DemoCo wants to determine if the content needs to be shown on their website. If yes, there should be a flag to indicates that the content is active and the date when the content was set to active.

So, let's start by creating a new model called *Document*. It has a custom type - *whitePaper* with properties *writer* and *domain*. The model has an aspect - *Webable* with properties *published* and *isActive*.



- Step 1: Navigate to Model Manager
- Step 2: Create a new model, *Document*
- Step 3: Create a new type, *whitePaper*
- Step 4: Create new type properties, *writer* and *domain*
- Step 5: Add new type properties to the Layout Designer
- Step 6: Create new aspect, *webable*
- Step 7: Create new aspect properties, *published* and *isActive*
- Step 8: Add new aspect properties to the Layout Designer
- Step 9: Activate the model, *Document*
- Step 10: Apply type (*whitePaper*) and aspect (*webable*) to a file in Alfresco Share

Step 1: Navigate to Model Manager

1. Click **Admin Tools**.
2. Under **Tools**, click **Model Manager**.

The screenshot shows the Model Manager interface. On the left, a sidebar titled 'Tools' lists various management options: Application, Category Manager, Node Browser, Tag Manager, Model Manager (which is highlighted with a red box), and Sites Manager. Below these are Repository, Replication Jobs, Users and Groups, Groups, and Users. The main content area has a heading 'Models' and a sub-section 'Models'. It displays a table with columns 'Name' and 'Namespace'. A message at the top of the table says 'No models found. Click Create Model to get started.' Below the table are two buttons: 'Create Model' (in blue) and 'Import Model'.

The **Model Manager** page is displayed.

Step 2: Create a new model, *Document*

1. Click **Create Custom Model**.

In the **Create Model** window, specify the following:

- Namespace: `http://www.democo.com/model/document/1.0`
- Prefix: dc
- Name: Document

Create Model

Namespace: * Enter a unique model namespace. For example, if you're creating a model for financial documents you might use a URI similar to http://www.mycompany.com/model/finance/1.0.

Prefix: * Enter a namespace prefix for the model. This is shorthand for the namespace URI and is used as the namespace prefix and for advanced search property syntax.

Name: * Enter a name for the model.

Creator: Enter an optional creator of the model.

Description:

2. Click **Create**.

The new model, *Document* appears in the **Custom Models** table on the **Model Manager** page. The current status of the model is **Inactive**.

[back to top](#)

Step 3: Create a new type, *whitePaper*

1. Click the model, *Document*.

The *Document* model page is displayed.

2. To create a type - *whitePaper*, click **Create Custom Type**.

The screenshot shows the Model Manager interface with the 'Document' tab selected. On the left, a sidebar titled 'Tools' lists various management options. The 'Model Manager' option is currently selected. The main area displays sections for 'Custom Types' and 'Aspects'. Under 'Custom Types', there is a table with columns 'Name', 'Display Label', and 'Parent'. A message indicates 'No types found. Click Create Custom Type to get started.' Below this is another table for 'Aspects' with similar columns, also indicating 'No aspects found. Click Create Aspect to get started.' At the top right of the main area, there are two buttons: 'Create Custom Type' (highlighted with a red box) and 'Create Aspect'.

The **Create Custom Type** window appears.

3. In the **Name** field, enter `whitePaper`.
4. In the **Display Label** field, enter `White Paper`.
5. Specify other optional fields, if required.

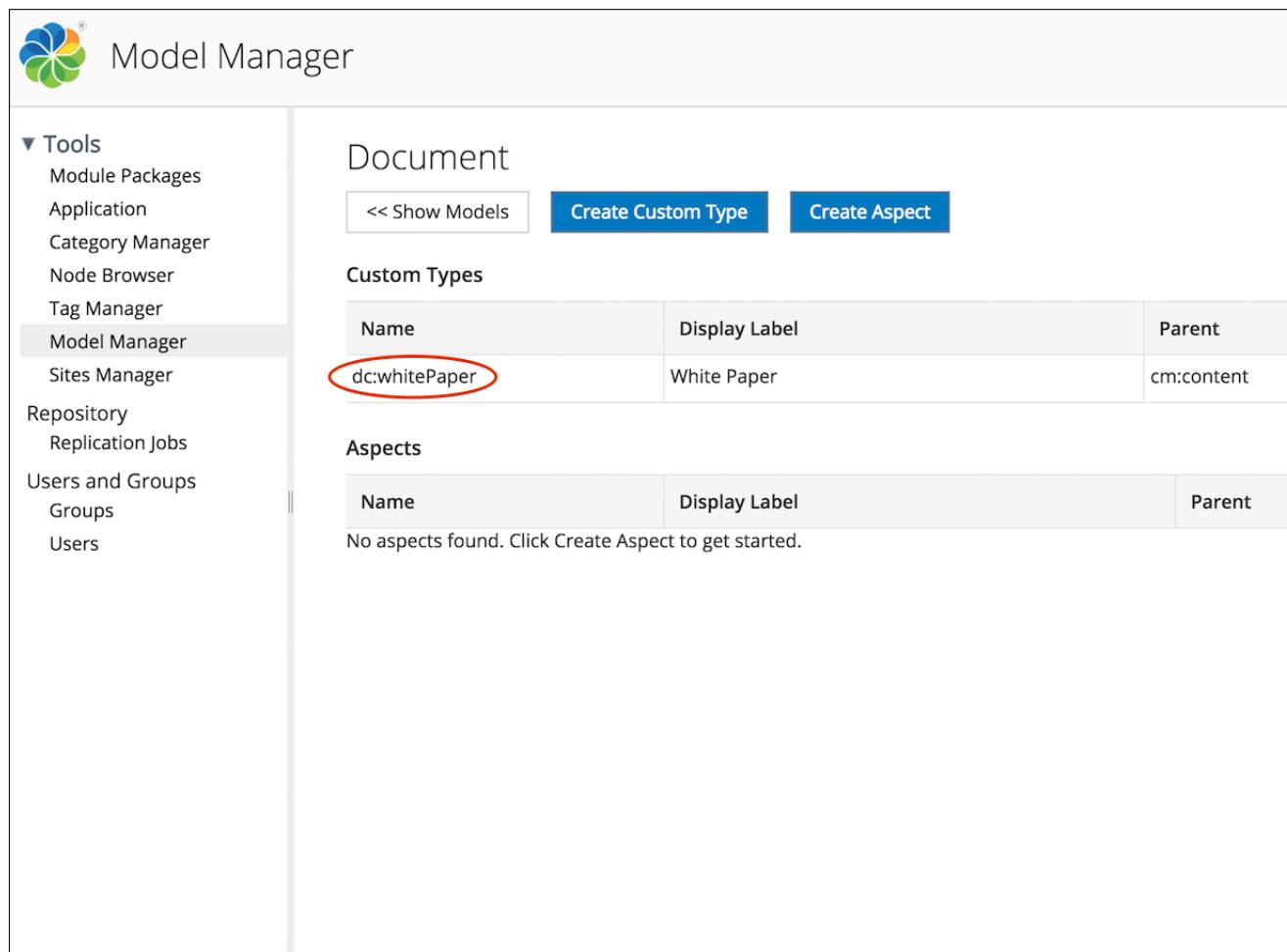
The 'Create Custom Type' dialog box is shown. It has fields for 'Name' (set to `whitePaper`), 'Parent Type' (set to `cm:content (Content)`), 'Display Label' (set to `White Paper`), and a 'Description' field which is empty. At the bottom, there are 'Create' and 'Cancel' buttons, with 'Create' being highlighted with a red box.

6. Click **Create**.

The new custom type, *dc:whitePaper* appears in the **Custom Types** table.

Step 4: Create new type properties, *writer* and *domain*

- Under the **Custom Types** list, click *dc:whitePaper*.



The screenshot shows the Alfresco Model Manager interface. On the left, there's a sidebar with various tools like Module Packages, Application, Category Manager, etc. The 'Model Manager' option is selected. The main area is titled 'Document' and contains two tables: 'Custom Types' and 'Aspects'. The 'Custom Types' table has columns for Name, Display Label, and Parent. It shows one entry: 'dc:whitePaper' (with the 'Name' field circled in red), 'White Paper' (Display Label), and 'cm:content' (Parent). The 'Aspects' table is currently empty, showing the message 'No aspects found. Click Create Aspect to get started.'

| Name | Display Label | Parent |
|---------------|---------------|------------|
| dc:whitePaper | White Paper | cm:content |

2. Click **Create Property**.

3. Enter the details of the new property, *writer*:

- In the **Name** field, enter *writer*.
- In the **Display Label** field, enter *Writer*.
- From the **Data Type** drop-down list, select **d:text**.
- To make the property mandatory in Alfresco Share, select **Mandatory** from the **Requirement** drop-down list.
- Specify other optional fields, if required.

Create Property

| | | |
|---|-------------------------------------|--|
| Name: * | writer | Enter a name for the property. This will automatically prefix the model namespace. |
| Display Label: | Writer | Enter a display label for the property. |
| Description: | | |
| Data Type: | d:text | Select the data type this property can contain. |
| Requirement: | Mandatory | Select if the property value is optional or mandatory. |
| Multiple: | <input checked="" type="checkbox"/> | Select if the property should be multivalued. |
| Default Value: | | |
| Constraint: | None | |
| Indexing: | Free Text | |
| Select whether the property can be searched on and what search | | |
| <input type="button" value="Create"/> <input style="background-color: red; color: white; border: 1px solid red;" type="button" value="Create and Start Another"/> <input type="button" value="Cancel"/> | | |

4. Click **Create and Start Another**.
5. Enter the details of the new property, *domain*:

- In the **Name** field, enter domain.
- In the **Display Label** field, enter Domain.
- From the **Data Type** drop-down list, select **d:text**.
- From the **Constraints** drop-down list, select **List of Values**.
- In the **List of Values** field, enter Engineering, Marketing, HR, Sales, Finance, and Operations in separate lines.
- Select **Sort Alphanumerically** to display the above specified values in an alphabetical order in Alfresco Share.
- Specify other optional fields, if required.

Create Property

Display Label: Domain
Enter a display label for the property.

Description:

Data Type: d:text
Select the data type this property can contain.

Requirement: Optional
Select if the property value is optional or mandatory.

Multiple:
Select if the property should be multivalued.

Default Value:

Constraint: List of Values
Select an optional constraint for the property.

List of Values: * Engineering
Marketing
HR
Sales
Finance
Enter the list of allowed values, with each new item on a new line.

Sort Alphanumerically:

Create **Create and Start Another** **Cancel**

6. Click **Create**.

[back to top](#)

Step 5: Add new type properties to the Layout Designer

1. Navigate to the page displaying the custom types list.
2. Click **Layout Designer** from the **Actions** drop-down list for the type, *dc:whitePaper*.

The screenshot shows the Model Manager interface with the following details:

- Tools:** Module Packages, Application, Category Manager, Node Browser, Tag Manager, Model Manager (selected), Sites Manager, Repository, Replication Jobs, Users and Groups, Groups, Users.
- Document:** Buttons: << Show Models, Create Custom Type (highlighted in blue), Create Aspect.
- Custom Types:** Table with columns: Name, Display Label, Parent.

| Name | Display Label | Parent |
|---------------|---------------|------------|
| dc:whitePaper | White Paper | cm:content |
- Aspects:** Table with columns: Name, Display Label, Parent.

| Name | Display Label | Parent |
|---|---------------|--------|
| No aspects found. Click Create Aspect to get started. | | |

The Layout Designer page is displayed.

3. Drag the double column panel layout from top onto the layout area.
4. To specify the panel label, click anywhere on the element's top panel displaying the label, **double column panel**.

This displays the **Edit Properties** window.

5. In the **Label** field, enter `Details`.
6. Configure other optional fields, if required.
7. Drag the properties, *writer* and *domain* onto the **Details** element.
8. To configure the properties, click anywhere on the property.
9. On the Layout Designer, click **Save**.

Document - whitePaper

<< Show Types and Aspects

Use the Layout Designer to define how properties you create are displayed in Alfresco.

Properties

- Aa Name [cm:name]
- Aa Title [cm:title]
- Aa Description [cm:description]
- Aa Creator [cm:creator]
- Created Date [cm:created]
- Aa Modifier [cm:modifier]
- Modified Date [cm:modified]
- # Size [size]
- Mimetype [mimetype]
- Aa Author [cm:author]
- Taggable [cm:taggable]
- Categories [cm:categories]

Layout Area

Icon: Details

Property: Aa Writer [writer]

Property: Aa Domain [domain]

Buttons: Save (highlighted), Apply

+ Create Property

Step 6: Create new aspect, `webable`

1. Navigate to the **Model Manager** page.
2. Click the model, **Document**.

The *Document* model page is displayed.

3. To create new aspect - webable, click **Create Aspect**.

The **Create Aspect** window appears.

4. In the **Name** field, enter `webable`.
5. In the **Display Label** field, enter `Webable`.
6. Specify other optional fields, if required.
7. Click **Create**.

Create Aspect

| | | |
|---|---------|--|
| Name: * | webable | Enter a name for the aspect. This will automatically prefix the model namespace. |
| Parent Aspect: | None | Select an optional parent aspect. The aspect inherits all properties and aspects assigned to the parent. |
| Display Label: | Webable | Enter a display label for the aspect. |
| Description: | | |
| <input type="button" value="Create"/> <input type="button" value="Cancel"/> | | |

The new aspect, `dc:webable` appears in the **Aspects** table.

The screenshot shows the Model Manager interface with the 'Model Manager' tab selected in the sidebar. The main area displays the 'Document' section with 'Custom Types' and 'Aspects' tables. The 'Aspects' table has a single row with the following data:

| Name | Display Label | Parent |
|------------|---------------|--------|
| dc:webable | Webable | |

The row for 'dc:webable' is circled in red, indicating it is the newly created aspect. The 'Create Custom Type' and 'Create Aspect' buttons are visible above the tables.

[back to top](#)

Step 7: Create new aspect properties, `published` and `isActive`

- Under the **Aspects** list, click `dc:webable`.

2. Click **Create Property**.
3. Enter the details of the new property, *published*:
 - In the **Name** field, enter *published*.
 - In the **Display Label** field, enter *Published*.
 - From the **Data Type** drop-down list, select **d:datetime**.
 - Specify other optional fields, if required.
4. Click **Create and Start Another**.

The screenshot shows the 'Create Property' dialog box. The 'Name' field contains 'published'. The 'Display Label' field contains 'Published'. The 'Data Type' dropdown is set to 'd:datetime'. Other fields like 'Description', 'Requirement', 'Multiple', 'Default Value', 'Constraint', and 'Indexing' are also visible but not highlighted.

| Create Property | |
|---|-------------------------------------|
| Name: * | published |
| Enter a name for the property. This will automatically prefix the model namespace. | |
| Display Label: | Published |
| Enter a display label for the property. | |
| Description: | |
| Data Type: | d:datetime |
| Select the data type this property can contain. | |
| Requirement: | Optional |
| Select if the property value is optional or mandatory. | |
| Multiple: | <input checked="" type="checkbox"/> |
| Select if the property should be multivalued. | |
| Default Value: | |
| Enter a default value for this property that is appropriate for the selected data type. | |
| Constraint: | None |
| Select an optional constraint for the property. | |
| Indexing: | Basic |
| Select whether the property can be searched on and what search | |
| <input type="button" value="Create"/> <input type="button" value="Create and Start Another"/> <input type="button" value="Cancel"/> | |

5. Enter the details of the new property, *isActive*:
 - In the **Name** field, enter *isActive*.
 - In the **Display Label** field, enter *Is Active*.
 - From the **Data Type** drop-down list, select **d:boolean**.
 - From the **Default Value** drop-down list, select **False**.
 - Specify other optional fields, if required.
6. Click **Create**.

[back to top](#)

Step 8: Add new aspect properties to the Layout Designer

1. Navigate to the page displaying the aspects list.
2. Click **Layout Designer** from the **Actions** drop-down list for the aspect, **dc:webable**.

The screenshot shows the Model Manager interface. On the left, there's a sidebar with a navigation tree under 'Tools'. The 'Model Manager' item is selected and highlighted with a grey background. The main content area has two sections: 'Custom Types' and 'Aspects'. In the 'Custom Types' section, there is one entry: 'dc:whitePaper' with 'White Paper' as the display label and 'cm:content' as the parent. In the 'Aspects' section, there is one entry: 'dc:webable' with 'Webable' as the display label. The entry 'dc:webable' is circled with a red oval.

| Name | Display Label | Parent |
|---------------|---------------|------------|
| dc:whitePaper | White Paper | cm:content |

| Name | Display Label | Parent |
|------------|---------------|--------|
| dc:webable | Webable | |

The Layout Designer page is displayed.

3. Drag the single column panel layout from top onto the layout area.
4. To specify the panel label, click anywhere on the element's top panel displaying the label, *single column panel*.

This displays the **Edit Properties** window.

5. In the **Label** field, enter `Publication details`.
6. Configure other optional fields, if required.
7. Drag the properties, `published` and `isActive` onto the **Publication details** element.
8. To configure the properties, click anywhere on the property.
9. On the Layout Designer, click **Save**.

Document - webable

<< Show Types and Aspects

Use the Layout Designer to define how properties you create are displayed in Alfresco.

Properties Layout Area

+ Create Property

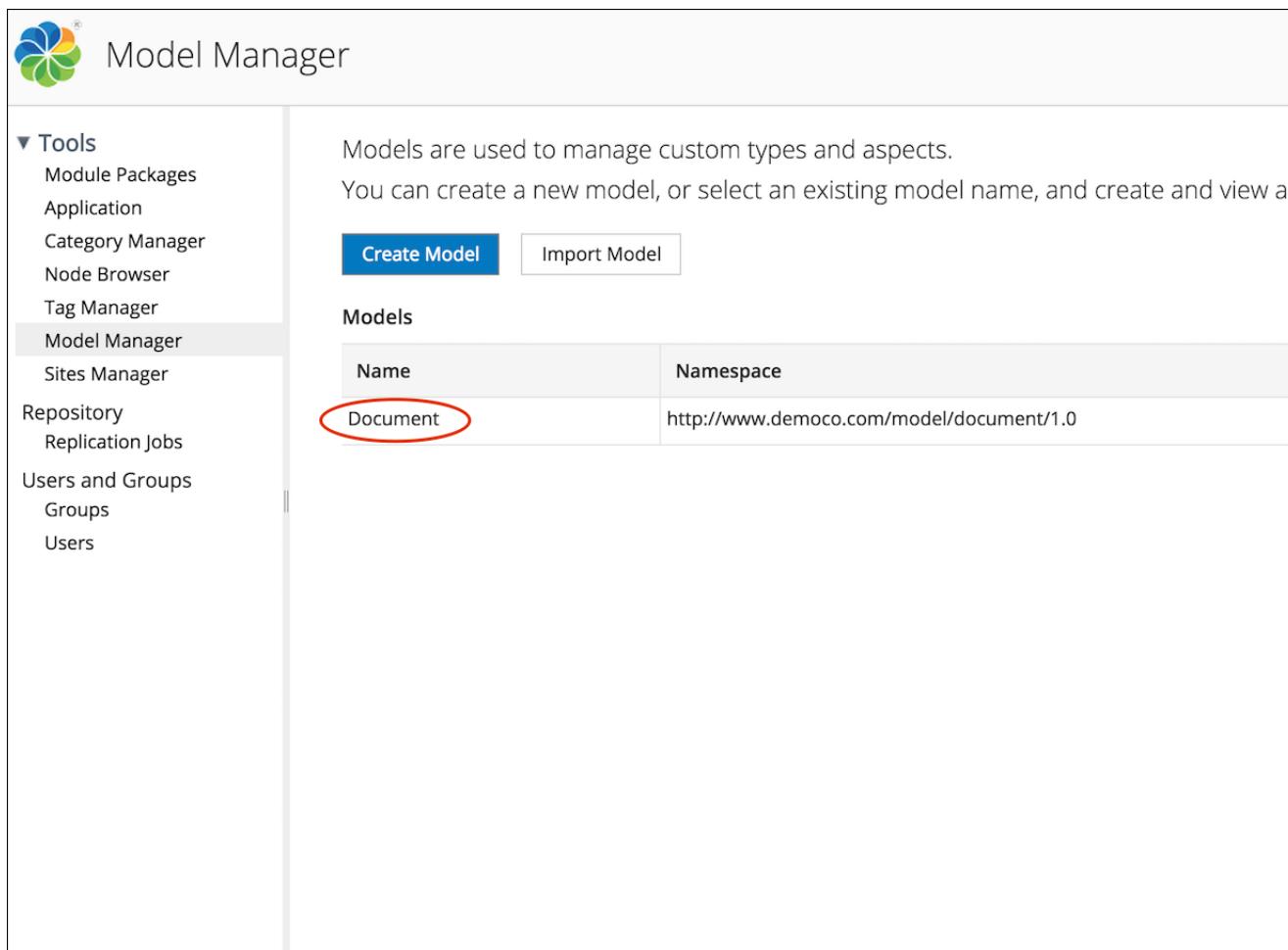
Publication details
✓ Is Active [isActive]
Published [published]

Save Apply

The screenshot shows the 'Layout Area' configuration for a 'Document - webable' model. It displays three properties: 'Publication details', 'Is Active [isActive]', and 'Published [published]'. Each property is represented by a small icon followed by its name in brackets. The entire 'Layout Area' section is enclosed in a dashed box. At the bottom right of the screen, there are 'Save' and 'Apply' buttons, with the 'Save' button being highlighted by a red box.

Step 9: Activate the model, *Document*

1. Navigate to the **Model Manager** page.
2. Click the **Actions** drop-down list for the model, *Document*.
3. Click **Activate**.



The screenshot shows the Alfresco Model Manager interface. On the left, there's a sidebar with a 'Tools' section containing links like Module Packages, Application, Category Manager, Node Browser, Tag Manager, Model Manager (which is selected), Sites Manager, Repository, Replication Jobs, Users and Groups, Groups, and Users. Below this is a large empty white area. On the right, there's a 'Models' section with a 'Create Model' button and an 'Import Model' button. A table titled 'Models' lists one entry: 'Name' (Document) and 'Namespace' (http://www.democo.com/model/document/1.0). The 'Document' entry is circled in red.

The status of **Document** changes to **Active**.



This screenshot shows the same 'Models' section as before, but now the 'Status' column for the 'Document' row is highlighted with a green border, indicating it is active. The 'Actions' column also has a green border.

Now that we have created a new model with its custom type and aspect, let's see how we can use it to capture a file's metadata in Alfresco Share.

To do so, follow the steps below:

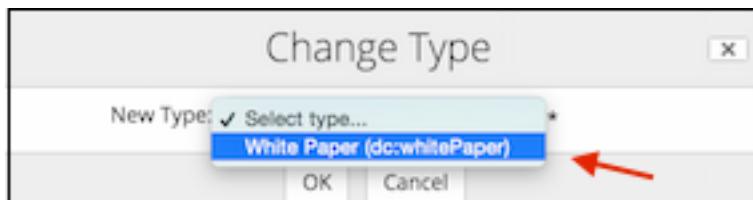
Step 10: Apply type (*whitePaper*) and aspect (*webable*) to a file in Alfresco Share

1. In your site, click **Document Library** to access the library.
2. Click on a file in the library to view it in the file preview screen.
3. To apply the type, perform the following steps:
 - a. Under **Document Actions**, click **Change Type**.

Configuring

The screenshot shows the Alfresco interface for managing documents. At the top, there's a navigation bar with links like Home, My Files, Shared Files, Sites, Tasks, People, Repository, and Admin Tools. Below the navigation bar is a title bar for a 'Sample: Web Site Design Project' document, showing it was modified by Administrator on Fri 21 Aug 2015 16:48:37. The main area displays the document content: 'This is a sample document. This is a sample document.' To the right of the document preview is a vertical sidebar titled 'Document Actions' containing various options: Download, View In Browser, Edit Properties (which is highlighted with a red box), Upload New Version, Inline Edit, Edit Offline, Copy to..., Move to..., Edit in Google, Delete Document, Start Workflow, Manage Permissions, Manage Aspects, and Change Type (which is also highlighted with a red box).

- b. In the **Change Type** window, select *whitePaper (dc:whitePaper)* from the **New Type** drop-down list.



- c. Click **OK**.

The type (*dc:whitePaper*) and its properties (*writer* and *domain*) are successfully applied to your file.

The type properties are displayed on the file preview page, under **Properties**. You can edit these properties using **Edit Properties** under **Document Actions**.

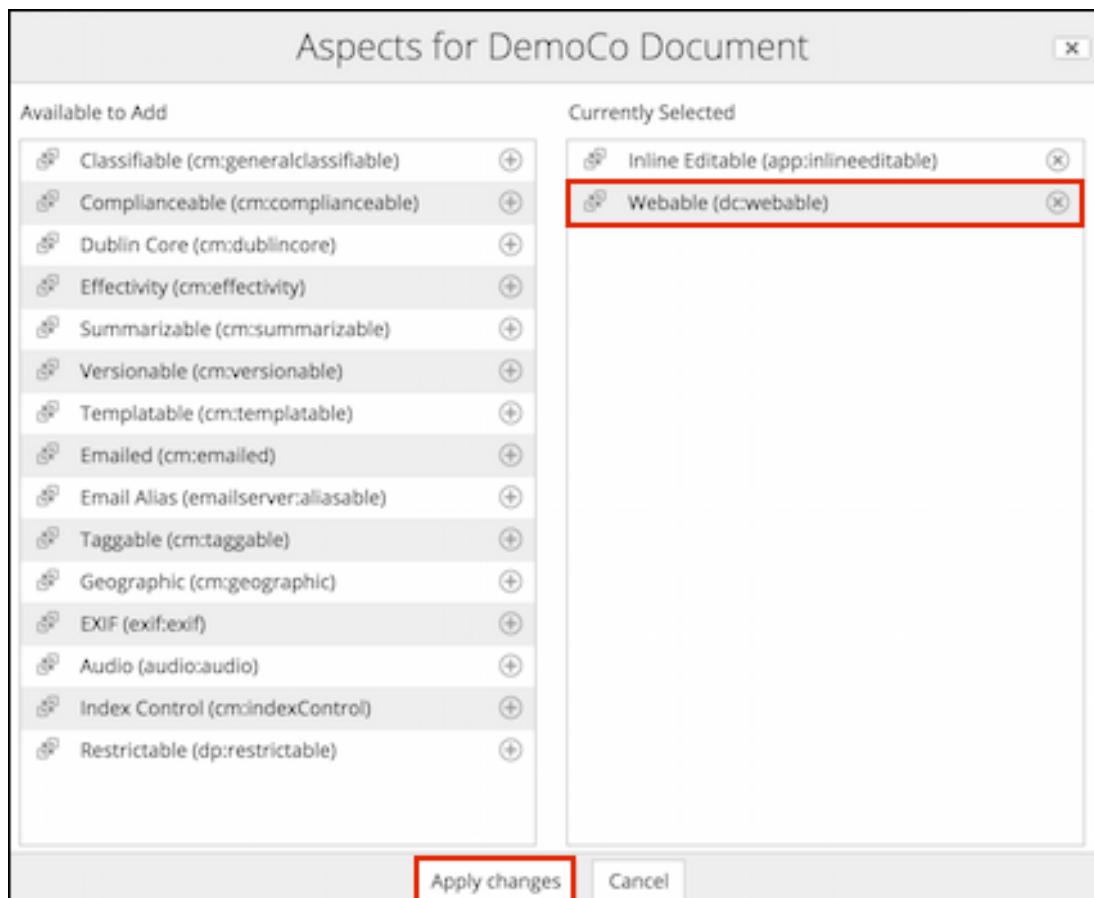
This screenshot compares two pages related to the document configuration. On the left is the 'File preview page', which shows the document content and a 'Properties' section at the bottom. On the right is the 'Edit Properties page', which is a detailed configuration form for the document. Both pages have red circles highlighting specific fields: 'Writer' and 'Domain' on the right, and a red box around the 'Properties' section on the left.

- d. In the **Writer** field, enter **User1**.
e. In the **domain** field, select **Sales**.
f. Click **Save**.

4. To apply the aspect, perform the following steps:

- Under **Document Actions**, click **Manage Aspects**.
- In the **Available to Add** list, click next to **Webable**.

Click to remove any existing aspects from the **Currently Selected** list.



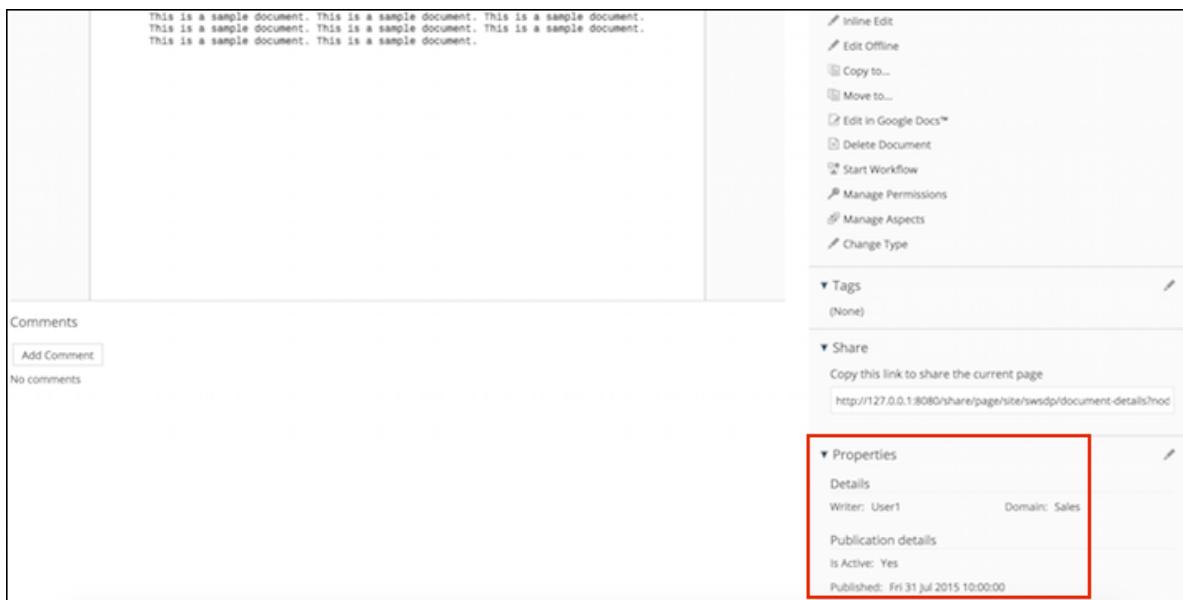
- Click **Apply changes**.

The aspect (*Webable*) and its properties (*Published* and *Is Active*) are successfully applied to the file.

The aspect properties are displayed on the file preview page, under **Properties**. You can edit these properties using **Edit Properties** under **Document Actions**.

- Select **Is Active**.
- In the **Published** field, enter the date and time as 31/07/2015 and 10:00, respectively.
- Click **Save**.

The updated information is displayed on the file preview page, under **Properties**.



In this tutorial, you learned how to create and apply models, custom types, and aspects using the Model Manager to capture metadata about files in Alfresco.

[back to top](#)

Model Manager video tutorials

Watch these videos to see how you can create and use models, custom types, and aspects with the Model Manager.

Create models with Model Manager

Learn how create models, custom types, and aspects using the Model Manager.

Use models

Learn how to use models in Alfresco.

Administering

There are a number of processes and procedures for maintaining and administering an Alfresco production environment.

Starting and stopping Alfresco

Use this information to understand 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 One Service > Start Alfresco One service**.
- 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 One > Alfresco One Service > Stop Alfresco One 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 One > Alfresco Share**.

Alfresco Share opens in a browser.

2. Log in using a user name and password.

Managing Share features

Use the Admin Tools to manage features of Alfresco Share such as look and feel, tagging, categories, and sites.

Changing the Share 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 Share 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**.

Using 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**.

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.

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.

Managing cloud syncs

With Cloud Sync Manager, Alfresco administrators have full visibility of all files and folders that have been synced from their organization to Alfresco Cloud sites, including their original and cloud locations, and the user who synced them.

Several filters are available so that you can search for specific synced items based on their ID or creator, or just view items where the sync has failed.

Syncs to cloud can fail for a number of reasons, such as the sync creator being removed from the system. If you have too many failed syncs on your network this can cause problems with all synced items. With Cloud Sync Manager you can identify any failed syncs and take ownership of them so that you can clear them out.

You can also **Export** sync details as a csv file.

-  Click the Original Item Location to view more details in the [Node Browser](#).

Becoming sync owner

With Cloud Sync Manager you can identify any failed syncs and take ownership of them so that you can clear them out.

You need to be administrator of both the Alfresco One site and the Alfresco Cloud network to become owner of a sync.

1. Click **Admin Tools**, and then **Cloud Sync Manager**.

All synced files and folders are displayed along with their creator, and the reason for any sync failure.

 You can use the filters to display specific synced items based on their ID or creator, or just view items where the sync has failed. Click **Apply Filter** to refresh the table to show only the syncs you're interested in.

2. Click **Actions** and select **Become Owner**. Click **OK** to confirm.

You are now the owner of this sync and can update or delete it as required.

Managing sites

The Sites Manager is used 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 users and groups

Use this information to administer your users and groups in Alfresco.

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 a 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 by 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 user. 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 | | daybar@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,malitos12345,,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 CSV 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 required.

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 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 a 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 a 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 required 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.

Working with Alfresco licenses

Access to Alfresco One is licensed on a per user basis.

You can register any number of users in Alfresco (see [Setting up Alfresco authentication and security](#)) and a license is only consumed when the registered user logs in to Alfresco and is authorized.

You can see a list of users along with their authorization states on the **Admin Console > Users and Groups > Users** page. A registered user can have any one of the following states:

- **Never Authorized**: Specifies that the user has been registered but never logged in to Alfresco.
- **Authorized**: Specifies that the user has successfully logged in to Alfresco.
- **Deauthorized**: Specifies that the Administrator has removed the user from the authorization list.

| Name | User Name | Job Title | Email | Usage | Quota | Authorization State | Deleted? | Actions |
|---------------|-----------|------------------|----------------------|---------|-------|---------------------|----------|---------|
| Alice Beecher | abeecher | Graphic Designer | abeecher@example.com | 8 MB | | Never Authorized | | |
| Administrator | admin | | admin@alfresco.com | 0 bytes | | Authorized | | |
| Guest | guest | | | 0 bytes | | Never Authorized | | |
| Mike Jackson | mjackson | Web Site Manager | mjackson@example.com | 8 MB | | Never Authorized | | |
| user1 | user1 | | user1@test.com | 0 bytes | | Deauthorized | | |
| user2 | user2 | | user2@test.com | 0 bytes | | Never Authorized | | |

If a user attempts to login for the first time and the user license limit has been exceeded, the login attempt will fail.

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.
-  In a clustered Alfresco environment, you should only apply the license to a single node. The license is shared by all the members of the cluster.

Authorize users

Registered users are authorized the first time they login to Alfresco unless they are Alfresco administrators, in which case they are pre-authorized.

When the user first logs in to Alfresco using the a user name and password, the login mechanism validates the login credentials. If the system is within the license limit, the user can successfully login to Alfresco, otherwise the login attempt fails and the user remains in the *Never Authorized* state. Periodically, the Authorization Audit Service job runs to check usage conforms to the license limit. The period at which the Authorization Audit Service job runs is set to default values in repository.properties as follows:

```
# Configuration of the Authorization Audit trigger

# By default every Second day of the week at 03:00

authorization.audit.day=2
authorization.audit.hour=3
authorization.audit.minute=0
```

This causes the job to run weekly. These properties can be overridden in alfresco-global.properties.

To authorize an administrative user, follow the steps below:

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. From the user groups list, click ALFRESCO_ADMINISTRATORS.
4. 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.
The individual user is added as a child to the ALFRESCO_ADMINISTRATORS group.
5. On the **Admin Tools** page, click **Users**.
The **User Search** page displays a list of all users along with their authorization status. The authorization state of the user added to the ALFRESCO_ADMINISTRATORS group has changed from *Never Authorized* to *Authorized*.

 To enable an Alfresco administrator to perform the administrative tasks, members of the ALFRESCO_ADMINISTRATORS permissions group are automatically authorised, if they are not already authorised.

Deauthorize an authorized user

You can remove authorized users from the authorization list by deauthorizing them.

Users from the ALFRESCO_ADMINISTRATORS permissions group cannot be deauthorized without being first removed from this group.

To deauthorize a user, follow the steps below:

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.
3. Click **Search**.
The result table displays the user name along with its authorization status.
4. Click the deauthorization icon () next to the user you want to deauthorize.

- You will be prompted to confirm your action.
5. In the prompt window, select the checkbox to confirm your action.
 6. Click **Deauthorize**.
- The relevant user's authorization status changes to Deauthorized.

To reauthorize a previously deauthorized user, contact [Alfresco Support](#).

Setting up clustering

You can 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 5.1

There are a number of prerequisites for upgrading from a version of Alfresco prior to Alfresco One 4.2 to Alfresco One 5.1 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 5.1 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 5.1. 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 5.1:

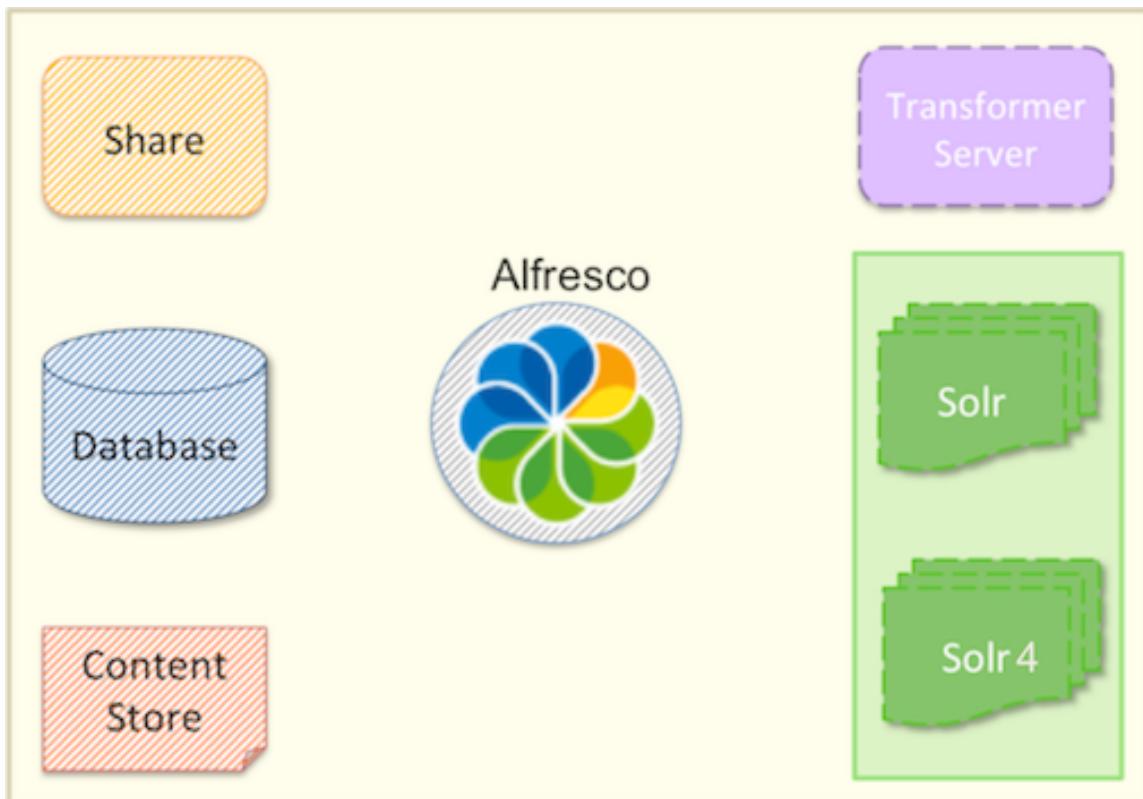
1. Browse to the `<classpathRoot>` directory.
For example, for Tomcat 7, browse to the `$TOMCAT_HOME/shared/classes/alfresco/extension/` directory.
2. Delete the `ehcache-custom.xml` file.
3. Browse to the `<classpathRoot>` directory.
For example, for Tomcat 7, browse to the `$TOMCAT_HOME/shared/classes/` directory.
4. Open the `alfresco-global.properties` file.
5. Remove the following legacy properties from the `alfresco-global.properties` file:
 - `alfresco.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 7, 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 One 5.1 cluster.

Components of an Alfresco solution

This information gives an overview of the main components of your Alfresco solution.

Within your Alfresco solution, you have components such as Share, Alfresco, database, indexes (Solr), transformations, and the content store. As illustrated in the following diagram, some of these components can be clustered while others are optional.



Clusterable components

- Share
- Alfresco
- Content store
- Database

Non-clusterable and replicable components

- Transformation Server
- Solr index
- Solr 4 index

It is important to decide how many Alfresco installations you need and which of the specified components you will put on which node.

To explain this further, let's consider an example. Assume that your application has six nodes. Ideally, Alfresco recommends that you should have one component on each node. So, for example, node1 has database, node2 is content store, node3 is Alfresco, node4 is Share, node5 is Solr, and node6 is Transformation Server.

Use the following table as a template to design your Alfresco solution.

| Clusters/
nodes | Alfresco
components | DNS | IP address (optional) | Additional information |
|--------------------|--------------------------|-----|-----------------------|------------------------|
| 1 | Database | | | |
| 2 | Content store | | | |
| 3 | Alfresco | | | |
| 4 | Share | | | |
| 5 | Solr | | | |
| 5 | Solr 4 | | | |
| 6 | Transformation
Server | | | |

However, you can configure your distribution differently. Each distribution and clustering solution has its own advantages and disadvantages. To configure distribution and clustering optimally, contact [Alfresco Consulting](#) or your Alfresco certified partner.

For a better understanding of initiating clustering for high availability and high throughput, see [Scenario: Clustering for redundancy](#) and [Scenario: Clustering for high throughput](#).

Setting up Share cluster

Use this information to configure a cluster for Share.

Configuring Hazelcast between Share instances

This information 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:

```

<?xml version='1.0' encoding='UTF-8'?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:hz="http://www.hazelcast.com/schema/spring"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
                           http://www.springframework.org/schema/beans/spring-
beans-2.5.xsd
                           http://www.hazelcast.com/schema/spring
                           http://www.hazelcast.com/schema/spring/hazelcast-
spring-2.4.xsd">

    <!--
        Hazelcast distributed messaging configuration - Share web-tier cluster
        config
        - see http://www.hazelcast.com/docs.jsp
        - and specifically http://docs.hazelcast.org/docs/2.4/manual/html-
single/#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.cluster.clusterservice"
    class="org.alfresco.web.site.ClusterTopicService" init-method="init">
        <property name="hazelcastInstance" ref="webframework.cluster.slingshot" />
        <property name="hazelcastTopicName"><value>slingshot-topic</value></property>
    </bean>
</beans>
```

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: /127.0.0.1]:5801 [slingshot] Hazelcast Community Edition 2.4 (20121017)
      starting at Address[127.0.0.1]:5801
Dec 13, 2014 12:09:36 PM com.hazelcast.system
INFO: /127.0.0.1]:5801 [slingshot] Copyright (C) 2008-2012 Hazelcast.com
Dec 13, 2014 12:09:36 PM com.hazelcast.impl.LifecycleServiceImpl
INFO: /127.0.0.1]:5801 [slingshot] Address[127.0.0.1]:5801 is STARTING
Dec 13, 2014 12:09:37 PM com.hazelcast.impl.TcpIpJoiner
INFO: /127.0.0.1]:5801 [slingshot] Connecting to possible member:
      Address[127.0.0.1]:5802
Dec 13, 2014 12:09:37 PM com.hazelcast.impl.TcpIpJoiner
INFO: /127.0.0.1]:5801 [slingshot] Connecting to possible member:
      Address[127.0.0.1]:5803
Dec 13, 2014 12:09:37 PM com.hazelcast.nio.SocketConnector
INFO: /127.0.0.1]:5801 [slingshot]

Members [1] {
  Member [127.0.0.1]:5801 this
}

Dec 13, 2014 12:09:38 PM com.hazelcast.impl.management.ManagementCenterService
INFO: /127.0.0.1]:5801 [slingshot] Hazelcast Management Center started at port
      5901.
Dec 13, 2014 12:09:38 com.hazelcast.impl.LifecycleServiceImpl
INFO: /127.0.0.1]:5801 [slingshot] Address[127.0.0.1]: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

There are a number of recommendations 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

There are a number of 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

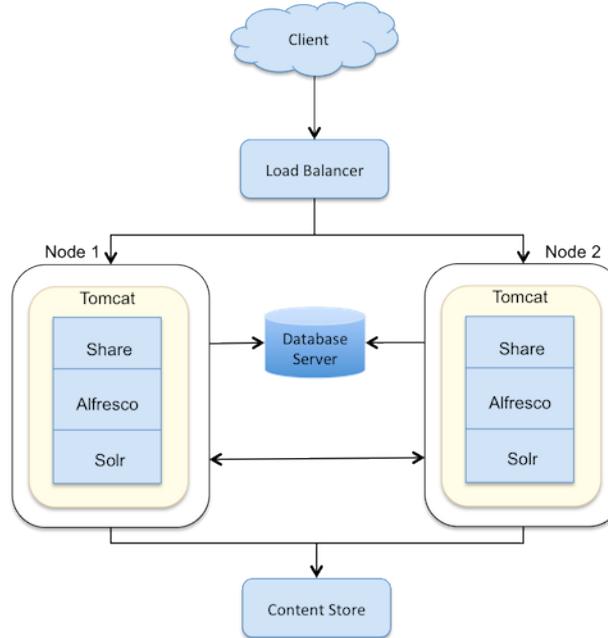
When you have decided to upgrade from a single node environment to a distributed or clustered environment, you must find the most appropriate way to cluster Alfresco's architecture.

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.



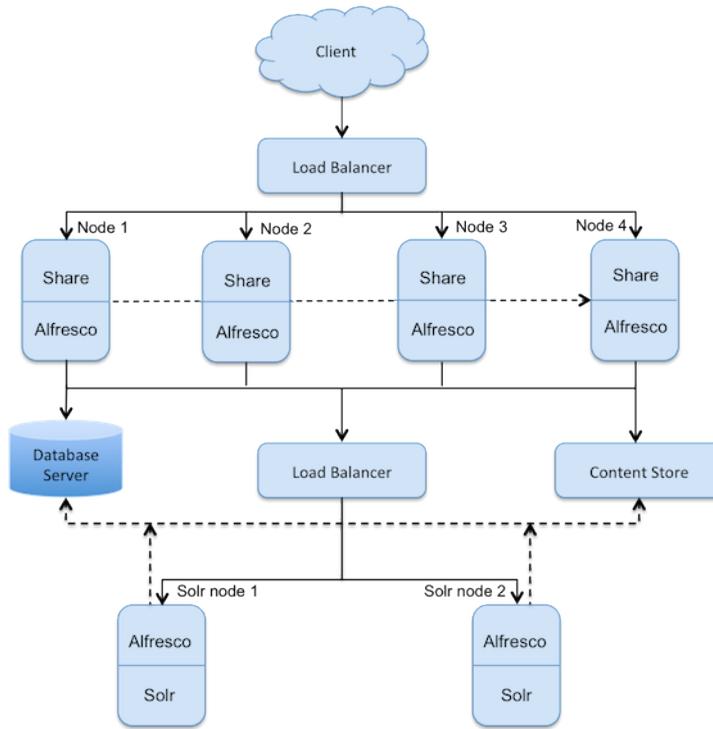
A hardware load balancer balances the web requests among multiple servers. The load balancer must support 'sticky' sessions so that each client always connects to the same server during the session. The content store and database will reside on separate servers, which allows us to use alternative means 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

Use this information to implement an Alfresco One repository server cluster.

The repository server cluster consists of the following components:

- Database server
- Content store
- Solr server
- Load balancer
- Hazelcast mancenter server (optional)

Setting up repository server cluster

Follow these steps to set 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.
 - 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

Use this information to start 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.

Managing members of a cluster

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

 Ensure that clustering is enabled for your license.

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	Enabled or Disabled	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	xxxxxx	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**.

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.

Clustering property	Example setting	What is it?
<code>alfresco.cluster.nodetype</code>	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 is attached to the same database as the cluster, but not participating in it (for example, <code>alfresco.cluster.enabled=false</code>).
<code>alfresco.hazelcast.password</code>	<code>alfrescocluster</code>	This specifies the password used by the cluster members to access or join the Hazelcast cluster.
<code>alfresco.hazelcast.port</code>	5701	This specifies the port to use for clustering.
<code>alfresco.hazelcast.autoinc.port</code>	<code>false</code>	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.
<code>alfresco.hazelcast.mancenter.enabled</code>	<code>false</code>	This enables the server to push stats and other useful information to Hazelcast's mancenter (management center) dashboard application. See Setting up Hazelcast dashboard .
<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.

The Hazelcast diagnostics and reporting application is a useful addition to an Alfresco repository cluster. It can be installed on any servlet container.

-  The Hazelcast 2.4.1 Management Center does not work with Java 8. Alfresco recommends that you use Hazelcast Management Center with Java 7.
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.

CIFS clustering through load balancer

Follow these steps to configure CIFS clustering through the load balancer.

Setting up a CIFS cluster involves configuring the Balance application and the HAProxy load balancer.

Balance is a load balancing solution for simple TCP proxy with round robin load balancing and fail over mechanisms.

1. Configure Balance.

For Linux from Source:

- a. Download Balance from the [Balance download page](#).

Ensure that building toolchains specific to your OS version are installed. For example, GNU Compiler Collection (GCC), GNU make, or any other related packages.

- b. Run the following commands to install Balance:

```
make
make install
```

This installs Balance at /usr/sbin/ and the man page at /usr/man/man1.

- c. Enable Balance to bind on port 445 of the local IPv4 IP address and distribute connections to <host1_IP>, port 445, and <host2_IP>, port 445.

```
#balance -fb ::ffff:<IP> 445 <host1_IP>:445 % <host2_IP>:445
```

where <IP> is the local IPv4 IP address, <host1_IP> is the IPv4 address of the first server hostname, and <host2_IP> is the IPv4 address of the second server hostname.

2. Configure HAProxy.

- a. To configure HAProxy on Solaris 11.2, [download the appropriate version of HAProxy](#) in accordance to your server (x86 or Sparc).

For example, haproxy-1.4.18-pcre-solaris10-x86.stripped.gz.

- b. Get Perl Compatible Regular Expressions (PCRE) and its dependencies from <http://www.pcre.org/>.

1. Unzip the PCRE library.

```
gunzip pcre-x.x.tar.gz
tar xf pcre-x.x.tar
cd pcre-x.x
```

2. Run the following commands:

```
./configure --enable-static --enable-shared --prefix=/usr/
local--enable-unicode-properties
make && make install
```

- c. Create a new user and group with name haproxy.

- d. Run the following commands:

```
gunzip haproxy-1.4.x-pcre-solaris10-x86.stripped.gz
mv haproxy-1.4.x-pcre-solaris10-x86.stripped haproxy
```

```
mv haproxy /usr/bin/
chmod +x /usr/bin/haproxy
mkdir -p /etc/haproxy
```

- e. Create and edit the `/etc/haproxy/haproxy.cfg` configuration file by adding the configuration shown below:

```
global
    log 127.0.0.1  local0 notice
    user haproxy
    group haproxy
    chroot /etc/haproxy #directory
    daemon
    nbproc 7
    pidfile /var/run/haproxy.pid

defaults
    log global
    option tcplog
    option redispatch
    contimeout      3000
    clitimeout     5000
    srvtimeout     5000

listen hostname <IP>:445
    mode tcp
    balance roundrobin
    server hostname <host1_IP>:445 weight 77
    server hostname <host2_IP>:445 weight 179
```

 Make sure you have `/usr/bin` in your environment path.

- f. Run HAProxy with the following command:

```
haproxy -f /etc/haproxy/haproxy.cfg -D
```

Avoiding cluster clash

Use this information to understand cluster cloning clash and how to avoid it.

When cloning a clustered environment (including the database), the cloned nodes will try to join the original cluster because they will point to the same repository reference. The repository ID is copied along with all the other information in the database and it is not possible to change or delete it.

This causes cluster clash because the nodes think that they are in the same cluster and share information, although they are actually pointing to different repositories and databases.

How to avoid this?

The most commonly used ways to avoid cluster clash are:

- Change the Hazelcast port in the `alfresco-global.properties` file using `alfresco.hazelcast.port` (the default is 5701).
- Change the Hazelcast password in the `alfresco-global.properties` file using `alfresco.hazelcast.password` (the default is `alfrescocluster`).
- It is also possible to provide isolation between the instances at the network level. For more information, contact your network administrator.

Tracking clustering issues

Use this information to track clustering issues in Alfresco.

- 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 5.1 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
```

Setting up 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.
-  If you have pre-existing user logins with syntax <name>@<domain>, you should not create a tenant with that domain name. This will break the login functionality of the existing users with logins <name>@<domain>.

However, if you wish to disable multi-tenancy, you need to delete all the tenants. See [Managing tenants](#) on page 480 for more information.

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 13tm31n /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.

-  If you have pre-existing user logins with syntax <name>@<domain>, you should not create a tenant with that domain name. This will break the login functionality of the existing users with logins <name>@<domain>.

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
- Smart Folders
- CIFS
- Portlets
- LDAP, NTLM and authentication methods other than alfresco
- Inbound email
- Content replication
- IMAP
- Encrypted Content Store

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.

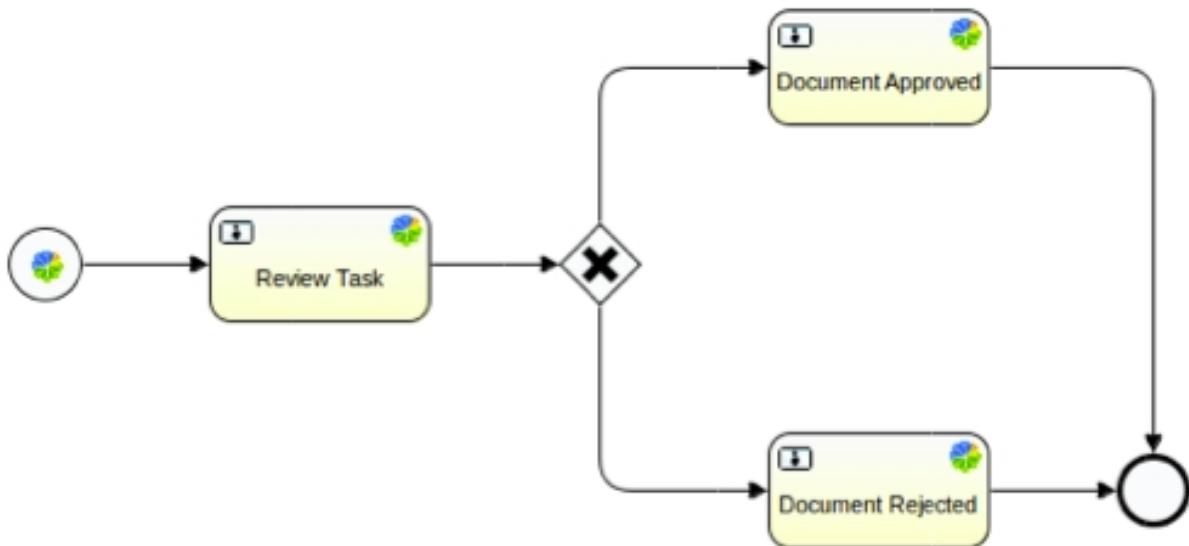
Pooled Review & Approve

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>
    ....

```

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:

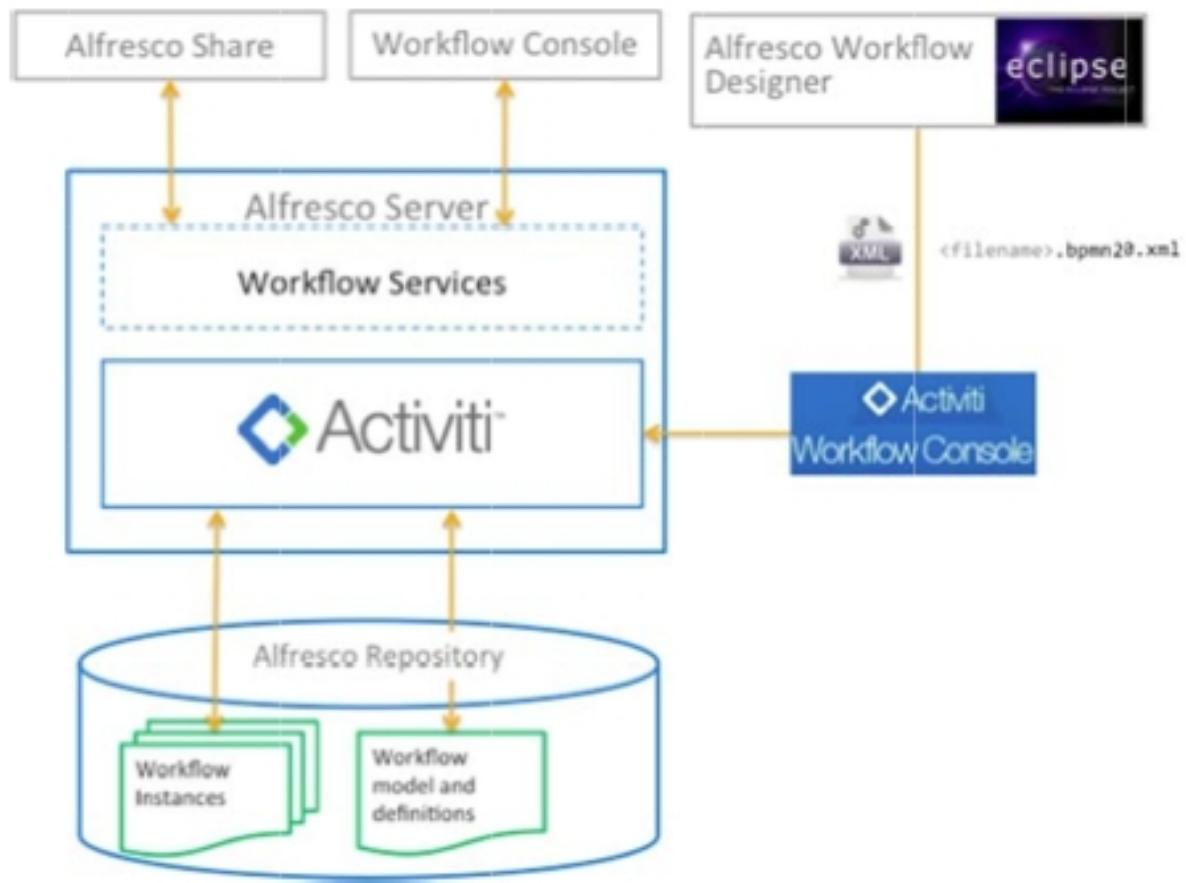
- An Alfresco license that includes access to Hybrid workflow
- Access to an appropriate Cloud network

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.

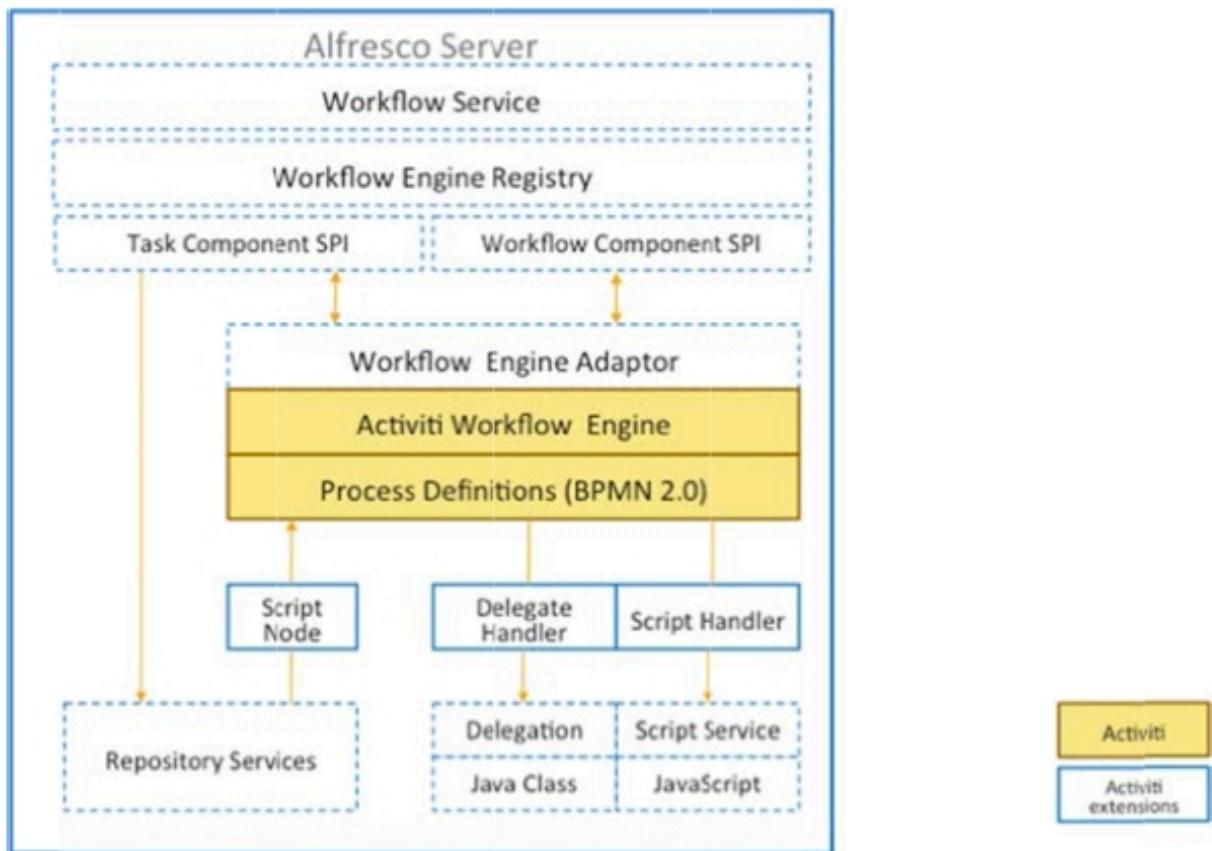


You can design workflow definitions using a graphical workflow designer that supports BPMN 2.0 or write the XML BPMN 2.0 process definition directly using an XML editor. Many workflow editors support BPMN 2.0 but might not understand some of the features of Alfresco workflow. We recommend the use of 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 :-



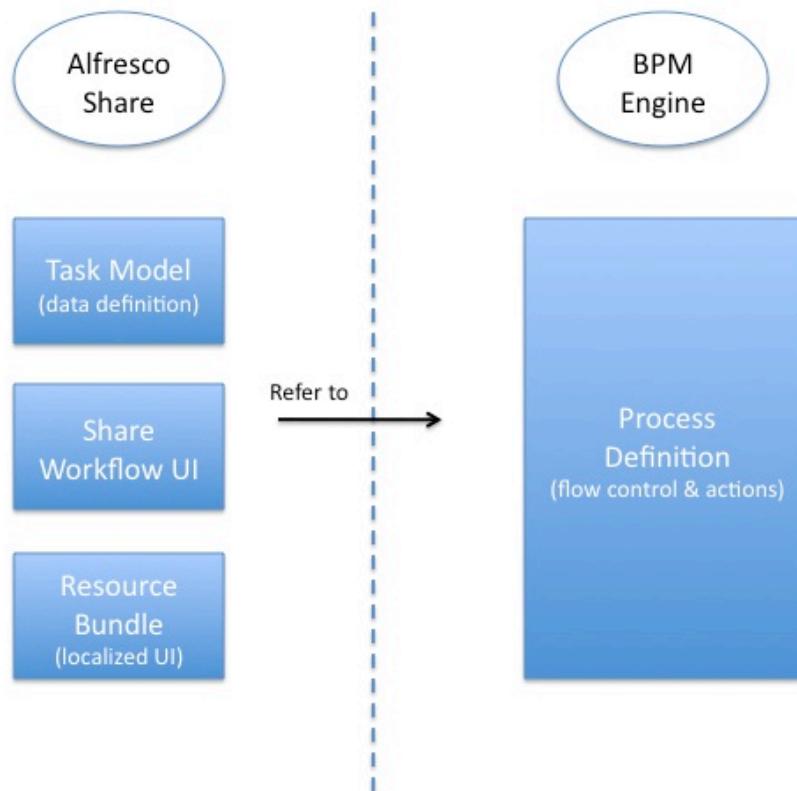
Workflow instances

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:

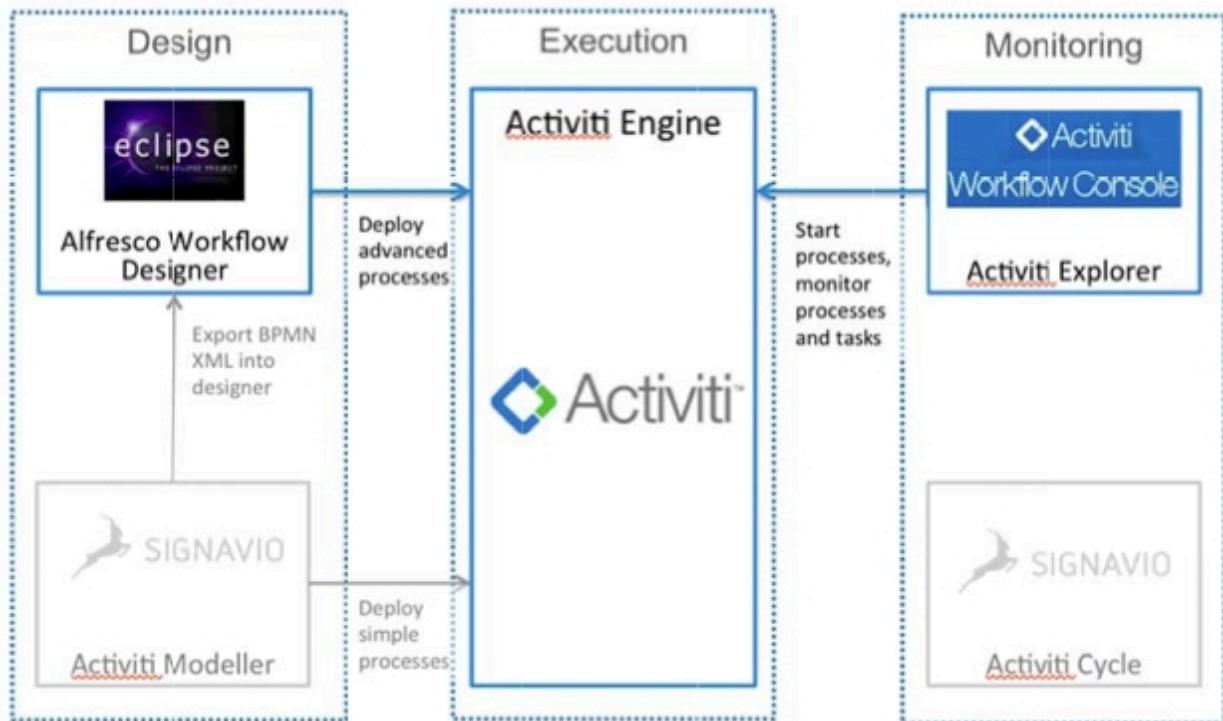
- Control over which Task properties are displayed
- Control over which Task properties are read-only and required
- Control over how each Task property is rendered in the forms

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 Workflow Console

Use the **Workflow Console** in the Admin Console to manage Activiti workflows, including testing of newly developed workflows. You can also debug current in-flight workflows.

- 💡 The Workflow Console must not be used to cancel in-flight 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 (activiti) and <definitionClassPath> is the class path to workflow definition.
 - c. To redeploy the last workflow definition, type `redeploy`.
 - 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.
 - 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).

Enabling workflow process engines

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	Displays the state of 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 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. 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](#).

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

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

The Activiti workflow console

With the **Activiti Workflow Console** you can:

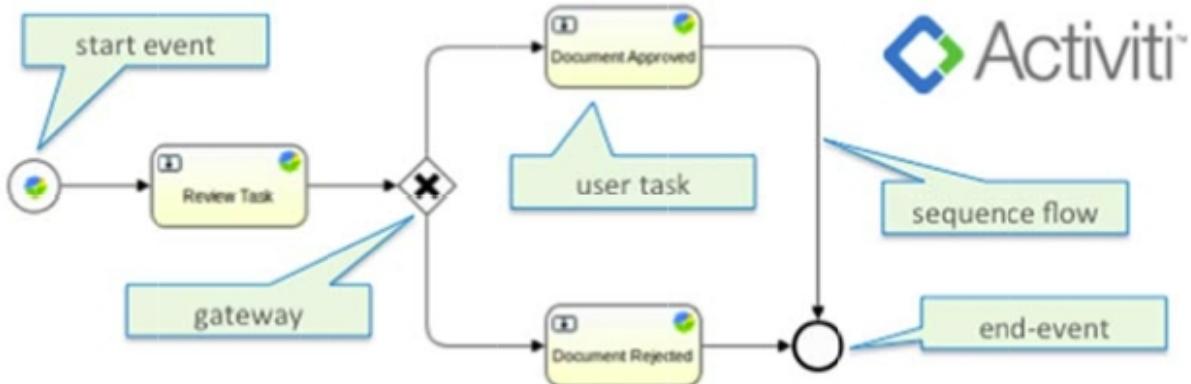
- View process definitions
- Manage deployments; deploy, view versions, and delete versions
- Manage process instances
- View task variables
- Examine the process database

To start the Activiti workflow console:

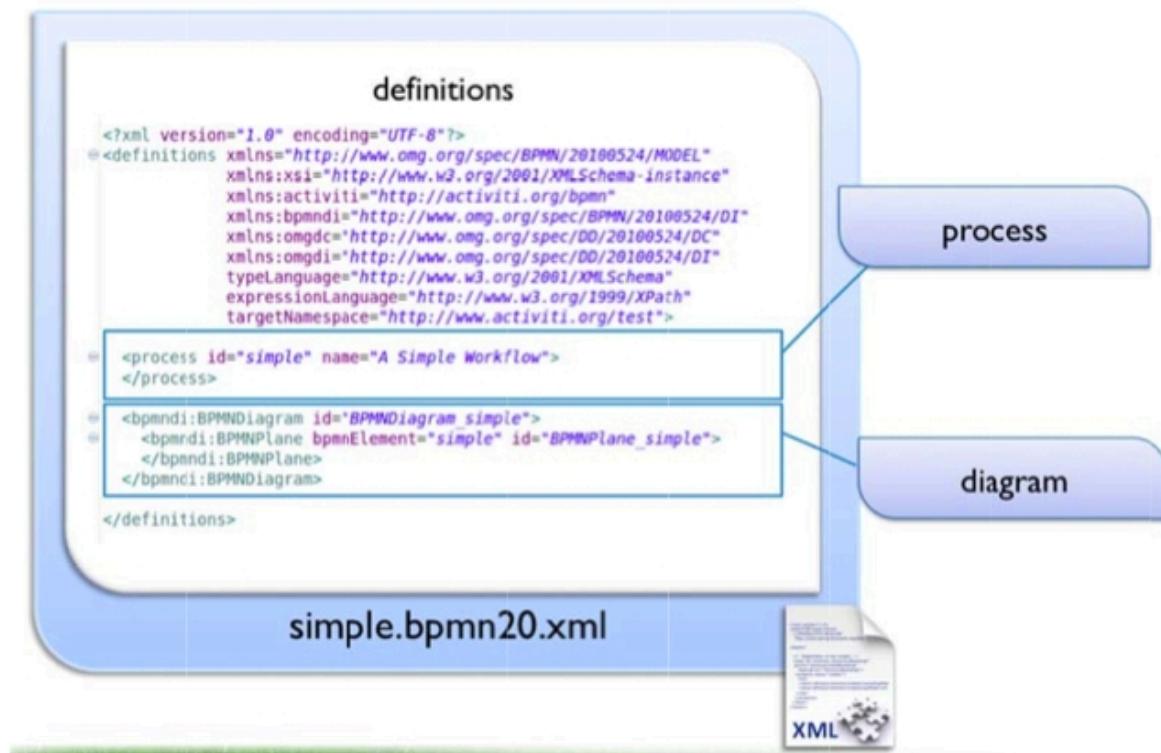
1. Launch the [Admin Console](#)
2. In the **Repository Services** section, click **Process Engines**
3. In the bottom right panel, click **Activiti Workflow Console**

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.

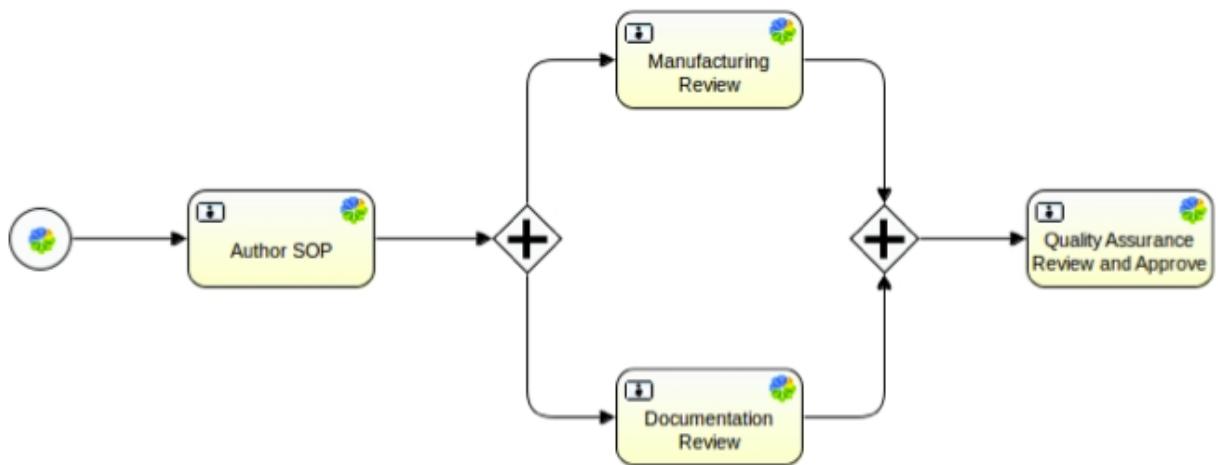
join

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.

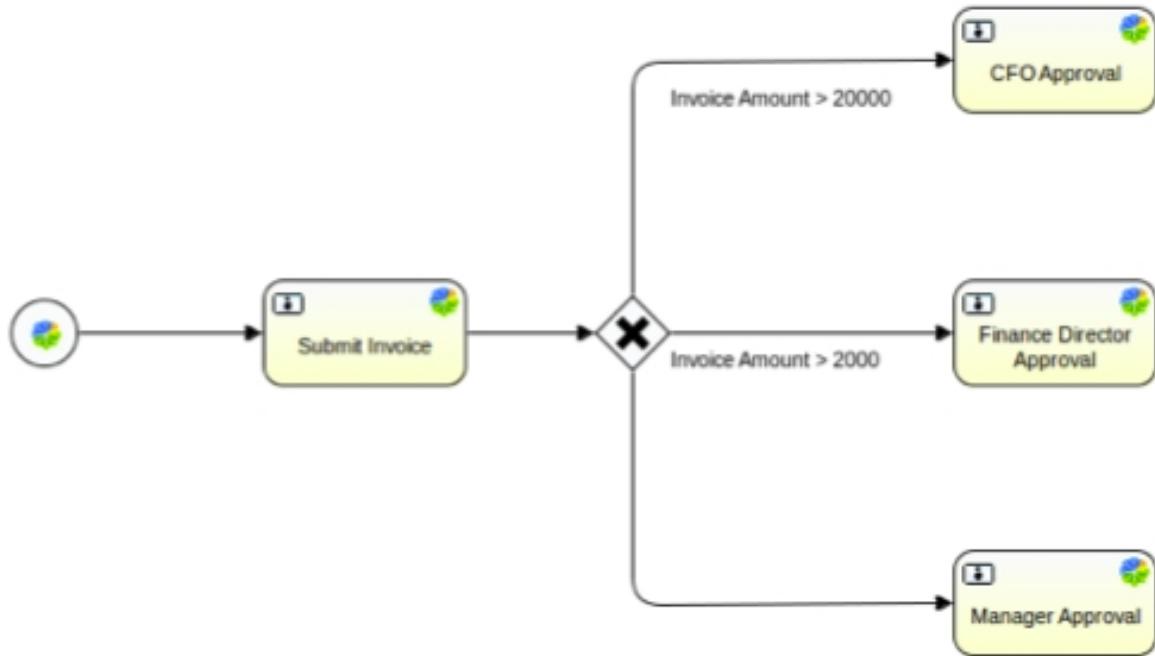
Exclusive gateways

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.

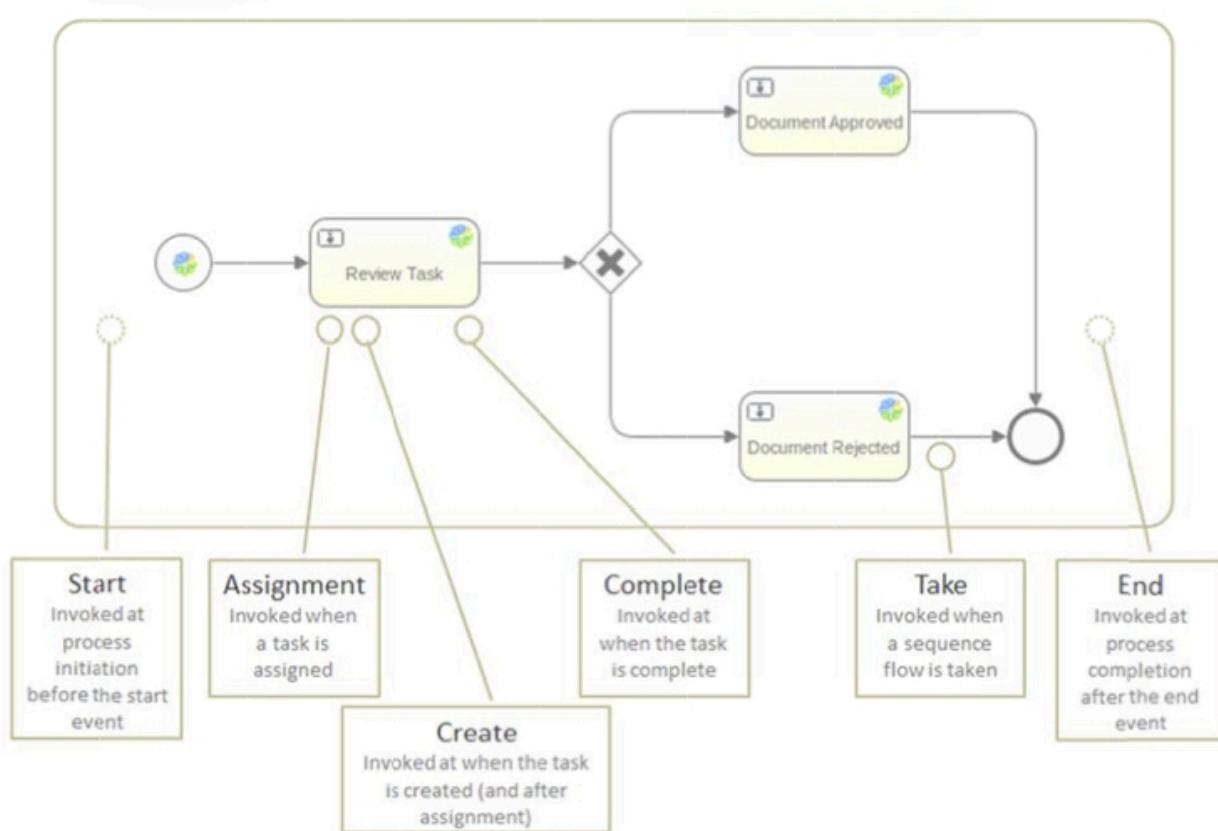
companyhome

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: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.

```

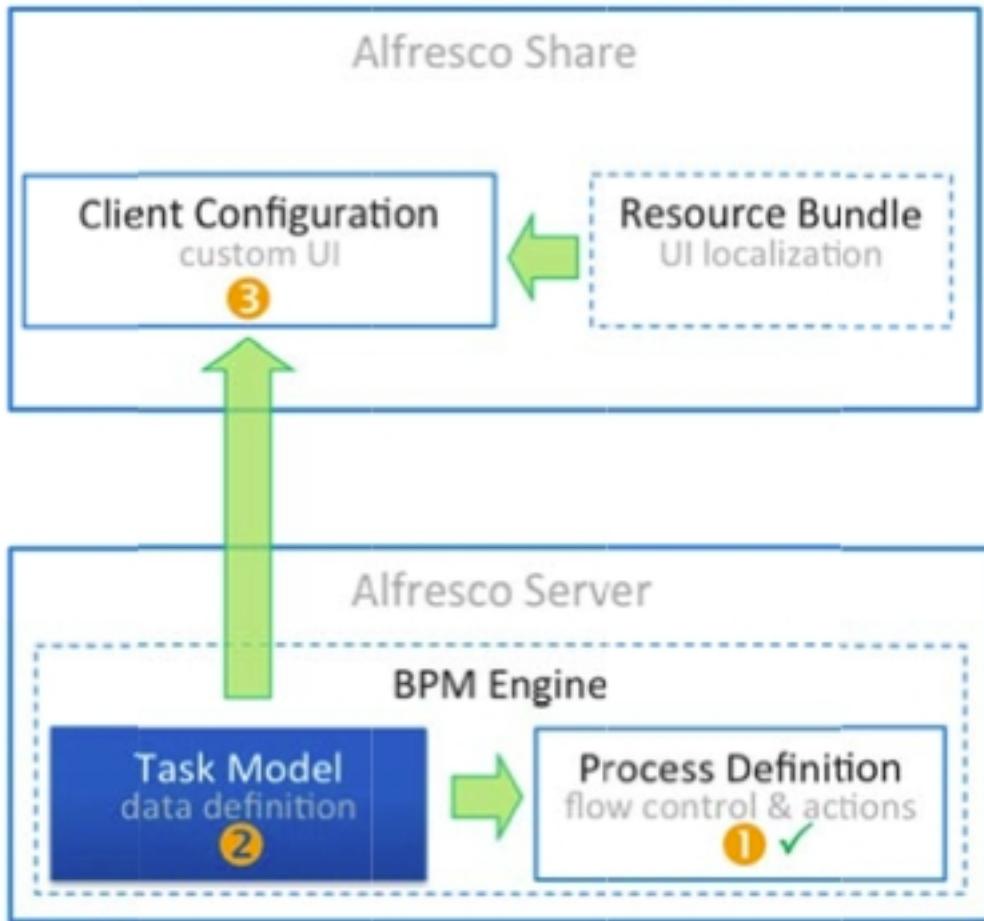
<process id="StandardGroupReview" name="Parallel Group Review And Approve Activiti Process">
  <extensionElements>
    <activiti:executionListener
      event="start"
      class="org.alfresco.repo.workflow.activiti.listener.ScriptExecutionListener">
      <activiti:field name="script">
        <activiti:string>
          execution.setVariable('wf_approveCount', 0);
          execution.setVariable('wf_actualPercent', 0);
          execution.setVariable('wf_requiredPercent', wf_requiredApprovePercent);</activiti:string>
        </activiti: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.

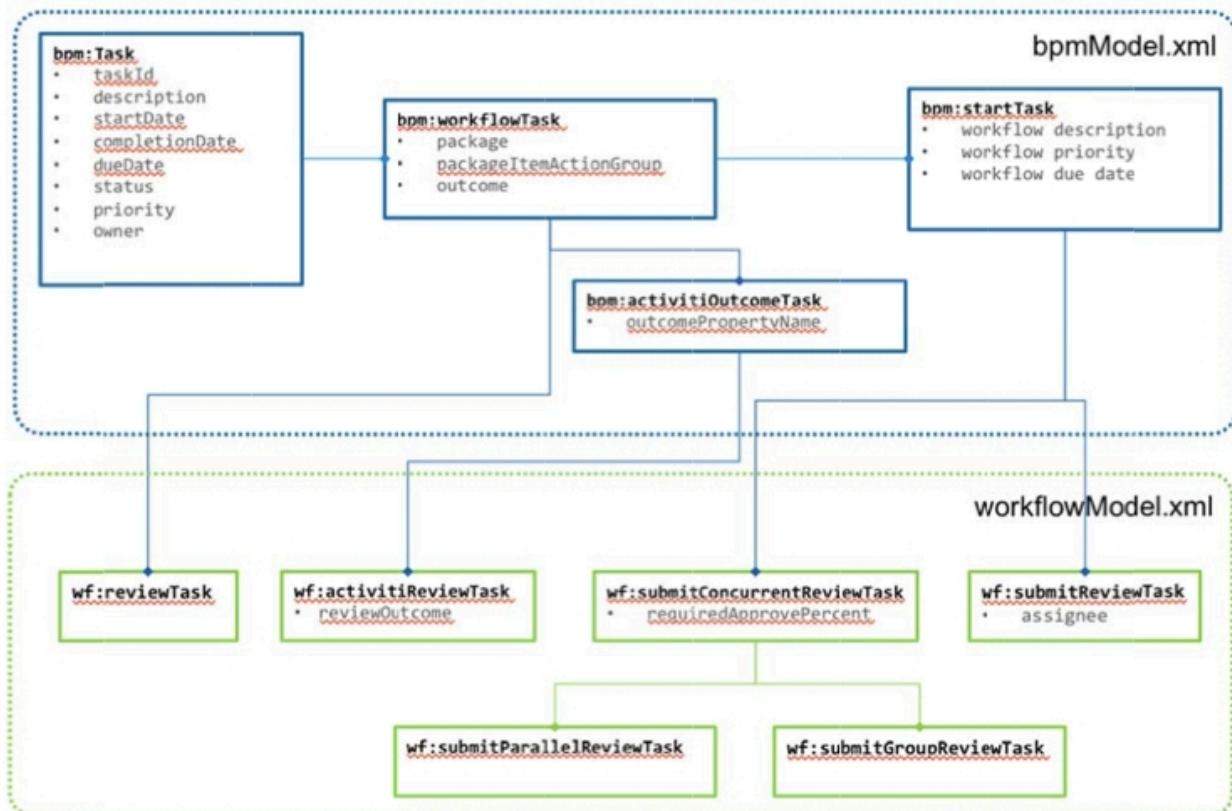
Alfresco ships with two default workflow models that support the default set of process definitions.

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.

The screenshot shows the Activiti Task Management interface. On the left, there is a table titled 'Variables' listing various BPMN variables and their values. One variable, 'NAME', has a value 'Review Task' which is highlighted with a green border. A green arrow points from this highlighted value to the main task editor window on the right. The main window is titled 'Edit Task: Review Task'. It contains sections for 'Info', 'Progress', 'Items', and 'Response'. In the 'Info' section, the message is 'Please review this before I publish', owner is 'Alice Beecher', priority is 'High', and due date is 'Tue 14 Feb 2012'. The 'Progress' section shows status as 'Not Yet Started'. The 'Items' section displays an attachment named 'low consumption bulb.png' with a small thumbnail, a description '(None)', and a modification date of 'Thu 3 Mar 2011 04:34:53'. The 'Response' section has a comment input field and two buttons: 'Approve' and 'Reject'.

NAME	PRIORITY	ASSIGNEE	DUEDATE	CREATED
Review Task	1	abeecher	Not finished yet	moments ago

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.

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.

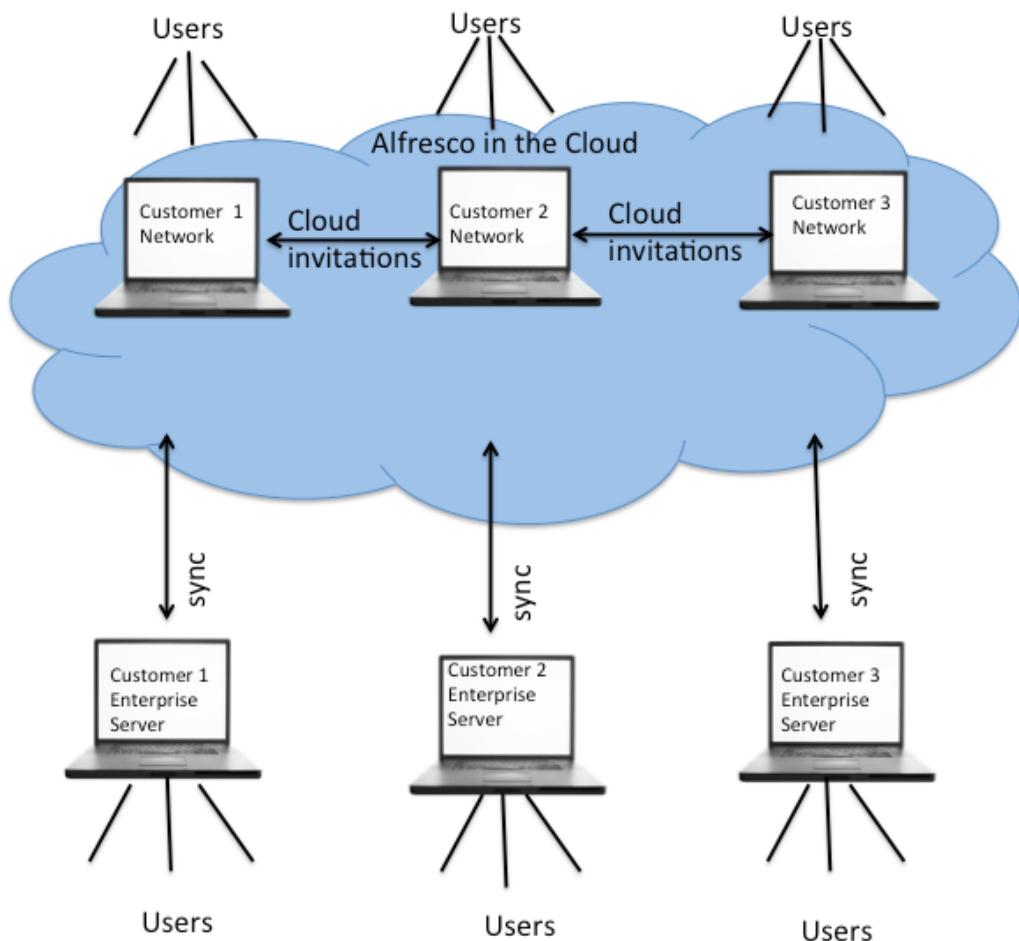
Enterprise to Cloud Sync overview

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

Use this information 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. Both `audit.enabled=true` and `audit.sync.enabled=true` are required in the application configuration to enable auditing.

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 for the Enterprise to Cloud Sync feature.
4. Restart the Alfresco server for the Cloud Sync actions to be visible in Alfresco Share.
5. 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

Use these error messages and solutions when troubleshooting unexpected behavior with Enterprise to Cloud Sync.

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, or you have exceeded your Cloud storage quota.	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.
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.	Make sure that Alfresco Cloud is up and running.
			Rename the on-premise node and try syncing again or delete the node with the same name from Alfresco Cloud and try synchronizing again.

End user error message	Description	Possible causes	Solutions
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.
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	Unsyncronize the content.

End user error message	Description	Possible causes	Solutions
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 alfresco-global.properties file.	Set a valid URL for Alfresco Cloud in the alfresco-global.properties file and run on-premise Alfresco again.

Enterprise to Cloud Sync FAQs

Frequently asked questions 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 Alfresco Outlook Integration, or Transformation Server, or Alfresco Media Management, there are additional transform options, which are listed in [Additional transform options](#) on page 567.

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	No	Yes
application/msword	No	Yes
application/pdf	No	Yes
application/rtf	No	Yes
application/vnd.apple.keynote	No	Yes
application/vnd.apple.numbers	No	Yes
application/vnd.apple.pages	No	Yes
application/vnd.ms-excel	No	Yes
application/vnd.ms-excel.addin.macroenabled.12	No	Yes
application/vnd.ms-excel.sheet.binary.macroenabled.12	No	Yes
application/vnd.ms-excel.sheet.macroenabled.12	No	Yes
application/vnd.ms-excel.template.macroenabled.12	No	Yes
application/vnd.ms-outlook	No	Yes
application/vnd.ms-powerpoint	No	Yes
application/vnd.ms-powerpoint.addin.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.presentation.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.slide.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.slideshow.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.template.macroenabled.12	No	Yes
application/vnd.ms-word.document.macroenabled.12	No	Yes
application/vnd.ms-word.template.macroenabled.12	No	Yes
application/vnd.oasis.opendocument.graphics	No	Yes
application/vnd.oasis.opendocument.presentation	No	Yes
application/vnd.oasis.opendocument.presentation-template	No	Yes
application/vnd.oasis.opendocument.spreadsheet	No	Yes
application/vnd.oasis.opendocument.spreadsheet-template	No	Yes
application/vnd.oasis.opendocument.text	No	Yes
application/vnd.oasis.opendocument.text-template	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.presentation	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.slide	No	Yes

Format	Transformable to:	Transformable from:
application/vnd.openxmlformats-officedocument.presentationml.slideshow	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.template	No	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	No	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.template	No	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.document	No	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.template	No	Yes
application/vnd.sun.xml.calc	No	Yes
application/vnd.sun.xml.calc.template	No	Yes
application/vnd.sun.xml.impress	No	Yes
application/vnd.sun.xml.impress.template	No	Yes
application/vnd.sun.xml.writer	No	Yes
application/vnd.sun.xml.writer.template	No	Yes
application/vnd.visio	No	Yes
application/wordperfect	No	Yes
image/bmp	Yes	Yes
image/cgm	Yes	Yes
image/gif	Yes	Yes
image/ief	Yes	Yes
image/jp2	Yes	Yes
image/jpeg	Yes	Yes
image/png	Yes	Yes
image/tiff	Yes	Yes
image/vnd.adobe.photoshop	Yes	Yes
image/vnd.adobe.premiere	Yes	Yes
image/x-cmu-raster	Yes	Yes
image/x-dwt	Yes	Yes
image/x-portable-anymap	Yes	Yes
image/x-portable-bitmap	Yes	Yes
image/x-portable-graymap	Yes	Yes
image/x-portable-pixmap	Yes	Yes
image/x-raw-adobe	Yes	Yes
image/x-raw-canon	Yes	Yes
image/x-raw-fuji	Yes	Yes
image/x-raw-hasselblad	Yes	Yes
image/x-raw-kodak	Yes	Yes
image/x-raw-leica	Yes	Yes
image/x-raw-minolta	Yes	Yes
image/x-raw-nikon	Yes	Yes
image/x-raw-olympus	Yes	Yes
image/x-raw-panasonic	Yes	Yes
image/x-raw-pentax	Yes	Yes

Format	Transformable to:	Transformable from:
image/x-raw-red	Yes	Yes
image/x-raw-sigma	Yes	Yes
image/x-raw-sony	Yes	Yes
image/x-xbitmap	Yes	Yes
image/x-xpixmap	Yes	Yes
image/x-xwindowdump	Yes	Yes
text/csv	No	Yes
text/html	No	Yes
text/plain	No	Yes
text/xml	No	Yes

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	Yes
image/bmp	Yes
image/cgm	Yes
image/gif	Yes
image/ief	Yes
image/jp2	Yes
image/jpeg	Yes
image/png	Yes
image/tiff	Yes
image/vnd.adobe.photoshop	Yes
image/vnd.adobe.premiere	Yes
image/x-cmu-raster	Yes
image/x-dwt	Yes
image/x-portable-anymap	Yes
image/x-portable-bitmap	Yes
image/x-portable-graymap	Yes
image/x-portable-pixmap	Yes
image/x-raw-adobe	Yes
image/x-raw-canon	Yes
image/x-raw-fuji	Yes
image/x-raw-hasselblad	Yes
image/x-raw-kodak	Yes
image/x-raw-leica	Yes
image/x-raw-minolta	Yes
image/x-raw-nikon	Yes
image/x-raw-olympus	Yes
image/x-raw-panasonic	Yes
image/x-raw-pentax	Yes
image/x-raw-red	Yes

Format	Transformable to:
image/x-raw-sigma	Yes
image/x-raw-sony	Yes
image/x-bitmap	Yes
image/x-xpixmap	Yes
image/x-xwindowdump	Yes

application/java, application/json and application/mac-binhex40 - class, json, hqx

These formats cannot be transformed into, or generated from, any other format.

application/java-archive - jar

 This format cannot be generated from any other format.

Format	Transformable to:
application/xhtml+xml	Yes
text/html	Yes
text/plain	Yes
text/xml	Yes

application/msword - doc

Format	Transformable to:	Transformable from:
application/eps	Yes	No
application/pdf	Yes	No
application/rtf	Yes	Yes
application/vnd.ms-outlook	No	Yes
application/vnd.ms-word.document.macroenabled.12	No	Yes
application/vnd.ms-word.template.macroenabled.12	No	Yes
application/vnd.oasis.opendocument.text	Yes	Yes
application/vnd.oasis.opendocument.text-template	Yes	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.document	No	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.template	No	Yes
application/vnd.sun.xml.writer	Yes	Yes
application/vnd.sun.xml.writer.template	Yes	Yes
application/wordperfect	No	Yes
application/xhtml+xml	Yes	No
image/bmp	Yes	No
image/cgm	Yes	No
image/gif	Yes	No
image/ief	Yes	No
image/jp2	Yes	No
image/jpeg	Yes	No
image/png	Yes	No
image/tiff	Yes	No
image/vnd.adobe.photoshop	Yes	No
image/vnd.adobe.premiere	Yes	No
image/x-cmu-raster	Yes	No
image/x-dwt	Yes	No

Format	Transformable to:	Transformable from:
image/x-portable-anymap	Yes	No
image/x-portable-bitmap	Yes	No
image/x-portable-graymap	Yes	No
image/x-portable-pixmap	Yes	No
image/x-raw-adobe	Yes	No
image/x-raw-canon	Yes	No
image/x-raw-fuji	Yes	No
image/x-raw-hasselblad	Yes	No
image/x-raw-kodak	Yes	No
image/x-raw-leica	Yes	No
image/x-raw-minolta	Yes	No
image/x-raw-nikon	Yes	No
image/x-raw-olympus	Yes	No
image/x-raw-panasonic	Yes	No
image/x-raw-pentax	Yes	No
image/x-raw-red	Yes	No
image/x-raw-sigma	Yes	No
image/x-raw-sony	Yes	No
image/x-xbitmap	Yes	No
image/x-x pixmap	Yes	No
image/x-xwindowdump	Yes	No
text/html	Yes	Yes
text/plain	Yes	Yes
text/xml	Yes	No

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	Yes
text/html	Yes
text/plain	Yes
text/xml	Yes

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	Yes	No
application/msword	No	Yes
application/rtf	No	Yes
application/vnd.apple.keynote	No	Yes
application/vnd.apple.numbers	No	Yes

Format	Transformable to:	Transformable from:
application/vnd.apple.pages	No	Yes
application/vnd.ms-excel	No	Yes
application/vnd.ms-excel.sheet.binary.macroenabled.12	No	Yes
application/vnd.ms-excel.sheet.macroenabled.12	No	Yes
application/vnd.ms-excel.template.macroenabled.12	No	Yes
application/vnd.ms-outlook	No	Yes
application/vnd.ms-powerpoint	No	Yes
application/vnd.ms-powerpoint.addin.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.presentation.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.slide.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.template.macroenabled.12	No	Yes
application/vnd.ms-word.document.macroenabled.12	No	Yes
application/vnd.ms-word.template.macroenabled.12	No	Yes
application/vnd.oasis.opendocument.graphics	No	Yes
application/vnd.oasis.opendocument.presentation	No	Yes
application/vnd.oasis.opendocument.presentation-template	No	Yes
application/vnd.oasis.opendocument.spreadsheet	No	Yes
application/vnd.oasis.opendocument.spreadsheet-template	No	Yes
application/vnd.oasis.opendocument.text	No	Yes
application/vnd.oasis.opendocument.text-template	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.presentation	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.slide	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.template	No	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	No	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.template	No	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.document	No	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.template	No	Yes
application/vnd.sun.xml.calc	No	Yes
application/vnd.sun.xml.calc.template	No	Yes
application/vnd.sun.xml.impress	No	Yes
application/vnd.sun.xml.impress.template	No	Yes
application/vnd.sun.xml.writer	No	Yes
application/vnd.sun.xml.writer.template	No	Yes
application/vnd.visio	No	Yes
application/wordperfect	No	Yes
application/xhtml+xml	Yes	No
image/bmp	Yes	No
image/cgm	Yes	No
image/gif	Yes	No
image/ief	Yes	No

Format	Transformable to:	Transformable from:
image/jp2	Yes	No
image/jpeg	Yes	No
image/png	Yes	No
image/tiff	Yes	Yes
image/vnd.adobe.photoshop	Yes	No
image/vnd.adobe.premiere	Yes	No
image/x-cmu-raster	Yes	No
image/x-dwt	Yes	No
image/x-portable-anymap	Yes	No
image/x-portable-bitmap	Yes	No
image/x-portable-graymap	Yes	No
image/x-portable-pixmap	Yes	No
image/x-raw-adobe	Yes	No
image/x-raw-canon	Yes	No
image/x-raw-fuji	Yes	No
image/x-raw-hasselblad	Yes	No
image/x-raw-kodak	Yes	No
image/x-raw-leica	Yes	No
image/x-raw-minolta	Yes	No
image/x-raw-nikon	Yes	No
image/x-raw-olympus	Yes	No
image/x-raw-panasonic	Yes	No
image/x-raw-pentax	Yes	No
image/x-raw-red	Yes	No
image/x-raw-sigma	Yes	No
image/x-raw-sony	Yes	No
image/x-xbitmap	Yes	No
image/x-xpixmap	Yes	No
image/x-xwindowdump	Yes	No
text/csv	No	Yes
text/html	Yes	Yes
text/plain	Yes	Yes
text/xml	Yes	Yes

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	Yes
text/html	Yes
text/plain	Yes
text/xml	Yes

application/rtf - rtf

Format	Transformable to:	Transformable from:
application/eps	Yes	No
application/msword	Yes	Yes
application/pdf	Yes	No
application/vnd.ms-outlook	No	Yes
application/vnd.ms-word.document.macroenabled.12	No	Yes
application/vnd.ms-word.template.macroenabled.12	No	Yes
application/vnd.oasis.opendocument.text	Yes	Yes
application/vnd.oasis.opendocument.text-template	Yes	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.document	No	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.template	No	Yes
application/vnd.sun.xml.writer	Yes	Yes
application/vnd.sun.xml.writer.template	Yes	Yes
application/wordperfect	No	Yes
application/xhtml+xml	Yes	No
image/bmp	Yes	No
image/cgm	Yes	No
image/gif	Yes	No
image/ief	Yes	No
image/jp2	Yes	No
image/jpeg	Yes	No
image/png	Yes	No
image/tiff	Yes	No
image/vnd.adobe.photoshop	Yes	No
image/vnd.adobe.premiere	Yes	No
image/x-cmu-raster	Yes	No
image/x-dwt	Yes	No
image/x-portable-anymap	Yes	No
image/x-portable-bitmap	Yes	No
image/x-portable-graymap	Yes	No
image/x-portable-pixmap	Yes	No
image/x-raw-adobe	Yes	No
image/x-raw-canon	Yes	No
image/x-raw-fuji	Yes	No
image/x-raw-hasselblad	Yes	No
image/x-raw-kodak	Yes	No
image/x-raw-leica	Yes	No
image/x-raw-minolta	Yes	No
image/x-raw-nikon	Yes	No
image/x-raw-olympus	Yes	No
image/x-raw-panasonic	Yes	No
image/x-raw-pentax	Yes	No
image/x-raw-red	Yes	No
image/x-raw-sigma	Yes	No
image/x-raw-sony	Yes	No
image/x-bitmap	Yes	No

Format	Transformable to:	Transformable from:
image/x-xpixmap	Yes	No
image/x-xwindowdump	Yes	No
text/html	Yes	Yes
text/plain	Yes	Yes
text/xml	Yes	No

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 - gml, 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	Yes
application/pdf	Yes
application/xhtml+xml	Yes
image/bmp	Yes
image/cgm	Yes
image/gif	Yes
image/ief	Yes
image/jp2	Yes
image/jpeg	Yes
image/png	Yes
image/tiff	Yes
image/vnd.adobe.photoshop	Yes
image/vnd.adobe.premiere	Yes
image/x-cmu-raster	Yes
image/x-dwt	Yes
image/x-portable-anymap	Yes
image/x-portable-bitmap	Yes
image/x-portable-graymap	Yes
image/x-portable-pixmap	Yes
image/x-raw-adobe	Yes
image/x-raw-canon	Yes
image/x-raw-fuji	Yes
image/x-raw-hasselblad	Yes
image/x-raw-kodak	Yes
image/x-raw-leica	Yes
image/x-raw-minolta	Yes
image/x-raw-nikon	Yes
image/x-raw-olympus	Yes
image/x-raw-panasonic	Yes
image/x-raw-pentax	Yes
image/x-raw-red	Yes

Format	Transformable to:
image/x-raw-sigma	Yes
image/x-raw-sony	Yes
image/x-xbitmap	Yes
image/x-xpixmap	Yes
image/x-xwindowdump	Yes
text/html	Yes
text/plain	Yes
text/xml	Yes

application/vnd.ms-excel - xls

Format	Transformable to:	Transformable from:
application/eps	Yes	No
application/pdf	Yes	No
application/vnd.ms-excel.sheet.binary.macroenabled.12	No	Yes
application/vnd.ms-excel.sheet.macroenabled.12	No	Yes
application/vnd.ms-excel.template.macroenabled.12	No	Yes
application/vnd.oasis.opendocument.spreadsheet	Yes	Yes
application/vnd.oasis.opendocument.spreadsheet-template	Yes	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	No	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.template	No	Yes
application/vnd.sun.xml.calc	Yes	Yes
application/vnd.sun.xml.calc.template	Yes	Yes
application/xhtml+xml	Yes	No
image/bmp	Yes	No
image/cgm	Yes	No
image/gif	Yes	No
image/ief	Yes	No
image/jp2	Yes	No
image/jpeg	Yes	No
image/png	Yes	No
image/tiff	Yes	No
image/vnd.adobe.photoshop	Yes	No
image/vnd.adobe.premiere	Yes	No
image/x-cmu-raster	Yes	No
image/x-dwt	Yes	No
image/x-portable-anymap	Yes	No
image/x-portable-bitmap	Yes	No
image/x-portable-graymap	Yes	No
image/x-portable-pixmap	Yes	No
image/x-raw-adobe	Yes	No
image/x-raw-canon	Yes	No
image/x-raw-fuji	Yes	No
image/x-raw-hasselblad	Yes	No
image/x-raw-kodak	Yes	No
image/x-raw-leica	Yes	No

Format	Transformable to:	Transformable from:
image/x-raw-minolta	Yes	No
image/x-raw-nikon	Yes	No
image/x-raw-olympus	Yes	No
image/x-raw-panasonic	Yes	No
image/x-raw-pentax	Yes	No
image/x-raw-red	Yes	No
image/x-raw-sigma	Yes	No
image/x-raw-sony	Yes	No
image/x-bitmap	Yes	No
image/x-pixmap	Yes	No
image/x-xwindowdump	Yes	No
text/csv	Yes	No
text/html	Yes	No
text/plain	Yes	No
text/xml	Yes	No

application/vnd.ms-excel.addin.macroenabled.12 - xlam

 This format cannot be generated from any other format.

Format	Transformable to:
application/eps	Yes
application/xhtml+xml	Yes
image/bmp	Yes
image/cgm	Yes
image/gif	Yes
image/ief	Yes
image/jp2	Yes
image/jpeg	Yes
image/png	Yes
image/tiff	Yes
image/vnd.adobe.photoshop	Yes
image/vnd.adobe.premiere	Yes
image/x-cmu-raster	Yes
image/x-dwt	Yes
image/x-portable-anymap	Yes
image/x-portable-bitmap	Yes
image/x-portable-graymap	Yes
image/x-portable-pixmap	Yes
image/x-raw-adobe	Yes
image/x-raw-canon	Yes
image/x-raw-fuji	Yes
image/x-raw-hasselblad	Yes
image/x-raw-kodak	Yes
image/x-raw-leica	Yes
image/x-raw-minolta	Yes
image/x-raw-nikon	Yes
image/x-raw-olympus	Yes

Format	Transformable to:
image/x-raw-panasonic	Yes
image/x-raw-pentax	Yes
image/x-raw-red	Yes
image/x-raw-sigma	Yes
image/x-raw-sony	Yes
image/x-bitmap	Yes
image/x-pixmap	Yes
image/x-xwindowdump	Yes
text/html	Yes
text/plain	Yes
text/xml	Yes

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 - xlsm, xltx, xltx

 These formats cannot be generated from any other format.

Format	Transformable to:
application/eps	Yes
application/pdf	Yes
application/vnd.ms-excel	Yes
application/vnd.oasis.opendocument.spreadsheet	Yes
application/vnd.oasis.opendocument.spreadsheet-template	Yes
application/vnd.sun.xml.calc	Yes
application/vnd.sun.xml.calc.template	Yes
application/xhtml+xml	Yes
image/bmp	Yes
image/cgm	Yes
image/gif	Yes
image/ief	Yes
image/jp2	Yes
image/jpeg	Yes
image/png	Yes
image/tiff	Yes
image/vnd.adobe.photoshop	Yes
image/vnd.adobe.premiere	Yes
image/x-cmu-raster	Yes
image/x-dwt	Yes
image/x-portable-anymap	Yes
image/x-portable-bitmap	Yes
image/x-portable-graymap	Yes
image/x-portable-pixmap	Yes
image/x-raw-adobe	Yes
image/x-raw-canon	Yes
image/x-raw-fuji	Yes
image/x-raw-hasselblad	Yes

Format	Transformable to:
image/x-raw-kodak	Yes
image/x-raw-leica	Yes
image/x-raw-minolta	Yes
image/x-raw-nikon	Yes
image/x-raw-olympus	Yes
image/x-raw-panasonic	Yes
image/x-raw-pentax	Yes
image/x-raw-red	Yes
image/x-raw-sigma	Yes
image/x-raw-sony	Yes
image/x-xbitmap	Yes
image/x-xpixmap	Yes
image/x-xwindowdump	Yes
text/html	Yes
text/plain	Yes
text/xml	Yes

application/vnd.ms-outlook - msg

 This format cannot be generated from any other format.

Format	Transformable to:
application/eps	Yes
application/msword	Yes
application/pdf	Yes
application/rtf	Yes
application/vnd.oasis.opendocument.text	Yes
application/vnd.oasis.opendocument.text-template	Yes
application/vnd.sun.xml.writer	Yes
application/vnd.sun.xml.writer.template	Yes
application/xhtml+xml	Yes
image/bmp	Yes
image/cgm	Yes
image/gif	Yes
image/ief	Yes
image/jp2	Yes
image/jpeg	Yes
image/png	Yes
image/tiff	Yes
image/vnd.adobe.photoshop	Yes
image/vnd.adobe.premiere	Yes
image/x-cmu-raster	Yes
image/x-dwt	Yes
image/x-portable-anymap	Yes
image/x-portable-bitmap	Yes
image/x-portable-graymap	Yes
image/x-portable-pixmap	Yes
image/x-raw-adobe	Yes

Format	Transformable to:
image/x-raw-canon	Yes
image/x-raw-fuji	Yes
image/x-raw-hasselblad	Yes
image/x-raw-kodak	Yes
image/x-raw-leica	Yes
image/x-raw-minolta	Yes
image/x-raw-nikon	Yes
image/x-raw-olympus	Yes
image/x-raw-panasonic	Yes
image/x-raw-pentax	Yes
image/x-raw-red	Yes
image/x-raw-sigma	Yes
image/x-raw-sony	Yes
image/x-bitmap	Yes
image/x-xpixmap	Yes
image/x-xwindowdump	Yes
text/html	Yes
text/plain	Yes
text/xml	Yes

application/vnd.ms-powerpoint - ppt

Format	Transformable to:	Transformable from:
application/eps	Yes	No
application/pdf	Yes	No
application/vnd.ms-powerpoint.addin.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.presentation.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.slide.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.template.macroenabled.12	No	Yes
application/vnd.oasis.opendocument.graphics	Yes	Yes
application/vnd.oasis.opendocument.presentation	Yes	Yes
application/vnd.oasis.opendocument.presentation-template	Yes	Yes
application/vnd.openxmlformats-officedocument.presentationml.presentation	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.slide	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.template	No	Yes
application/vnd.sun.xml.impress	Yes	Yes
application/vnd.sun.xml.impress.template	Yes	Yes
application/vnd.visio	Yes	Yes
application/xhtml+xml	Yes	No
image/bmp	Yes	No
image/cgm	Yes	No
image/gif	Yes	No
image/ief	Yes	No
image/jp2	Yes	No

Format	Transformable to:	Transformable from:
image/jpeg	Yes	No
image/png	Yes	No
image/tiff	Yes	No
image/vnd.adobe.photoshop	Yes	No
image/vnd.adobe.premiere	Yes	No
image/x-cmu-raster	Yes	No
image/x-dwt	Yes	No
image/x-portable-anymap	Yes	No
image/x-portable-bitmap	Yes	No
image/x-portable-graymap	Yes	No
image/x-portable-pixmap	Yes	No
image/x-raw-adobe	Yes	No
image/x-raw-canon	Yes	No
image/x-raw-fuji	Yes	No
image/x-raw-hasselblad	Yes	No
image/x-raw-kodak	Yes	No
image/x-raw-leica	Yes	No
image/x-raw-minolta	Yes	No
image/x-raw-nikon	Yes	No
image/x-raw-olympus	Yes	No
image/x-raw-panasonic	Yes	No
image/x-raw-pentax	Yes	No
image/x-raw-red	Yes	No
image/x-raw-sigma	Yes	No
image/x-raw-sony	Yes	No
image/x-bitmap	Yes	No
image/x-xpixmap	Yes	No
image/x-xwindowdump	Yes	No
text/html	Yes	No
text/plain	Yes	No
text/xml	Yes	No

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	Yes
application/pdf	Yes
application/vnd.ms-powerpoint	Yes
application/vnd.oasis.opendocument.graphics	Yes
application/vnd.oasis.opendocument.presentation	Yes
application/vnd.oasis.opendocument.presentation-template	Yes
application/vnd.sun.xml.impress	Yes
application/vnd.sun.xml.impress.template	Yes
application/vnd.visio	Yes

Format	Transformable to:
application/xhtml+xml	Yes
image/bmp	Yes
image/cgm	Yes
image/gif	Yes
image/ief	Yes
image/jp2	Yes
image/jpeg	Yes
image/png	Yes
image/tiff	Yes
image/vnd.adobe.photoshop	Yes
image/vnd.adobe.premiere	Yes
image/x-cmu-raster	Yes
image/x-dwt	Yes
image/x-portable-anymap	Yes
image/x-portable-bitmap	Yes
image/x-portable-graymap	Yes
image/x-portable-pixmap	Yes
image/x-raw-adobe	Yes
image/x-raw-canon	Yes
image/x-raw-fuji	Yes
image/x-raw-hasselblad	Yes
image/x-raw-kodak	Yes
image/x-raw-leica	Yes
image/x-raw-minolta	Yes
image/x-raw-nikon	Yes
image/x-raw-olympus	Yes
image/x-raw-panasonic	Yes
image/x-raw-pentax	Yes
image/x-raw-red	Yes
image/x-raw-sigma	Yes
image/x-raw-sony	Yes
image/x-xbitmap	Yes
image/x-xpixmap	Yes
image/x-xwindowdump	Yes
text/html	Yes
text/plain	Yes
text/xml	Yes

application/vnd.ms-powerpoint.slide.macroenabled.12 - sldm

 This format cannot be generated from any other format.

Format	Transformable to:
application/eps	Yes
application/pdf	Yes
application/vnd.ms-powerpoint	Yes
application/vnd.oasis.opendocument.graphics	Yes
application/vnd.oasis.opendocument.presentation	Yes

Format	Transformable to:
application/vnd.oasis.opendocument.presentation-template	Yes
application/vnd.sun.xml.impress	Yes
application/vnd.sun.xml.impress.template	Yes
application/vnd.visio	Yes
application/xhtml+xml	Yes
image/bmp	Yes
image/cgm	Yes
image/gif	Yes
image/ief	Yes
image/jp2	Yes
image/jpeg	Yes
image/png	Yes
image/tiff	Yes
image/vnd.adobe.photoshop	Yes
image/vnd.adobe.premiere	Yes
image/x-cmu-raster	Yes
image/x-dwt	Yes
image/x-portable-anymap	Yes
image/x-portable-bitmap	Yes
image/x-portable-graymap	Yes
image/x-portable-pixmap	Yes
image/x-raw-adobe	Yes
image/x-raw-canon	Yes
image/x-raw-fuji	Yes
image/x-raw-hasselblad	Yes
image/x-raw-kodak	Yes
image/x-raw-leica	Yes
image/x-raw-minolta	Yes
image/x-raw-nikon	Yes
image/x-raw-olympus	Yes
image/x-raw-panasonic	Yes
image/x-raw-pentax	Yes
image/x-raw-red	Yes
image/x-raw-sigma	Yes
image/x-raw-sony	Yes
image/x-bitmap	Yes
image/x-xpixmap	Yes
image/x-xwindowdump	Yes
text/html	Yes
text/xml	Yes

application/vnd.ms-powerpoint.slideshow.macroenabled.12 - ppsm

 This format cannot be generated from any other format.

Format	Transformable to:
application/eps	Yes
application/xhtml+xml	Yes

Format	Transformable to:
image/bmp	Yes
image/cgm	Yes
image/gif	Yes
image/ief	Yes
image/jp2	Yes
image/jpeg	Yes
image/png	Yes
image/tiff	Yes
image/vnd.adobe.photoshop	Yes
image/vnd.adobe.premiere	Yes
image/x-cmu-raster	Yes
image/x-dwt	Yes
image/x-portable-anymap	Yes
image/x-portable-bitmap	Yes
image/x-portable-graymap	Yes
image/x-portable-pixmap	Yes
image/x-raw-adobe	Yes
image/x-raw-canon	Yes
image/x-raw-fuji	Yes
image/x-raw-hasselblad	Yes
image/x-raw-kodak	Yes
image/x-raw-leica	Yes
image/x-raw-minolta	Yes
image/x-raw-nikon	Yes
image/x-raw-olympus	Yes
image/x-raw-panasonic	Yes
image/x-raw-pentax	Yes
image/x-raw-red	Yes
image/x-raw-sigma	Yes
image/x-raw-sony	Yes
image/x-xbitmap	Yes
image/x-xpixmap	Yes
image/x-xwindowdump	Yes
text/html	Yes
text/plain	Yes
text/xml	Yes

application/vnd.ms-project - mpp

 This format cannot be generated from any other format.

Format	Transformable to:
application/xhtml+xml	Yes
text/html	Yes
text/plain	Yes
text/xml	Yes

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	Yes
application/msword	Yes
application/pdf	Yes
application/rtf	Yes
application/vnd.oasis.opendocument.text	Yes
application/vnd.oasis.opendocument.text-template	Yes
application/vnd.sun.xml.writer	Yes
application/vnd.sun.xml.writer.template	Yes
application/xhtml+xml	Yes
image/bmp	Yes
image/cgm	Yes
image/gif	Yes
image/ief	Yes
image/jp2	Yes
image/jpeg	Yes
image/png	Yes
image/tiff	Yes
image/vnd.adobe.photoshop	Yes
image/vnd.adobe.premiere	Yes
image/x-cmu-raster	Yes
image/x-dwt	Yes
image/x-portable-anymap	Yes
image/x-portable-bitmap	Yes
image/x-portable-graymap	Yes
image/x-portable-pixmap	Yes
image/x-raw-adobe	Yes
image/x-raw-canon	Yes
image/x-raw-fuji	Yes
image/x-raw-hasselblad	Yes
image/x-raw-kodak	Yes
image/x-raw-leica	Yes
image/x-raw-minolta	Yes
image/x-raw-nikon	Yes
image/x-raw-olympus	Yes
image/x-raw-panasonic	Yes
image/x-raw-pentax	Yes
image/x-raw-red	Yes
image/x-raw-sigma	Yes
image/x-raw-sony	Yes
image/x-bitmap	Yes
image/x-pixmap	Yes
image/x-xwindowdump	Yes
text/html	Yes

Format	Transformable to:
text/plain	Yes
text/xml	Yes

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	Yes
text/html	Yes
text/plain	Yes
text/xml	Yes

application/vnd.oasis.opendocument.database, application/vnd.oasis.opendocument.formula and application/vnd.oasis.opendocument.graphics-template - odb, odf, otg

These formats cannot be transformed into, or generated from, any other format.

application/vnd.oasis.opendocument.graphics - odg

Format	Transformable to:	Transformable from:
application/eps	Yes	No
application/pdf	Yes	No
application/vnd.ms-powerpoint	Yes	Yes
application/vnd.ms-powerpoint.addin.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.presentation.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.slide.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.template.macroenabled.12	No	Yes
application/vnd.oasis.opendocument.presentation	Yes	Yes
application/vnd.oasis.opendocument.presentation-template	Yes	Yes
application/vnd.openxmlformats-officedocument.presentationml.presentation	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.slide	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.template	No	Yes
application/vnd.sun.xml.impress	Yes	Yes
application/vnd.sun.xml.impress.template	Yes	Yes
application/vnd.visio	Yes	Yes
application/xhtml+xml	Yes	No
image/bmp	Yes	No
image/cgm	Yes	No
image/gif	Yes	No
image/ief	Yes	No
image/jp2	Yes	No
image/jpeg	Yes	No
image/png	Yes	No

Format	Transformable to:	Transformable from:
image/tiff	Yes	No
image/vnd.adobe.photoshop	Yes	No
image/vnd.adobe.premiere	Yes	No
image/x-cmu-raster	Yes	No
image/x-dwt	Yes	No
image/x-portable-anymap	Yes	No
image/x-portable-bitmap	Yes	No
image/x-portable-graymap	Yes	No
image/x-portable-pixmap	Yes	No
image/x-raw-adobe	Yes	No
image/x-raw-canon	Yes	No
image/x-raw-fuji	Yes	No
image/x-raw-hasselblad	Yes	No
image/x-raw-kodak	Yes	No
image/x-raw-leica	Yes	No
image/x-raw-minolta	Yes	No
image/x-raw-nikon	Yes	No
image/x-raw-olympus	Yes	No
image/x-raw-panasonic	Yes	No
image/x-raw-pentax	Yes	No
image/x-raw-red	Yes	No
image/x-raw-sigma	Yes	No
image/x-raw-sony	Yes	No
image/x-bitmap	Yes	No
image/x-pixmap	Yes	No
image/x-xwindowdump	Yes	No
text/html	Yes	No
text/plain	Yes	No
text/xml	Yes	No

application/vnd.oasis.opendocument.presentation - odp

Format	Transformable to:	Transformable from:
application/eps	Yes	No
application/pdf	Yes	No
application/vnd.ms-powerpoint	Yes	Yes
application/vnd.ms-powerpoint.addin.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.presentation.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.slide.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.template.macroenabled.12	No	Yes
application/vnd.oasis.opendocument.graphics	Yes	Yes
application/vnd.oasis.opendocument.presentation-template	Yes	Yes
application/vnd.openxmlformats-officedocument.presentationml.presentation	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.slide	No	Yes

Format	Transformable to:	Transformable from:
application/vnd.openxmlformats-officedocument.presentationml.template	No	Yes
application/vnd.sun.xml.impress	Yes	Yes
application/vnd.sun.xml.impress.template	Yes	Yes
application/vnd.visio	Yes	Yes
application/xhtml+xml	Yes	No
image/bmp	Yes	No
image/cgm	Yes	No
image/gif	Yes	No
image/ief	Yes	No
image/jp2	Yes	No
image/jpeg	Yes	No
image/png	Yes	No
image/tiff	Yes	No
image/vnd.adobe.photoshop	Yes	No
image/vnd.adobe.premiere	Yes	No
image/x-cmu-raster	Yes	No
image/x-dwt	Yes	No
image/x-portable-anymap	Yes	No
image/x-portable-bitmap	Yes	No
image/x-portable-graymap	Yes	No
image/x-portable-pixmap	Yes	No
image/x-raw-adobe	Yes	No
image/x-raw-canon	Yes	No
image/x-raw-fuji	Yes	No
image/x-raw-hasselblad	Yes	No
image/x-raw-kodak	Yes	No
image/x-raw-leica	Yes	No
image/x-raw-minolta	Yes	No
image/x-raw-nikon	Yes	No
image/x-raw-olympus	Yes	No
image/x-raw-panasonic	Yes	No
image/x-raw-pentax	Yes	No
image/x-raw-red	Yes	No
image/x-raw-sigma	Yes	No
image/x-raw-sony	Yes	No
image/x-xbitmap	Yes	No
image/x-xpixmap	Yes	No
image/x-xwindowdump	Yes	No
text/html	Yes	No
text/plain	Yes	No
text/xml	Yes	No

application/vnd.oasis.opendocument.presentation-template - otp

Format	Transformable to:	Transformable from:
application/eps	Yes	No
application/pdf	Yes	No
application/vnd.ms-powerpoint	Yes	Yes
application/vnd.ms-powerpoint.addin.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.presentation.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.slide.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.template.macroenabled.12	No	Yes
application/vnd.oasis.opendocument.graphics	Yes	Yes
application/vnd.oasis.opendocument.presentation	Yes	Yes
application/vnd.openxmlformats-officedocument.presentationml.presentation	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.slide	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.template	No	Yes
application/vnd.sun.xml.impress	Yes	Yes
application/vnd.sun.xml.impress.template	Yes	Yes
application/vnd.visio	Yes	Yes
application/xhtml+xml	Yes	No
image/bmp	Yes	No
image/cgm	Yes	No
image/gif	Yes	No
image/ief	Yes	No
image/jp2	Yes	No
image/jpeg	Yes	No
image/png	Yes	No
image/tiff	Yes	No
image/vnd.adobe.photoshop	Yes	No
image/vnd.adobe.premiere	Yes	No
image/x-cmu-raster	Yes	No
image/x-dwt	Yes	No
image/x-portable-anymap	Yes	No
image/x-portable-bitmap	Yes	No
image/x-portable-graymap	Yes	No
image/x-portable-pixmap	Yes	No
image/x-raw-adobe	Yes	No
image/x-raw-canon	Yes	No
image/x-raw-fuji	Yes	No
image/x-raw-hasselblad	Yes	No
image/x-raw-kodak	Yes	No
image/x-raw-leica	Yes	No
image/x-raw-minolta	Yes	No
image/x-raw-nikon	Yes	No
image/x-raw-olympus	Yes	No
image/x-raw-panasonic	Yes	No
image/x-raw-pentax	Yes	No

Format	Transformable to:	Transformable from:
image/x-raw-red	Yes	No
image/x-raw-sigma	Yes	No
image/x-raw-sony	Yes	No
image/x-xbitmap	Yes	No
image/x-xpixmap	Yes	No
image/x-xwindowdump	Yes	No
text/html	Yes	No
text/plain	Yes	No
text/xml	Yes	No

application/vnd.oasis.opendocument.spreadsheet - ods

Format	Transformable to:	Transformable from:
application/eps	Yes	No
application/pdf	Yes	No
application/vnd.ms-excel	Yes	Yes
application/vnd.ms-excel.sheet.binary.macroenabled.12	No	Yes
application/vnd.ms-excel.sheet.macroenabled.12	No	Yes
application/vnd.ms-excel.template.macroenabled.12	No	Yes
application/vnd.oasis.opendocument.spreadsheet-template	Yes	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	No	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.template	No	Yes
application/vnd.sun.xml.calc	Yes	Yes
application/vnd.sun.xml.calc.template	Yes	Yes
application/xhtml+xml	Yes	No
image/bmp	Yes	No
image/cgm	Yes	No
image/gif	Yes	No
image/ief	Yes	No
image/jp2	Yes	No
image/jpeg	Yes	No
image/png	Yes	No
image/tiff	Yes	No
image/vnd.adobe.photoshop	Yes	No
image/vnd.adobe.premiere	Yes	No
image/x-cmu-raster	Yes	No
image/x-dwt	Yes	No
image/x-portable-anymap	Yes	No
image/x-portable-bitmap	Yes	No
image/x-portable-graymap	Yes	No
image/x-portable-pixmap	Yes	No
image/x-raw-adobe	Yes	No
image/x-raw-canon	Yes	No
image/x-raw-fuji	Yes	No
image/x-raw-hasselblad	Yes	No

Format	Transformable to:	Transformable from:
image/x-raw-kodak	Yes	No
image/x-raw-leica	Yes	No
image/x-raw-minolta	Yes	No
image/x-raw-nikon	Yes	No
image/x-raw-olympus	Yes	No
image/x-raw-panasonic	Yes	No
image/x-raw-pentax	Yes	No
image/x-raw-red	Yes	No
image/x-raw-sigma	Yes	No
image/x-raw-sony	Yes	No
image/x-xbitmap	Yes	No
image/x-xpixmap	Yes	No
image/x-xwindowdump	Yes	No
text/html	Yes	No
text/plain	Yes	No
text/xml	Yes	No

application/vnd.oasis.opendocument.spreadsheet-template - ots

Format	Transformable to:	Transformable from:
application/eps	Yes	No
application/pdf	Yes	No
application/vnd.ms-excel	Yes	Yes
application/vnd.ms-excel.sheet.binary.macroenabled.12	No	Yes
application/vnd.ms-excel.sheet.macroenabled.12	No	Yes
application/vnd.ms-excel.template.macroenabled.12	No	Yes
application/vnd.oasis.opendocument.spreadsheet	Yes	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	No	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.template	No	Yes
application/vnd.sun.xml.calc	Yes	Yes
application/vnd.sun.xml.calc.template	Yes	Yes
application/xhtml+xml	Yes	No
image/bmp	Yes	No
image/cgm	Yes	No
image/gif	Yes	No
image/ief	Yes	No
image/jp2	Yes	No
image/jpeg	Yes	No
image/png	Yes	No
image/tiff	Yes	No
image/vnd.adobe.photoshop	Yes	No
image/vnd.adobe.premiere	Yes	No
image/x-cmu-raster	Yes	No
image/x-dwt	Yes	No
image/x-portable-anymap	Yes	No
image/x-portable-bitmap	Yes	No

Format	Transformable to:	Transformable from:
image/x-portable-graymap	Yes	No
image/x-portable-pixmap	Yes	No
image/x-raw-adobe	Yes	No
image/x-raw-canon	Yes	No
image/x-raw-fuji	Yes	No
image/x-raw-hasselblad	Yes	No
image/x-raw-kodak	Yes	No
image/x-raw-leica	Yes	No
image/x-raw-minolta	Yes	No
image/x-raw-nikon	Yes	No
image/x-raw-olympus	Yes	No
image/x-raw-panasonic	Yes	No
image/x-raw-pentax	Yes	No
image/x-raw-red	Yes	No
image/x-raw-sigma	Yes	No
image/x-raw-sony	Yes	No
image/x-bitmap	Yes	No
image/x-xpixmap	Yes	No
image/x-xwindowdump	Yes	No
text/html	Yes	No
text/plain	Yes	No
text/xml	Yes	No

application/vnd.oasis.opendocument.text - odt

Format	Transformable to:	Transformable from:
application/eps	Yes	No
application/msword	Yes	Yes
application/pdf	Yes	No
application/rtf	Yes	Yes
application/vnd.ms-outlook	No	Yes
application/vnd.ms-word.document.macroenabled.12	No	Yes
application/vnd.ms-word.template.macroenabled.12	No	Yes
application/vnd.oasis.opendocument.text-template	Yes	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.document	No	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.template	No	Yes
application/vnd.sun.xml.writer	Yes	Yes
application/vnd.sun.xml.writer.template	Yes	Yes
application/wordperfect	No	Yes
application/xhtml+xml	Yes	No
image/bmp	Yes	No
image/cgm	Yes	No
image/gif	Yes	No
image/ief	Yes	No
image/jp2	Yes	No
image/jpeg	Yes	No

Format	Transformable to:	Transformable from:
image/png	Yes	No
image/tiff	Yes	No
image/vnd.adobe.photoshop	Yes	No
image/vnd.adobe.premiere	Yes	No
image/x-cmu-raster	Yes	No
image/x-dwt	Yes	No
image/x-portable-anymap	Yes	No
image/x-portable-bitmap	Yes	No
image/x-portable-graymap	Yes	No
image/x-portable-pixmap	Yes	No
image/x-raw-adobe	Yes	No
image/x-raw-canon	Yes	No
image/x-raw-fuji	Yes	No
image/x-raw-hasselblad	Yes	No
image/x-raw-kodak	Yes	No
image/x-raw-leica	Yes	No
image/x-raw-minolta	Yes	No
image/x-raw-nikon	Yes	No
image/x-raw-olympus	Yes	No
image/x-raw-panasonic	Yes	No
image/x-raw-pentax	Yes	No
image/x-raw-red	Yes	No
image/x-raw-sigma	Yes	No
image/x-raw-sony	Yes	No
image/x-xbitmap	Yes	No
image/x-xpixmap	Yes	No
image/x-xwindowdump	Yes	No
text/html	Yes	Yes
text/plain	Yes	Yes
text/xml	Yes	No

application/vnd.oasis.opendocument.text-template - ott

Format	Transformable to:	Transformable from:
application/eps	Yes	No
application/msword	Yes	Yes
application/pdf	Yes	No
application/rtf	Yes	Yes
application/vnd.ms-outlook	No	Yes
application/vnd.ms-word.document.macroenabled.12	No	Yes
application/vnd.ms-word.template.macroenabled.12	No	Yes
application/vnd.oasis.opendocument.text	Yes	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.document	No	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.template	No	Yes
application/vnd.sun.xml.writer	Yes	Yes
application/vnd.sun.xml.writer.template	Yes	Yes

Format	Transformable to:	Transformable from:
application/wordperfect	No	Yes
application/xhtml+xml	Yes	No
image/bmp	Yes	No
image/cgm	Yes	No
image/gif	Yes	No
image/ief	Yes	No
image/jp2	Yes	No
image/jpeg	Yes	No
image/png	Yes	No
image/tiff	Yes	No
image/vnd.adobe.photoshop	Yes	No
image/vnd.adobe.premiere	Yes	No
image/x-cmu-raster	Yes	No
image/x-dwt	Yes	No
image/x-portable-anymap	Yes	No
image/x-portable-bitmap	Yes	No
image/x-portable-graymap	Yes	No
image/x-portable-pixmap	Yes	No
image/x-raw-adobe	Yes	No
image/x-raw-canon	Yes	No
image/x-raw-fuji	Yes	No
image/x-raw-hasselblad	Yes	No
image/x-raw-kodak	Yes	No
image/x-raw-leica	Yes	No
image/x-raw-minolta	Yes	No
image/x-raw-nikon	Yes	No
image/x-raw-olympus	Yes	No
image/x-raw-panasonic	Yes	No
image/x-raw-pentax	Yes	No
image/x-raw-red	Yes	No
image/x-raw-sigma	Yes	No
image/x-raw-sony	Yes	No
image/x-xbitmap	Yes	No
image/x-xpixmap	Yes	No
image/x-xwindowdump	Yes	No
text/html	Yes	Yes
text/plain	Yes	Yes
text/xml	Yes	No

application/vnd.openxmlformats-officedocument.presentationml.presentation - pptx

 This format cannot be generated from any other format.

Format	Transformable to:
application/eps	Yes
application/pdf	Yes
application/vnd.ms-powerpoint	Yes
application/vnd.oasis.opendocument.graphics	Yes

Format	Transformable to:
application/vnd.oasis.opendocument.presentation	Yes
application/vnd.oasis.opendocument.presentation-template	Yes
application/vnd.sun.xml.impress	Yes
application/vnd.sun.xml.impress.template	Yes
application/vnd.visio	Yes
application/xhtml+xml	Yes
image/bmp	Yes
image/cgm	Yes
image/gif	Yes
image/ief	Yes
image/jp2	Yes
image/jpeg	Yes
image/png	Yes
image/tiff	Yes
image/vnd.adobe.photoshop	Yes
image/vnd.adobe.premiere	Yes
image/x-cmu-raster	Yes
image/x-dwt	Yes
image/x-portable-anymap	Yes
image/x-portable-bitmap	Yes
image/x-portable-graymap	Yes
image/x-portable-pixmap	Yes
image/x-raw-adobe	Yes
image/x-raw-canon	Yes
image/x-raw-fuji	Yes
image/x-raw-hasselblad	Yes
image/x-raw-kodak	Yes
image/x-raw-leica	Yes
image/x-raw-minolta	Yes
image/x-raw-nikon	Yes
image/x-raw-olympus	Yes
image/x-raw-panasonic	Yes
image/x-raw-pentax	Yes
image/x-raw-red	Yes
image/x-raw-sigma	Yes
image/x-raw-sony	Yes
image/x-xbitmap	Yes
image/x-xpixmap	Yes
image/x-xwindowdump	Yes
text/html	Yes
text/plain	Yes
text/xml	Yes

application/vnd.openxmlformats-officedocument.presentationml.slide - sldx

 This format cannot be generated from any other format.

Format	Transformable to:
application/eps	Yes
application/pdf	Yes
application/vnd.ms-powerpoint	Yes
application/vnd.oasis.opendocument.graphics	Yes
application/vnd.oasis.opendocument.presentation	Yes
application/vnd.oasis.opendocument.presentation-template	Yes
application/vnd.sun.xml.impress	Yes
application/vnd.sun.xml.impress.template	Yes
application/vnd.visio	Yes
application/xhtml+xml	Yes
image/bmp	Yes
image/cgm	Yes
image/gif	Yes
image/ief	Yes
image/jp2	Yes
image/jpeg	Yes
image/png	Yes
image/tiff	Yes
image/vnd.adobe.photoshop	Yes
image/vnd.adobe.premiere	Yes
image/x-cmu-raster	Yes
image/x-dwt	Yes
image/x-portable-anymap	Yes
image/x-portable-bitmap	Yes
image/x-portable-graymap	Yes
image/x-portable-pixmap	Yes
image/x-raw-adobe	Yes
image/x-raw-canon	Yes
image/x-raw-fuji	Yes
image/x-raw-hasselblad	Yes
image/x-raw-kodak	Yes
image/x-raw-leica	Yes
image/x-raw-minolta	Yes
image/x-raw-nikon	Yes
image/x-raw-olympus	Yes
image/x-raw-panasonic	Yes
image/x-raw-pentax	Yes
image/x-raw-red	Yes
image/x-raw-sigma	Yes
image/x-raw-sony	Yes
image/x-bitmap	Yes
image/x-xpixmap	Yes
image/x-xwindowdump	Yes
text/html	Yes
text/xml	Yes

application/vnd.openxmlformats-officedocument.presentationml.slideshow - ppsx



This format cannot be generated from any other format.

Format	Transformable to:
application/eps	Yes
application/xhtml+xml	Yes
image/bmp	Yes
image/cgm	Yes
image/gif	Yes
image/ief	Yes
image/jp2	Yes
image/jpeg	Yes
image/png	Yes
image/tiff	Yes
image/vnd.adobe.photoshop	Yes
image/vnd.adobe.premiere	Yes
image/x-cmu-raster	Yes
image/x-dwt	Yes
image/x-portable-anymap	Yes
image/x-portable-bitmap	Yes
image/x-portable-graymap	Yes
image/x-portable-pixmap	Yes
image/x-raw-adobe	Yes
image/x-raw-canon	Yes
image/x-raw-fuji	Yes
image/x-raw-hasselblad	Yes
image/x-raw-kodak	Yes
image/x-raw-leica	Yes
image/x-raw-minolta	Yes
image/x-raw-nikon	Yes
image/x-raw-olympus	Yes
image/x-raw-panasonic	Yes
image/x-raw-pentax	Yes
image/x-raw-red	Yes
image/x-raw-sigma	Yes
image/x-raw-sony	Yes
image/x-bitmap	Yes
image/x-xpixmap	Yes
image/x-xwindowdump	Yes
text/html	Yes
text/plain	Yes
text/xml	Yes

application/vnd.openxmlformats-officedocument.presentationml.template - potx



This format cannot be generated from any other format.

Format	Transformable to:
application/eps	Yes
application/pdf	Yes
application/vnd.ms-powerpoint	Yes

Format	Transformable to:
application/vnd.oasis.opendocument.graphics	Yes
application/vnd.oasis.opendocument.presentation	Yes
application/vnd.oasis.opendocument.presentation-template	Yes
application/vnd.sun.xml.impress	Yes
application/vnd.sun.xml.impress.template	Yes
application/vnd.visio	Yes
application/xhtml+xml	Yes
image/bmp	Yes
image/cgm	Yes
image/gif	Yes
image/ief	Yes
image/jp2	Yes
image/jpeg	Yes
image/png	Yes
image/tiff	Yes
image/vnd.adobe.photoshop	Yes
image/vnd.adobe.premiere	Yes
image/x-cmu-raster	Yes
image/x-dwt	Yes
image/x-portable-anymap	Yes
image/x-portable-bitmap	Yes
image/x-portable-graymap	Yes
image/x-portable-pixmap	Yes
image/x-raw-adobe	Yes
image/x-raw-canon	Yes
image/x-raw-fuji	Yes
image/x-raw-hasselblad	Yes
image/x-raw-kodak	Yes
image/x-raw-leica	Yes
image/x-raw-minolta	Yes
image/x-raw-nikon	Yes
image/x-raw-olympus	Yes
image/x-raw-panasonic	Yes
image/x-raw-pentax	Yes
image/x-raw-red	Yes
image/x-raw-sigma	Yes
image/x-raw-sony	Yes
image/x-xbitmap	Yes
image/x-xpixmap	Yes
image/x-xwindowdump	Yes
text/html	Yes
text/plain	Yes
text/xml	Yes

application/vnd.openxmlformats-officedocument.spreadsheetml.sheet - xlsx

 This format cannot be generated from any other format.

Format	Transformable to:
application/eps	Yes
application/pdf	Yes
application/vnd.ms-excel	Yes
application/vnd.oasis.opendocument.spreadsheet	Yes
application/vnd.oasis.opendocument.spreadsheet-template	Yes
application/vnd.sun.xml.calc	Yes
application/vnd.sun.xml.calc.template	Yes
application/xhtml+xml	Yes
image/bmp	Yes
image/cgm	Yes
image/gif	Yes
image/ief	Yes
image/jp2	Yes
image/jpeg	Yes
image/png	Yes
image/tiff	Yes
image/vnd.adobe.photoshop	Yes
image/vnd.adobe.premiere	Yes
image/x-cmu-raster	Yes
image/x-dwt	Yes
image/x-portable-anymap	Yes
image/x-portable-bitmap	Yes
image/x-portable-graymap	Yes
image/x-portable-pixmap	Yes
image/x-raw-adobe	Yes
image/x-raw-canon	Yes
image/x-raw-fuji	Yes
image/x-raw-hasselblad	Yes
image/x-raw-kodak	Yes
image/x-raw-leica	Yes
image/x-raw-minolta	Yes
image/x-raw-nikon	Yes
image/x-raw-olympus	Yes
image/x-raw-panasonic	Yes
image/x-raw-pentax	Yes
image/x-raw-red	Yes
image/x-raw-sigma	Yes
image/x-raw-sony	Yes
image/x-xbitmap	Yes
image/x-xpixmap	Yes
image/x-xwindowdump	Yes
text/csv	Yes
text/html	Yes
text/plain	Yes
text/xml	Yes

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	Yes	No
application/pdf	Yes	No
application/vnd.ms-excel	Yes	Yes
application/vnd.ms-excel.sheet.binary.macroenabled.12	No	Yes
application/vnd.ms-excel.sheet.macroenabled.12	No	Yes
application/vnd.ms-excel.template.macroenabled.12	No	Yes
application/vnd.oasis.opendocument.spreadsheet	Yes	Yes
application/vnd.oasis.opendocument.spreadsheet-template	Yes	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	No	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.template	No	Yes
application/vnd.sun.xml.calc.template	Yes	Yes
application/xhtml+xml	Yes	No
image/bmp	Yes	No
image/cgm	Yes	No
image/gif	Yes	No
image/ief	Yes	No
image/jp2	Yes	No
image/jpeg	Yes	No
image/png	Yes	No
image/tiff	Yes	No
image/vnd.adobe.photoshop	Yes	No
image/vnd.adobe.premiere	Yes	No
image/x-cmu-raster	Yes	No
image/x-dwt	Yes	No
image/x-portable-anymap	Yes	No
image/x-portable-bitmap	Yes	No
image/x-portable-graymap	Yes	No
image/x-portable-pixmap	Yes	No
image/x-raw-adobe	Yes	No
image/x-raw-canon	Yes	No
image/x-raw-fuji	Yes	No
image/x-raw-hasselblad	Yes	No
image/x-raw-kodak	Yes	No
image/x-raw-leica	Yes	No
image/x-raw-minolta	Yes	No
image/x-raw-nikon	Yes	No
image/x-raw-olympus	Yes	No
image/x-raw-panasonic	Yes	No

Format	Transformable to:	Transformable from:
image/x-raw-pentax	Yes	No
image/x-raw-red	Yes	No
image/x-raw-sigma	Yes	No
image/x-raw-sony	Yes	No
image/x-xbitmap	Yes	No
image/x-xpixmap	Yes	No
image/x-xwindowdump	Yes	No
text/html	Yes	No
text/plain	Yes	No
text/xml	Yes	No

application/vnd.sun.xml.calc.template - stc

Format	Transformable to:	Transformable from:
application/eps	Yes	No
application/pdf	Yes	No
application/vnd.ms-excel	Yes	Yes
application/vnd.ms-excel.sheet.binary.macroenabled.12	No	Yes
application/vnd.ms-excel.sheet.macroenabled.12	No	Yes
application/vnd.ms-excel.template.macroenabled.12	No	Yes
application/vnd.oasis.opendocument.spreadsheet	Yes	Yes
application/vnd.oasis.opendocument.spreadsheet-template	Yes	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	No	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.template	No	Yes
application/vnd.sun.xml.calc	Yes	Yes
application/xhtml+xml	Yes	No
image/bmp	Yes	No
image/cgm	Yes	No
image/gif	Yes	No
image/ief	Yes	No
image/jp2	Yes	No
image/jpeg	Yes	No
image/png	Yes	No
image/tiff	Yes	No
image/vnd.adobe.photoshop	Yes	No
image/vnd.adobe.premiere	Yes	No
image/x-cmu-raster	Yes	No
image/x-dwt	Yes	No
image/x-portable-anymap	Yes	No
image/x-portable-bitmap	Yes	No
image/x-portable-graymap	Yes	No
image/x-portable-pixmap	Yes	No
image/x-raw-adobe	Yes	No
image/x-raw-canon	Yes	No
image/x-raw-fuji	Yes	No
image/x-raw-hasselblad	Yes	No

Format	Transformable to:	Transformable from:
image/x-raw-kodak	Yes	No
image/x-raw-leica	Yes	No
image/x-raw-minolta	Yes	No
image/x-raw-nikon	Yes	No
image/x-raw-olympus	Yes	No
image/x-raw-panasonic	Yes	No
image/x-raw-pentax	Yes	No
image/x-raw-red	Yes	No
image/x-raw-sigma	Yes	No
image/x-raw-sony	Yes	No
image/x-xbitmap	Yes	No
image/x-xpixmap	Yes	No
image/x-xwindowdump	Yes	No
text/html	Yes	No
text/plain	Yes	No
text/xml	Yes	No

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	Yes	No
application/pdf	Yes	No
application/vnd.ms-powerpoint	Yes	Yes
application/vnd.ms-powerpoint.addin.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.presentation.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.slide.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.template.macroenabled.12	No	Yes
application/vnd.oasis.opendocument.graphics	Yes	Yes
application/vnd.oasis.opendocument.presentation	Yes	Yes
application/vnd.oasis.opendocument.presentation-template	Yes	Yes
application/vnd.openxmlformats-officedocument.presentationml.presentation	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.slide	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.template	No	Yes
application/vnd.sun.xml.impress.template	Yes	Yes
application/vnd.visio	Yes	Yes
application/xhtml+xml	Yes	No
image/bmp	Yes	No
image/cgm	Yes	No
image/gif	Yes	No
image/ief	Yes	No
image/jp2	Yes	No

Format	Transformable to:	Transformable from:
image/jpeg	Yes	No
image/png	Yes	No
image/tiff	Yes	No
image/vnd.adobe.photoshop	Yes	No
image/vnd.adobe.premiere	Yes	No
image/x-cmu-raster	Yes	No
image/x-dwt	Yes	No
image/x-portable-anymap	Yes	No
image/x-portable-bitmap	Yes	No
image/x-portable-graymap	Yes	No
image/x-portable-pixmap	Yes	No
image/x-raw-adobe	Yes	No
image/x-raw-canon	Yes	No
image/x-raw-fuji	Yes	No
image/x-raw-hasselblad	Yes	No
image/x-raw-kodak	Yes	No
image/x-raw-leica	Yes	No
image/x-raw-minolta	Yes	No
image/x-raw-nikon	Yes	No
image/x-raw-olympus	Yes	No
image/x-raw-panasonic	Yes	No
image/x-raw-pentax	Yes	No
image/x-raw-red	Yes	No
image/x-raw-sigma	Yes	No
image/x-raw-sony	Yes	No
image/x-xbitmap	Yes	No
image/x-xpixmap	Yes	No
image/x-xwindowdump	Yes	No
text/html	Yes	No
text/plain	Yes	No
text/xml	Yes	No

application/vnd.sun.xml.impress.template - sti

Format	Transformable to:	Transformable from:
application/eps	Yes	No
application/pdf	Yes	No
application/vnd.ms-powerpoint	Yes	Yes
application/vnd.ms-powerpoint.addin.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.presentation.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.slide.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.template.macroenabled.12	No	Yes
application/vnd.oasis.opendocument.graphics	Yes	Yes
application/vnd.oasis.opendocument.presentation	Yes	Yes
application/vnd.oasis.opendocument.presentation-template	Yes	Yes

Format	Transformable to:	Transformable from:
application/vnd.openxmlformats-officedocument.presentationml.presentation	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.slide	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.template	No	Yes
application/vnd.sun.xml.impress	Yes	Yes
application/vnd.visio	Yes	Yes
application/xhtml+xml	Yes	No
image/bmp	Yes	No
image/cgm	Yes	No
image/gif	Yes	No
image/ief	Yes	No
image/jp2	Yes	No
image/jpeg	Yes	No
image/png	Yes	No
image/tiff	Yes	No
image/vnd.adobe.photoshop	Yes	No
image/vnd.adobe.premiere	Yes	No
image/x-cmu-raster	Yes	No
image/x-dwt	Yes	No
image/x-portable-anymap	Yes	No
image/x-portable-bitmap	Yes	No
image/x-portable-graymap	Yes	No
image/x-portable-pixmap	Yes	No
image/x-raw-adobe	Yes	No
image/x-raw-canon	Yes	No
image/x-raw-fuji	Yes	No
image/x-raw-hasselblad	Yes	No
image/x-raw-kodak	Yes	No
image/x-raw-leica	Yes	No
image/x-raw-minolta	Yes	No
image/x-raw-nikon	Yes	No
image/x-raw-olympus	Yes	No
image/x-raw-panasonic	Yes	No
image/x-raw-pentax	Yes	No
image/x-raw-red	Yes	No
image/x-raw-sigma	Yes	No
image/x-raw-sony	Yes	No
image/x-bitmap	Yes	No
image/x-xpixmap	Yes	No
image/x-xwindowdump	Yes	No
text/html	Yes	No
text/plain	Yes	No
text/xml	Yes	No

application/vnd.sun.xml.writer - sxw

Format	Transformable to:	Transformable from:
application/eps	Yes	No
application/msword	Yes	Yes
application/pdf	Yes	No
application/rtf	Yes	Yes
application/vnd.ms-outlook	No	Yes
application/vnd.ms-word.document.macroenabled.12	No	Yes
application/vnd.ms-word.template.macroenabled.12	No	Yes
application/vnd.oasis.opendocument.text	Yes	Yes
application/vnd.oasis.opendocument.text-template	Yes	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.document	No	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.template	No	Yes
application/vnd.sun.xml.writer.template	Yes	Yes
application/wordperfect	No	Yes
application/xhtml+xml	Yes	No
image/bmp	Yes	No
image/cgm	Yes	No
image/gif	Yes	No
image/ief	Yes	No
image/jp2	Yes	No
image/jpeg	Yes	No
image/png	Yes	No
image/tiff	Yes	No
image/vnd.adobe.photoshop	Yes	No
image/vnd.adobe.premiere	Yes	No
image/x-cmu-raster	Yes	No
image/x-dwt	Yes	No
image/x-portable-anymap	Yes	No
image/x-portable-bitmap	Yes	No
image/x-portable-graymap	Yes	No
image/x-portable-pixmap	Yes	No
image/x-raw-adobe	Yes	No
image/x-raw-canon	Yes	No
image/x-raw-fuji	Yes	No
image/x-raw-hasselblad	Yes	No
image/x-raw-kodak	Yes	No
image/x-raw-leica	Yes	No
image/x-raw-minolta	Yes	No
image/x-raw-nikon	Yes	No
image/x-raw-olympus	Yes	No
image/x-raw-panasonic	Yes	No
image/x-raw-pentax	Yes	No
image/x-raw-red	Yes	No
image/x-raw-sigma	Yes	No
image/x-raw-sony	Yes	No
image/x-xbitmap	Yes	No

Format	Transformable to:	Transformable from:
image/x-xpixmap	Yes	No
image/x-xwindowdump	Yes	No
text/html	Yes	Yes
text/plain	Yes	Yes
text/xml	Yes	No

application/vnd.sun.xml.writer.template - stw

Format	Transformable to:	Transformable from:
application/eps	Yes	No
application/msword	Yes	Yes
application/pdf	Yes	No
application/rtf	Yes	Yes
application/vnd.ms-outlook	No	Yes
application/vnd.ms-word.document.macroenabled.12	No	Yes
application/vnd.ms-word.template.macroenabled.12	No	Yes
application/vnd.oasis.opendocument.text	Yes	Yes
application/vnd.oasis.opendocument.text-template	Yes	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.document	No	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.template	No	Yes
application/vnd.sun.xml.writer	Yes	Yes
application/wordperfect	No	Yes
application/xhtml+xml	Yes	No
image/bmp	Yes	No
image/cgm	Yes	No
image/gif	Yes	No
image/ief	Yes	No
image/jp2	Yes	No
image/jpeg	Yes	No
image/png	Yes	No
image/tiff	Yes	No
image/vnd.adobe.photoshop	Yes	No
image/vnd.adobe.premiere	Yes	No
image/x-cmu-raster	Yes	No
image/x-dwt	Yes	No
image/x-portable-anymap	Yes	No
image/x-portable-bitmap	Yes	No
image/x-portable-graymap	Yes	No
image/x-portable-pixmap	Yes	No
image/x-raw-adobe	Yes	No
image/x-raw-canon	Yes	No
image/x-raw-fuji	Yes	No
image/x-raw-hasselblad	Yes	No
image/x-raw-kodak	Yes	No
image/x-raw-leica	Yes	No
image/x-raw-minolta	Yes	No

Format	Transformable to:	Transformable from:
image/x-raw-nikon	Yes	No
image/x-raw-olympus	Yes	No
image/x-raw-panasonic	Yes	No
image/x-raw-pentax	Yes	No
image/x-raw-red	Yes	No
image/x-raw-sigma	Yes	No
image/x-raw-sony	Yes	No
image/x-bitmap	Yes	No
image/x-pixmap	Yes	No
image/x-xwindowdump	Yes	No
text/html	Yes	Yes
text/plain	Yes	Yes
text/xml	Yes	No

application/vnd.visio - vsd

Format	Transformable to:	Transformable from:
application/eps	Yes	No
application/pdf	Yes	No
application/vnd.ms-powerpoint	Yes	Yes
application/vnd.ms-powerpoint.addin.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.presentation.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.slide.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.template.macroenabled.12	No	Yes
application/vnd.oasis.opendocument.graphics	Yes	Yes
application/vnd.oasis.opendocument.presentation	Yes	Yes
application/vnd.oasis.opendocument.presentation-template	Yes	Yes
application/vnd.openxmlformats-officedocument.presentationml.presentation	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.slide	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.template	No	Yes
application/vnd.sun.xml.impress	Yes	Yes
application/vnd.sun.xml.impress.template	Yes	Yes
application/xhtml+xml	Yes	No
image/bmp	Yes	No
image/cgm	Yes	No
image/gif	Yes	No
image/ief	Yes	No
image/jp2	Yes	No
image/jpeg	Yes	No
image/png	Yes	No
image/tiff	Yes	No
image/vnd.adobe.photoshop	Yes	No
image/vnd.adobe.premiere	Yes	No
image/x-cmu-raster	Yes	No

Format	Transformable to:	Transformable from:
image/x-dwt	Yes	No
image/x-portable-anymap	Yes	No
image/x-portable-bitmap	Yes	No
image/x-portable-graymap	Yes	No
image/x-portable-pixmap	Yes	No
image/x-raw-adobe	Yes	No
image/x-raw-canon	Yes	No
image/x-raw-fuji	Yes	No
image/x-raw-hasselblad	Yes	No
image/x-raw-kodak	Yes	No
image/x-raw-leica	Yes	No
image/x-raw-minolta	Yes	No
image/x-raw-nikon	Yes	No
image/x-raw-olympus	Yes	No
image/x-raw-panasonic	Yes	No
image/x-raw-pentax	Yes	No
image/x-raw-red	Yes	No
image/x-raw-sigma	Yes	No
image/x-raw-sony	Yes	No
image/x-xbitmap	Yes	No
image/x-xpixmap	Yes	No
image/x-xwindowdump	Yes	No
text/html	Yes	No
text/plain	Yes	No
text/xml	Yes	No

application/wordperfect - wpd



This format cannot be generated from any other format.

Format	Transformable to:
application/eps	Yes
application/msword	Yes
application/pdf	Yes
application/rtf	Yes
application/vnd.oasis.opendocument.text	Yes
application/vnd.oasis.opendocument.text-template	Yes
application/vnd.sun.xml.writer	Yes
application/vnd.sun.xml.writer.template	Yes
application/xhtml+xml	Yes
image/bmp	Yes
image/cgm	Yes
image/gif	Yes
image/ief	Yes
image/jp2	Yes
image/jpeg	Yes
image/png	Yes

Format	Transformable to:
image/tiff	Yes
image/vnd.adobe.photoshop	Yes
image/vnd.adobe.premiere	Yes
image/x-cmu-raster	Yes
image/x-dwt	Yes
image/x-portable-anymap	Yes
image/x-portable-bitmap	Yes
image/x-portable-graymap	Yes
image/x-portable-pixmap	Yes
image/x-raw-adobe	Yes
image/x-raw-canon	Yes
image/x-raw-fuji	Yes
image/x-raw-hasselblad	Yes
image/x-raw-kodak	Yes
image/x-raw-leica	Yes
image/x-raw-minolta	Yes
image/x-raw-nikon	Yes
image/x-raw-olympus	Yes
image/x-raw-panasonic	Yes
image/x-raw-pentax	Yes
image/x-raw-red	Yes
image/x-raw-sigma	Yes
image/x-raw-sony	Yes
image/x-xbitmap	Yes
image/x-xpixmap	Yes
image/x-xwindowdump	Yes
text/html	Yes
text/plain	Yes
text/xml	Yes

application/x-bcpio, application/x-compress, application/x-csh, application/x-dosexec, application/x-dvi, application/x-fla, application/x-gtar, application/x-indesign, application/x-latex, application/x-mif, application/x-rar-compressed, application/x-sh, application/x-shar, application/x-shockwave-flash, 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, exe, dvi, fla, gtar, indd, latex, mif, rar, sh, shar, swf, sv4cpio, sv4crc, tcl, tex, texinfo, tr, man, me, ms, ustar, src, cer, fpx

These formats cannot be transformed into, or generated from, any other format.

application/x-cpio, application/x-gzip, application/x-hdf, application/x-netcdf, application/x-tar, application/zip and text/x-java-source - cpio, gzip, hdf, cdf, tar, zip, java

 These formats cannot be generated from any other format.

Format	Transformable to:
application/xhtml+xml	Yes
text/html	Yes
text/plain	Yes

Format	Transformable to:				
text/xml	Yes				
application/x-javascript - js					
 This format cannot be generated from any other format.					
<table border="1"> <thead> <tr> <th>Format</th><th>Transformable to:</th></tr> </thead> <tbody> <tr> <td>text/plain</td><td>Yes</td></tr> </tbody> </table>		Format	Transformable to:	text/plain	Yes
Format	Transformable to:				
text/plain	Yes				
application/xhtml+xml - xhtml					
Format	Transformable to:	Transformable from:			
application/java-archive	No	Yes			
application/msword	No	Yes			
application/ogg	No	Yes			
application/pdf	No	Yes			
application/rss+xml	No	Yes			
application/rtf	No	Yes			
application/vnd.apple.keynote	No	Yes			
application/vnd.apple.numbers	No	Yes			
application/vnd.apple.pages	No	Yes			
application/vnd.ms-excel	No	Yes			
application/vnd.ms-excel.addin.macroenabled.12	No	Yes			
application/vnd.ms-excel.sheet.binary.macroenabled.12	No	Yes			
application/vnd.ms-excel.sheet.macroenabled.12	No	Yes			
application/vnd.ms-excel.template.macroenabled.12	No	Yes			
application/vnd.ms-outlook	No	Yes			
application/vnd.ms-powerpoint	No	Yes			
application/vnd.ms-powerpoint.addin.macroenabled.12	No	Yes			
application/vnd.ms-powerpoint.presentation.macroenabled.12	No	Yes			
application/vnd.ms-powerpoint.slide.macroenabled.12	No	Yes			
application/vnd.ms-powerpoint.slideshow.macroenabled.12	No	Yes			
application/vnd.ms-powerpoint.template.macroenabled.12	No	Yes			
application/vnd.ms-project	No	Yes			
application/vnd.ms-word.document.macroenabled.12	No	Yes			
application/vnd.ms-word.template.macroenabled.12	No	Yes			
application/vnd.oasis.opendocument.chart	No	Yes			
application/vnd.oasis.opendocument.graphics	No	Yes			
application/vnd.oasis.opendocument.image	No	Yes			
application/vnd.oasis.opendocument.presentation	No	Yes			
application/vnd.oasis.opendocument.presentation-template	No	Yes			
application/vnd.oasis.opendocument.spreadsheet	No	Yes			
application/vnd.oasis.opendocument.spreadsheet-template	No	Yes			
application/vnd.oasis.opendocument.text	No	Yes			
application/vnd.oasis.opendocument.text-master	No	Yes			
application/vnd.oasis.opendocument.text-template	No	Yes			
application/vnd.oasis.opendocument.text-web	No	Yes			
application/vnd.openxmlformats-officedocument.presentationml.presentation	No	Yes			

Format	Transformable to:	Transformable from:
application/vnd.openxmlformats-officedocument.presentationml.slide	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.slideshow	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.template	No	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	No	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.template	No	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.document	No	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.template	No	Yes
application/vnd.sun.xml.calc	No	Yes
application/vnd.sun.xml.calc.template	No	Yes
application/vnd.sun.xml.impress	No	Yes
application/vnd.sun.xml.impress.template	No	Yes
application/vnd.sun.xml.writer	No	Yes
application/vnd.sun.xml.writer.template	No	Yes
application/vnd.visio	No	Yes
application/wordperfect	No	Yes
application/x-cpio	No	Yes
application/x-gzip	No	Yes
application/x-hdf	No	Yes
application/x-netcdf	No	Yes
application/x-tar	No	Yes
application/zip	No	Yes
text/html	Yes	Yes
text/plain	Yes	Yes
text/x-java-source	No	Yes
text/xml	Yes	Yes

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	Yes	Yes
application/illustrator	No	Yes
application/msword	No	Yes
application/pdf	Yes	Yes
application/rtf	No	Yes
application/vnd.apple.keynote	No	Yes
application/vnd.apple.numbers	No	Yes

Format	Transformable to:	Transformable from:
application/vnd.apple.pages	No	Yes
application/vnd.ms-excel	No	Yes
application/vnd.ms-excel.addin.macroenabled.12	No	Yes
application/vnd.ms-excel.sheet.binary.macroenabled.12	No	Yes
application/vnd.ms-excel.sheet.macroenabled.12	No	Yes
application/vnd.ms-excel.template.macroenabled.12	No	Yes
application/vnd.ms-outlook	No	Yes
application/vnd.ms-powerpoint	No	Yes
application/vnd.ms-powerpoint.addin.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.presentation.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.slide.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.slideshow.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.template.macroenabled.12	No	Yes
application/vnd.ms-word.document.macroenabled.12	No	Yes
application/vnd.ms-word.template.macroenabled.12	No	Yes
application/vnd.oasis.opendocument.presentation	No	Yes
application/vnd.oasis.opendocument.presentation-template	No	Yes
application/vnd.oasis.opendocument.spreadsheet	No	Yes
application/vnd.oasis.opendocument.spreadsheet-template	No	Yes
application/vnd.oasis.opendocument.text	No	Yes
application/vnd.oasis.opendocument.text-template	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.presentation	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.slide	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.slideshow	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.template	No	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	No	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.template	No	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.document	No	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.template	No	Yes
application/vnd.sun.xml.calc	No	Yes
application/vnd.sun.xml.calc.template	No	Yes
application/vnd.sun.xml.impress	No	Yes
application/vnd.sun.xml.impress.template	No	Yes
application/vnd.sun.xml.writer	No	Yes
application/vnd.sun.xml.writer.template	No	Yes
application/vnd.visio	No	Yes
application/wordperfect	No	Yes
image/bmp	Yes	No
image/cgm	Yes	Yes

Format	Transformable to:	Transformable from:
image/gif	Yes	Yes
image/ief	Yes	Yes
image/jp2	Yes	Yes
image/jpeg	Yes	Yes
image/png	Yes	Yes
image/tiff	Yes	Yes
image/vnd.adobe.photoshop	Yes	Yes
image/vnd.adobe.premiere	Yes	Yes
image/x-cmu-raster	Yes	Yes
image/x-dwt	Yes	Yes
image/x-portable-anymap	Yes	Yes
image/x-portable-bitmap	Yes	Yes
image/x-portable-graymap	Yes	Yes
image/x-portable-pixmap	Yes	Yes
image/x-raw-adobe	Yes	Yes
image/x-raw-canon	Yes	Yes
image/x-raw-fuji	Yes	Yes
image/x-raw-hasselblad	Yes	Yes
image/x-raw-kodak	Yes	Yes
image/x-raw-leica	Yes	Yes
image/x-raw-minolta	Yes	Yes
image/x-raw-nikon	Yes	Yes
image/x-raw-olympus	Yes	Yes
image/x-raw-panasonic	Yes	Yes
image/x-raw-pentax	Yes	Yes
image/x-raw-red	Yes	Yes
image/x-raw-sigma	Yes	Yes
image/x-raw-sony	Yes	Yes
image/x-bitmap	Yes	Yes
image/x-pixmap	Yes	Yes
image/x-xwindowdump	Yes	Yes
text/csv	No	Yes
text/html	No	Yes
text/plain	No	Yes
text/xml	No	Yes

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	Yes	Yes

Format	Transformable to:	Transformable from:
application/illustrator	No	Yes
application/msword	No	Yes
application/pdf	No	Yes
application/rtf	No	Yes
application/vnd.apple.keynote	No	Yes
application/vnd.apple.numbers	No	Yes
application/vnd.apple.pages	No	Yes
application/vnd.ms-excel	No	Yes
application/vnd.ms-excel.addin.macroenabled.12	No	Yes
application/vnd.ms-excel.sheet.binary.macroenabled.12	No	Yes
application/vnd.ms-excel.sheet.macroenabled.12	No	Yes
application/vnd.ms-excel.template.macroenabled.12	No	Yes
application/vnd.ms-outlook	No	Yes
application/vnd.ms-powerpoint	No	Yes
application/vnd.ms-powerpoint.addin.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.presentation.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.slide.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.slideshow.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.template.macroenabled.12	No	Yes
application/vnd.ms-word.document.macroenabled.12	No	Yes
application/vnd.ms-word.template.macroenabled.12	No	Yes
application/vnd.oasis.opendocument.graphics	No	Yes
application/vnd.oasis.opendocument.presentation	No	Yes
application/vnd.oasis.opendocument.presentation-template	No	Yes
application/vnd.oasis.opendocument.spreadsheet	No	Yes
application/vnd.oasis.opendocument.spreadsheet-template	No	Yes
application/vnd.oasis.opendocument.text	No	Yes
application/vnd.oasis.opendocument.text-template	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.presentation	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.slide	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.slideshow	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.template	No	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	No	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.template	No	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.document	No	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.template	No	Yes
application/vnd.sun.xml.calc	No	Yes
application/vnd.sun.xml.calc.template	No	Yes
application/vnd.sun.xml.impress	No	Yes

Format	Transformable to:	Transformable from:
application/vnd.sun.xml.impress.template	No	Yes
application/vnd.sun.xml.writer	No	Yes
application/vnd.sun.xml.writer.template	No	Yes
application/vnd.visio	No	Yes
application/wordperfect	No	Yes
image/bmp	Yes	Yes
image/cgm	Yes	Yes
image/gif	Yes	Yes
image/ief	Yes	Yes
image/jp2	Yes	Yes
image/jpeg	Yes	Yes
image/png	Yes	Yes
image/tiff	Yes	Yes
image/vnd.adobe.photoshop	Yes	Yes
image/vnd.adobe.premiere	Yes	Yes
image/x-cmu-raster	Yes	Yes
image/x-dwt	Yes	Yes
image/x-portable-anymap	Yes	Yes
image/x-portable-bitmap	Yes	Yes
image/x-portable-graymap	Yes	Yes
image/x-portable-pixmap	Yes	Yes
image/x-raw-adobe	Yes	Yes
image/x-raw-canon	Yes	Yes
image/x-raw-fuji	Yes	Yes
image/x-raw-hasselblad	Yes	Yes
image/x-raw-kodak	Yes	Yes
image/x-raw-leica	Yes	Yes
image/x-raw-minolta	Yes	Yes
image/x-raw-nikon	Yes	Yes
image/x-raw-olympus	Yes	Yes
image/x-raw-panasonic	Yes	Yes
image/x-raw-pentax	Yes	Yes
image/x-raw-red	Yes	Yes
image/x-raw-sigma	Yes	Yes
image/x-raw-sony	Yes	Yes
image/x-xbitmap	Yes	Yes
image/x-xpixmap	Yes	Yes
image/x-xwindowdump	Yes	Yes
text/csv	No	Yes
text/html	No	Yes
text/plain	No	Yes
text/xml	No	Yes

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-bitmap, image/x-xpixmap and image/x-xwindowdump - dng, cr2, raf, 3fr, k25, rwl, 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	Yes	Yes
application/illustrator	No	Yes
application/msword	No	Yes
application/pdf	No	Yes
application/rtf	No	Yes
application/vnd.apple.keynote	No	Yes
application/vnd.apple.numbers	No	Yes
application/vnd.apple.pages	No	Yes
application/vnd.ms-excel	No	Yes
application/vnd.ms-excel.addin.macroenabled.12	No	Yes
application/vnd.ms-excel.sheet.binary.macroenabled.12	No	Yes
application/vnd.ms-excel.sheet.macroenabled.12	No	Yes
application/vnd.ms-excel.template.macroenabled.12	No	Yes
application/vnd.ms-outlook	No	Yes
application/vnd.ms-powerpoint	No	Yes
application/vnd.ms-powerpoint.addin.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.presentation.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.slide.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.slideshow.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.template.macroenabled.12	No	Yes
application/vnd.ms-word.document.macroenabled.12	No	Yes
application/vnd.ms-word.template.macroenabled.12	No	Yes
application/vnd.oasis.opendocument.graphics	No	Yes
application/vnd.oasis.opendocument.presentation	No	Yes
application/vnd.oasis.opendocument.presentation-template	No	Yes
application/vnd.oasis.opendocument.spreadsheet	No	Yes
application/vnd.oasis.opendocument.spreadsheet-template	No	Yes
application/vnd.oasis.opendocument.text	No	Yes
application/vnd.oasis.opendocument.text-template	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.presentation	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.slide	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.slideshow	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.template	No	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	No	Yes

Format	Transformable to:	Transformable from:
application/vnd.openxmlformats-officedocument.spreadsheetml.template	No	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.document	No	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.template	No	Yes
application/vnd.sun.xml.calc	No	Yes
application/vnd.sun.xml.calc.template	No	Yes
application/vnd.sun.xml.impress	No	Yes
application/vnd.sun.xml.impress.template	No	Yes
application/vnd.sun.xml.writer	No	Yes
application/vnd.sun.xml.writer.template	No	Yes
application/vnd.visio	No	Yes
application/wordperfect	No	Yes
image/bmp	Yes	Yes
image/cgm	Yes	Yes
image/gif	Yes	Yes
image/ief	Yes	Yes
image/jp2	Yes	Yes
image/jpeg	Yes	Yes
image/png	Yes	Yes
image/tiff	Yes	Yes
image/vnd.adobe.photoshop	Yes	Yes
image/vnd.adobe.premiere	Yes	Yes
image/x-cmu-raster	Yes	Yes
image/x-dwt	Yes	Yes
image/x-portable-anymap	Yes	Yes
image/x-portable-bitmap	Yes	Yes
image/x-portable-graymap	Yes	Yes
image/x-portable-pixmap	Yes	Yes
image/x-raw-adobe	Yes	Yes
image/x-raw-canon	Yes	Yes
image/x-raw-fuji	Yes	Yes
image/x-raw-hasselblad	Yes	Yes
image/x-raw-kodak	Yes	Yes
image/x-raw-leica	Yes	Yes
image/x-raw-minolta	Yes	Yes
image/x-raw-nikon	Yes	Yes
image/x-raw-olympus	Yes	Yes
image/x-raw-panasonic	Yes	Yes
image/x-raw-pentax	Yes	Yes
image/x-raw-red	Yes	Yes
image/x-raw-sigma	Yes	Yes
image/x-raw-sony	Yes	Yes
image/x-bitmap	Yes	Yes
image/x-xpixmap	Yes	Yes

Format	Transformable to:	Transformable from:
image/x-xwindowdump	Yes	Yes
text/csv	No	Yes
text/html	No	Yes
text/plain	No	Yes
text/xml	No	Yes

message/rfc822, text/calendar, text/css, text/richtext, text/sgml, text/tab-separated-values, text/x-markdown, text/x-setext and text/x-jsp - eml, ics, css, rtx, sgml, tsv, md, etx, jpg

 These formats cannot be generated from any other format.

Format	Transformable to:
text/plain	Yes

text/csv - csv

Format	Transformable to:	Transformable from:
application/eps	Yes	No
application/pdf	Yes	No
application/vnd.ms-excel	No	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	No	Yes
image/bmp	Yes	No
image/cgm	Yes	No
image/gif	Yes	No
image/ief	Yes	No
image/jp2	Yes	No
image/jpeg	Yes	No
image/png	Yes	No
image/tiff	Yes	No
image/vnd.adobe.photoshop	Yes	No
image/vnd.adobe.premiere	Yes	No
image/x-cmu-raster	Yes	No
image/x-dwt	Yes	No
image/x-portable-anymap	Yes	No
image/x-portable-bitmap	Yes	No
image/x-portable-graymap	Yes	No
image/x-portable-pixmap	Yes	No
image/x-raw-adobe	Yes	No
image/x-raw-canon	Yes	No
image/x-raw-fuji	Yes	No
image/x-raw-hasselblad	Yes	No
image/x-raw-kodak	Yes	No
image/x-raw-leica	Yes	No
image/x-raw-minolta	Yes	No
image/x-raw-nikon	Yes	No
image/x-raw-olympus	Yes	No
image/x-raw-panasonic	Yes	No
image/x-raw-pentax	Yes	No

Format	Transformable to:	Transformable from:
image/x-raw-red	Yes	No
image/x-raw-sigma	Yes	No
image/x-raw-sony	Yes	No
image/x-xbitmap	Yes	No
image/x-xpixmap	Yes	No
image/x-xwindowdump	Yes	No
text/plain	Yes	No

text/html - html

Format	Transformable to:	Transformable from:
application/eps	Yes	No
application/java-archive	No	Yes
application/msword	Yes	Yes
application/ogg	No	Yes
application/pdf	Yes	Yes
application/rss+xml	No	Yes
application/rtf	Yes	Yes
application/vnd.apple.keynote	No	Yes
application/vnd.apple.numbers	No	Yes
application/vnd.apple.pages	No	Yes
application/vnd.ms-excel	No	Yes
application/vnd.ms-excel.addin.macroenabled.12	No	Yes
application/vnd.ms-excel.sheet.binary.macroenabled.12	No	Yes
application/vnd.ms-excel.sheet.macroenabled.12	No	Yes
application/vnd.ms-excel.template.macroenabled.12	No	Yes
application/vnd.ms-outlook	No	Yes
application/vnd.ms-powerpoint	No	Yes
application/vnd.ms-powerpoint.addin.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.presentation.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.slide.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.slideshow.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.template.macroenabled.12	No	Yes
application/vnd.ms-project	No	Yes
application/vnd.ms-word.document.macroenabled.12	No	Yes
application/vnd.ms-word.template.macroenabled.12	No	Yes
application/vnd.oasis.opendocument.chart	No	Yes
application/vnd.oasis.opendocument.graphics	No	Yes
application/vnd.oasis.opendocument.image	No	Yes
application/vnd.oasis.opendocument.presentation	No	Yes
application/vnd.oasis.opendocument.presentation-template	No	Yes
application/vnd.oasis.opendocument.spreadsheet	No	Yes
application/vnd.oasis.opendocument.spreadsheet-template	No	Yes
application/vnd.oasis.opendocument.text	Yes	Yes
application/vnd.oasis.opendocument.text-master	No	Yes
application/vnd.oasis.opendocument.text-template	Yes	Yes
application/vnd.oasis.opendocument.text-web	No	Yes

Format	Transformable to:	Transformable from:
application/vnd.openxmlformats-officedocument.presentationml.presentation	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.slide	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.slideshow	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.template	No	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	No	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.template	No	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.document	No	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.template	No	Yes
application/vnd.sun.xml.calc	No	Yes
application/vnd.sun.xml.calc.template	No	Yes
application/vnd.sun.xml.impress	No	Yes
application/vnd.sun.xml.impress.template	No	Yes
application/vnd.sun.xml.writer	Yes	Yes
application/vnd.sun.xml.writer.template	Yes	Yes
application/vnd.visio	No	Yes
application/wordperfect	No	Yes
application/x-cpio	No	Yes
application/x-gzip	No	Yes
application/x-hdf	No	Yes
application/x-netcdf	No	Yes
application/x-tar	No	Yes
application/xhtml+xml	Yes	Yes
application/zip	No	Yes
image/bmp	Yes	No
image/cgm	Yes	No
image/gif	Yes	No
image/ief	Yes	No
image/jp2	Yes	No
image/jpeg	Yes	No
image/png	Yes	No
image/tiff	Yes	No
image/vnd.adobe.photoshop	Yes	No
image/vnd.adobe.premiere	Yes	No
image/x-cmu-raster	Yes	No
image/x-dwt	Yes	No
image/x-portable-anymap	Yes	No
image/x-portable-bitmap	Yes	No
image/x-portable-graymap	Yes	No
image/x-portable-pixmap	Yes	No

Format	Transformable to:	Transformable from:
image/x-raw-adobe	Yes	No
image/x-raw-canon	Yes	No
image/x-raw-fuji	Yes	No
image/x-raw-hasselblad	Yes	No
image/x-raw-kodak	Yes	No
image/x-raw-leica	Yes	No
image/x-raw-minolta	Yes	No
image/x-raw-nikon	Yes	No
image/x-raw-olympus	Yes	No
image/x-raw-panasonic	Yes	No
image/x-raw-pentax	Yes	No
image/x-raw-red	Yes	No
image/x-raw-sigma	Yes	No
image/x-raw-sony	Yes	No
image/x-bitmap	Yes	No
image/x-xpixmap	Yes	No
image/x-xwindowdump	Yes	No
text/mediawiki	No	Yes
text/plain	Yes	Yes
text/x-java-source	No	Yes
text/xml	Yes	Yes

text/mediawiki - mw

 This format cannot be generated from any other format.

Format	Transformable to:
text/html	Yes
text/plain	Yes

text/plain - txt

Format	Transformable to:	Transformable from:
application/eps	Yes	No
application/java-archive	No	Yes
application/msword	Yes	Yes
application/ogg	No	Yes
application/pdf	Yes	Yes
application/rss+xml	No	Yes
application/rtf	Yes	Yes
application/vnd.apple.keynote	No	Yes
application/vnd.apple.numbers	No	Yes
application/vnd.apple.pages	No	Yes
application/vnd.ms-excel	No	Yes
application/vnd.ms-excel.addin.macroenabled.12	No	Yes
application/vnd.ms-excel.sheet.binary.macroenabled.12	No	Yes
application/vnd.ms-excel.sheet.macroenabled.12	No	Yes
application/vnd.ms-excel.template.macroenabled.12	No	Yes

Format	Transformable to:	Transformable from:
application/vnd.ms-outlook	No	Yes
application/vnd.ms-powerpoint	No	Yes
application/vnd.ms-powerpoint.addin.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.presentation.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.slideshow.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.template.macroenabled.12	No	Yes
application/vnd.ms-project	No	Yes
application/vnd.ms-word.document.macroenabled.12	No	Yes
application/vnd.ms-word.template.macroenabled.12	No	Yes
application/vnd.oasis.opendocument.chart	No	Yes
application/vnd.oasis.opendocument.graphics	No	Yes
application/vnd.oasis.opendocument.image	No	Yes
application/vnd.oasis.opendocument.presentation	No	Yes
application/vnd.oasis.opendocument.presentation-template	No	Yes
application/vnd.oasis.opendocument.spreadsheet	No	Yes
application/vnd.oasis.opendocument.spreadsheet-template	No	Yes
application/vnd.oasis.opendocument.text	Yes	Yes
application/vnd.oasis.opendocument.text-master	No	Yes
application/vnd.oasis.opendocument.text-template	Yes	Yes
application/vnd.oasis.opendocument.text-web	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.presentation	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.slideshow	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.template	No	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	No	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.template	No	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.document	No	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.template	No	Yes
application/vnd.sun.xml.calc	No	Yes
application/vnd.sun.xml.calc.template	No	Yes
application/vnd.sun.xml.impress	No	Yes
application/vnd.sun.xml.impress.template	No	Yes
application/vnd.sun.xml.writer	Yes	Yes
application/vnd.sun.xml.writer.template	Yes	Yes
application/vnd.visio	No	Yes
application/wordperfect	No	Yes
application/x-cpio	No	Yes
application/x-gzip	No	Yes
application/x-hdf	No	Yes
application/x-javascript	No	Yes
application/x-netcdf	No	Yes

Format	Transformable to:	Transformable from:
application/x-tar	No	Yes
application/xhtml+xml	Yes	Yes
application/zip	No	Yes
image/bmp	Yes	No
image/cgm	Yes	No
image/gif	Yes	No
image/ief	Yes	No
image/jp2	Yes	No
image/jpeg	Yes	No
image/png	Yes	No
image/tiff	Yes	No
image/vnd.adobe.photoshop	Yes	No
image/vnd.adobe.premiere	Yes	No
image/x-cmu-raster	Yes	No
image/x-dwt	Yes	No
image/x-portable-anymap	Yes	No
image/x-portable-bitmap	Yes	No
image/x-portable-graymap	Yes	No
image/x-portable-pixmap	Yes	No
image/x-raw-adobe	Yes	No
image/x-raw-canon	Yes	No
image/x-raw-fuji	Yes	No
image/x-raw-hasselblad	Yes	No
image/x-raw-kodak	Yes	No
image/x-raw-leica	Yes	No
image/x-raw-minolta	Yes	No
image/x-raw-nikon	Yes	No
image/x-raw-olympus	Yes	No
image/x-raw-panasonic	Yes	No
image/x-raw-pentax	Yes	No
image/x-raw-red	Yes	No
image/x-raw-sigma	Yes	No
image/x-raw-sony	Yes	No
image/x-xbitmap	Yes	No
image/x-xpixmap	Yes	No
image/x-xwindowdump	Yes	No
message/rfc822	No	Yes
text/calendar	No	Yes
text/css	No	Yes
text/csv	No	Yes
text/html	Yes	Yes
text/mediawiki	No	Yes
text/richtext	No	Yes
text/sgml	No	Yes
text/tab-separated-values	No	Yes
text/x-java-source	No	Yes

Format	Transformable to:	Transformable from:
text/x-jsp	No	Yes
text/x-markdown	No	Yes
text/x-setext	No	Yes
text/xml	Yes	Yes

text/xml - xml

Format	Transformable to:	Transformable from:
application/eps	Yes	No
application/java-archive	No	Yes
application/msword	No	Yes
application/ogg	No	Yes
application/pdf	Yes	Yes
application/rss+xml	No	Yes
application/rtf	No	Yes
application/vnd.apple.keynote	No	Yes
application/vnd.apple.numbers	No	Yes
application/vnd.apple.pages	No	Yes
application/vnd.ms-excel	No	Yes
application/vnd.ms-excel.addin.macroenabled.12	No	Yes
application/vnd.ms-excel.sheet.binary.macroenabled.12	No	Yes
application/vnd.ms-excel.sheet.macroenabled.12	No	Yes
application/vnd.ms-excel.template.macroenabled.12	No	Yes
application/vnd.ms-outlook	No	Yes
application/vnd.ms-powerpoint	No	Yes
application/vnd.ms-powerpoint.addin.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.presentation.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.slide.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.slideshow.macroenabled.12	No	Yes
application/vnd.ms-powerpoint.template.macroenabled.12	No	Yes
application/vnd.ms-project	No	Yes
application/vnd.ms-word.document.macroenabled.12	No	Yes
application/vnd.ms-word.template.macroenabled.12	No	Yes
application/vnd.oasis.opendocument.chart	No	Yes
application/vnd.oasis.opendocument.graphics		
application/vnd.oasis.opendocument.image	No	Yes
application/vnd.oasis.opendocument.presentation	No	Yes
application/vnd.oasis.opendocument.presentation-template	No	Yes
application/vnd.oasis.opendocument.spreadsheet	No	Yes
application/vnd.oasis.opendocument.spreadsheet-template	No	Yes
application/vnd.oasis.opendocument.text	No	Yes
application/vnd.oasis.opendocument.text-master	No	Yes
application/vnd.oasis.opendocument.text-template	No	Yes
application/vnd.oasis.opendocument.text-web	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.presentation	No	Yes

Format	Transformable to:	Transformable from:
application/vnd.openxmlformats-officedocument.presentationml.slide	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.slideshow	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.template	No	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	No	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.template	No	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.document	No	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.template	No	Yes
application/vnd.sun.xml.calc	No	Yes
application/vnd.sun.xml.calc.template	No	Yes
application/vnd.sun.xml.impress	No	Yes
application/vnd.sun.xml.impress.template	No	Yes
application/vnd.sun.xml.writer	No	Yes
application/vnd.sun.xml.writer.template	No	Yes
application/vnd.visio	No	Yes
application/wordperfect	No	Yes
application/x-cpio	No	Yes
application/x-gzip	No	Yes
application/x-hdf	No	Yes
application/x-netcdf	No	Yes
application/x-tar	No	Yes
application/xhtml+xml	Yes	Yes
application/zip	No	Yes
image/bmp	Yes	No
image/cgm	Yes	No
image/gif	Yes	No
image/ief	Yes	No
image/jp2	Yes	No
image/jpeg	Yes	No
image/png	Yes	No
image/tiff	Yes	No
image/vnd.adobe.photoshop	Yes	No
image/vnd.adobe.premiere	Yes	No
image/x-cmu-raster	Yes	No
image/x-dwt	Yes	No
image/x-portable-anymap	Yes	No
image/x-portable-bitmap	Yes	No
image/x-portable-graymap	Yes	No
image/x-portable-pixmap	Yes	No
image/x-raw-adobe	Yes	No
image/x-raw-canon	Yes	No

Format	Transformable to:	Transformable from:
image/x-raw-fuji	Yes	No
image/x-raw-hasselblad	Yes	No
image/x-raw-kodak	Yes	No
image/x-raw-leica	Yes	No
image/x-raw-minolta	Yes	No
image/x-raw-nikon	Yes	No
image/x-raw-olympus	Yes	No
image/x-raw-panasonic	Yes	No
image/x-raw-pentax	Yes	No
image/x-raw-red	Yes	No
image/x-raw-sigma	Yes	No
image/x-raw-sony	Yes	No
image/x-xbitmap	Yes	No
image/x-xpixmap	Yes	No
image/x-xwindowdump	Yes	No
text/html	Yes	Yes
text/plain	Yes	Yes
text/x-java-source	No	Yes

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 Alfresco Outlook Integration or 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 507 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 two transformers (com.westernacher.wps.alfresco.transformers.mail.aspose.EML_MSG2PdfTransformer and com.westernacher.wps.alfresco.transformers.mail.aspose.EML_MSG2PngTransformer) to manipulate PDF, image and Outlook email messages. The formats listed are in addition to the standard formats as specified in [Standard transform options](#) on page 507.

application/eps - eps, application/pdf - pdf, 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-cmu-raster - ras, 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-bitmap - xbm, image/x-xpixmap - xpm, image/x-xwindowdump - xwd

Format	Transformable from:
application/vnd.ms-outlook	Yes
message/rfc822	Yes

application/vnd.ms-outlook - msg and message/rfc822 - eml

Format	Transformable to:
application/eps	Yes
application/pdf	Yes
image/bmp	Yes
image/cgm	Yes
image/gif	Yes
image/ief	Yes
image/jp2	Yes
image/jpeg	Yes
image/png	Yes
image/tiff	Yes
image/vnd.adobe.photoshop	Yes
image/vnd.adobe.premiere	Yes
image/x-cmu-raster	Yes
image/x-dwt	Yes
image/x-portable-anymap	Yes
image/x-portable-bitmap	Yes
image/x-portable-graymap	Yes
image/x-portable-pixmap	Yes
image/x-raw-adobe	Yes
image/x-raw-canon	Yes
image/x-raw-fuji	Yes
image/x-raw-hasselblad	Yes
image/x-raw-kodak	Yes
image/x-raw-leica	Yes
image/x-raw-minolta	Yes
image/x-raw-nikon	Yes
image/x-raw-olympus	Yes
image/x-raw-panasonic	Yes
image/x-raw-pentax	Yes
image/x-raw-red	Yes
image/x-raw-sigma	Yes
image/x-raw-sony	Yes
image/x-bitmap	Yes
image/x-xpixmap	Yes
image/x-xwindowdump	Yes

Transformation Server

Transformation Server gives an alternative method of remote transformation for a range of applications including pdf, 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 507.

application/pdf - pdf

Format	Transformable from:
application/vnd.ms-powerpoint.slideshow.macroenabled.12	Yes
application/vnd.openxmlformats-officedocument.presentationml.slideshow	Yes

application/vnd.ms-powerpoint.slideshow.macroenabled.12 - ppsm, application/vnd.openxmlformats-officedocument.presentationml.slideshow - ppssx

Format	Transformable to:
application/pdf	Yes
application/x-shockwave-flash	Yes

Alfresco Media Management

Alfresco Media Management provides additional options for audio, video and image transformations. The formats are listed in [Media Management transform options](#).

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.

The following table shows the file types that support these capabilities. See [Standard transform options](#) on page 507 for a full listing of formats.

Formats that support preview and thumbnail generation

Format	File type extension	Supports preview	Supports thumbnail
application/eps	eps	No	Yes
application/illustrator	ai	Yes	Yes
application/msword	doc	Yes	Yes
application/pdf	pdf	Yes	Yes
application/rtf	rtf	Yes	Yes
application/vnd.apple.keynote	key	Yes	Yes
application/vnd.apple.numbers	numbers	Yes	Yes
application/vnd.apple.pages	pages	Yes	Yes
application/vnd.ms-excel	xls	Yes	Yes
application/vnd.ms-excel.addin.macroenabled.12	xlam	No	Yes
application/vnd.ms-excel.sheet.binary.macroenabled.12	xlsb	Yes	Yes
application/vnd.ms-excel.sheet.macroenabled.12	xlsm	Yes	Yes
application/vnd.ms-excel.template.macroenabled.12	xltm	Yes	Yes
application/vnd.ms-outlook	msg	Yes	Yes
application/vnd.ms-powerpoint	ppt	Yes	Yes
application/vnd.ms-powerpoint.addin.macroenabled.12	ppam	Yes	Yes
application/vnd.ms-powerpoint.presentation.macroenabled.12	pptm	Yes	Yes

Format	File type extension	Supports preview	Supports thumbnail
application/vnd.ms-powerpoint.slide.macroenabled.12	sldm	Yes	Yes
application/vnd.ms-powerpoint.slideshow.macroenabled.12	ppsm	No	Yes
application/vnd.ms-powerpoint.template.macroenabled.12	potm	Yes	Yes
application/vnd.ms-word.document.macroenabled.12	docm	Yes	Yes
application/vnd.ms-word.template.macroenabled.12	dotm	Yes	Yes
application/vnd.oasis.opendocument.presentation	odp	Yes	Yes
application/vnd.oasis.opendocument.presentation-template	otp	Yes	Yes
application/vnd.oasis.opendocument.spreadsheet	ods	Yes	Yes
application/vnd.oasis.opendocument.spreadsheet-template	ots	Yes	Yes
application/vnd.oasis.opendocument.text	odt	Yes	Yes
application/vnd.oasis.opendocument.text-template	ott	Yes	Yes
application/vnd.openxmlformats-officedocument.presentationml.presentation	pptx	Yes	Yes
application/vnd.openxmlformats-officedocument.presentationml.slide	sldx	Yes	Yes
application/vnd.openxmlformats-officedocument.presentationml.slideshow	ppsx	No	Yes
application/vnd.openxmlformats-officedocument.presentationml.template	potx	Yes	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	xlsx	Yes	Yes
application/vnd.openxmlformats-officedocument.spreadsheetml.template	xltx	Yes	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.document	docx	Yes	Yes
application/vnd.openxmlformats-officedocument.wordprocessingml.template	dotx	Yes	Yes
application/vnd.sun.xml.calc	sxc	Yes	Yes
application/vnd.sun.xml.calc.template	stc	Yes	Yes
application/vnd.sun.xml.impress	sxi	Yes	Yes
application/vnd.sun.xml.impress.template	sti	Yes	Yes
application/vnd.sun.xml.writer	sxw	Yes	Yes
application/vnd.sun.xml.writer.template	stw	Yes	Yes
application/vnd.visio	vsd	Yes	Yes
application/wordperfect	wpd	Yes	Yes
application/x-cpio	cpio	Yes	No
application/x-tar	tar	Yes	No
application/zip	zip	Yes	No
image/bmp	bmp	No	Yes
image/cgm	cgm	No	Yes
image/gif	gif	No	Yes
image/ief	ief	No	Yes
image/jp2	jp2	No	Yes
image/jpeg	jpg	No	Yes

Format	File type extension	Supports preview	Supports thumbnail
image/png	png	No	Yes
image/tiff	tiff	Yes	Yes
image/vnd.adobe.photoshop	psd	No	Yes
image/vnd.adobe.premiere	ppj	No	Yes
image/x-cmu-raster	ras	No	Yes
image/x-dwt	dwt	No	Yes
image/x-portable-anymap	pnm	No	Yes
image/x-portable-bitmap	pbm	No	Yes
image/x-portable-graymap	pgm	No	Yes
image/x-portable-pixmap	ppm	No	Yes
image/x-raw-adobe	dng	No	Yes
image/x-raw-canon	cr2	No	Yes
image/x-raw-fuji	raf	No	Yes
image/x-raw-hasselblad	3fr	No	Yes
image/x-raw-kodak	k25	No	Yes
image/x-raw-leica	rwl	No	Yes
image/x-raw-minolta	mrw	No	Yes
image/x-raw-nikon	nef	No	Yes
image/x-raw-olympus	orf	No	Yes
image/x-raw-panasonic	rw2	No	Yes
image/x-raw-pentax	pef	No	Yes
image/x-raw-red	r3d	No	Yes
image/x-raw-sigma	x3f	No	Yes
image/x-raw-sony	arw	No	Yes
image/x-xbitmap	xbm	No	Yes
image/x-xpixmap	xpm	No	Yes
image/x-xwindowdump	xwd	No	Yes
text/csv	csv	Yes	Yes
text/html	html	Yes	Yes
text/plain	txt	Yes	Yes
text/xml	xml	Yes	Yes

Setting up 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 information gives an overview on the content stores, their types, and configuration details with examples.

Content stores overview

Background information on the 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 information provides an overview on the File content store and additional content stores that you can use with Alfresco.

File content store

The File content store 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.

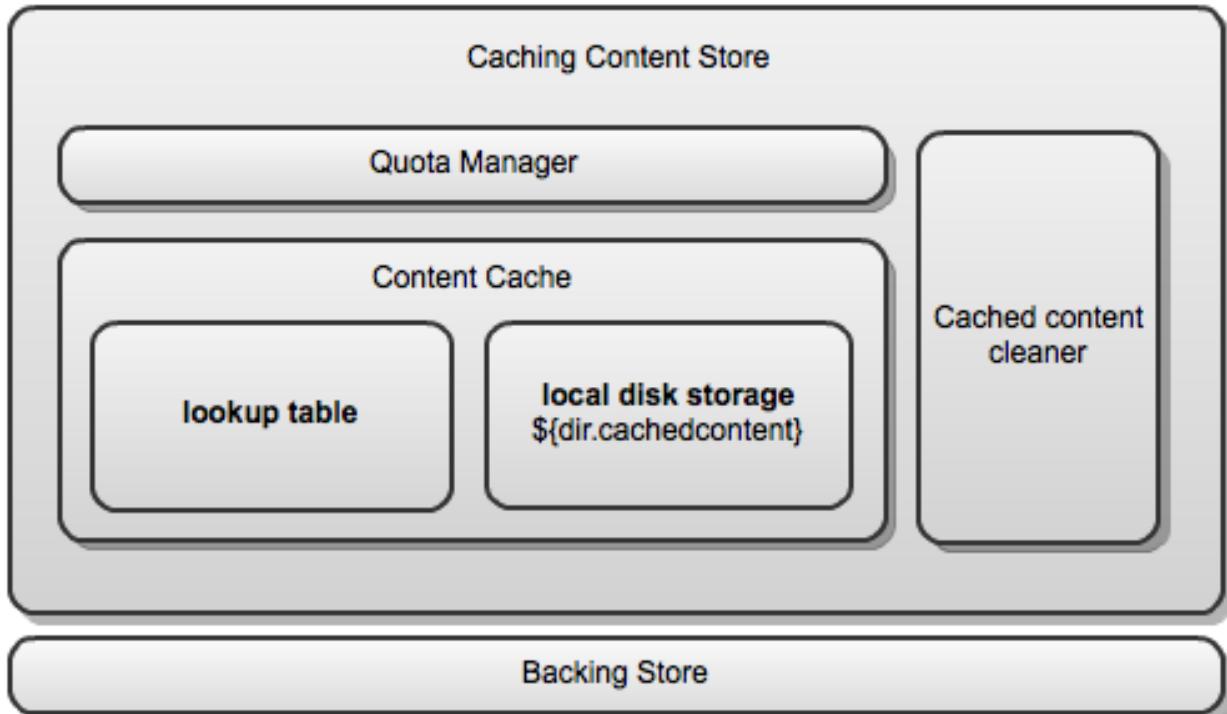
Caching content store (CCS)

This information 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

There are a number of 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>
```

4. 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="${system.content.caching.maxUsageMB}"/>
  <property name="maxFileSizeMB"
  value="${system.content.caching.maxFileSizeMB}"/>
  <property name="cache" ref="contentCache"/>
  <property name="cleaner" ref="cachedContentCleaner"/>
</bean>
```

5. 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.

-  The Aggregating content store is based upon the Replicating content store that was included in prior releases of Alfresco, but supports specifically the content aggregation use case, not content replication.

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>
    </bean>
</beans>
```

```

<!-- 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

The Encrypted content store is used for content encryption at rest. This information provides an overview of the Encrypted content store, its components, and how it is administered.

Content encryption overview

Use this information to understand Alfresco's implementation of content encryption using the Encrypted content store.

-  Once you make the decision to use Encrypted content store for content encryption, it is irrevocable. This is because when a document is written to this content store, it is encrypted. If you decide to revert to an unencrypted content store, the content cannot be decrypted.
-  If Encrypted content store is enabled on an existing or upgraded Alfresco installation, only new content will be encrypted but any existing content will not be encrypted.

Alfresco cryptography process

The Encrypted content store provides content 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.

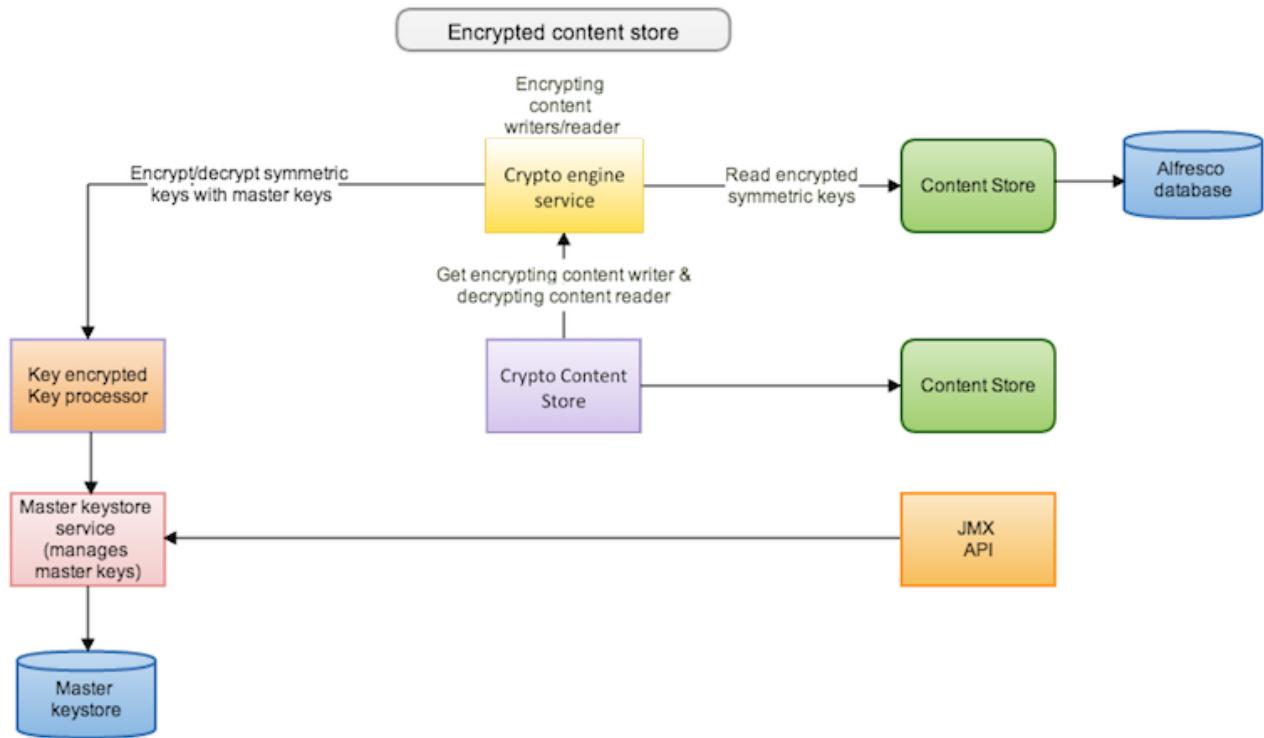
When a document is written to the Encrypted content store, the Encrypted content store uses symmetric encryption to encrypt the document before it is written to the wrapped content store. A new symmetric key is generated each time a document is written to the content store. This means that every document 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 implementation of content encryption using the Encrypted content store over the default Alfresco content store.



Issues to consider before using Encrypted content store

Consider these issues 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 any other 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

Follow these steps to install the Encrypted content store.

Before you begin, ensure that you have an instance of Alfresco installed on your machine. For more information, see [Installing Alfresco using setup wizards](#).

- Obtain the license (.lic) file with content encryption enabled from Alfresco Support.
- 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

Use this information to configure the Encrypted content store.

The Encrypted content store is configured using the properties in the `alfresco-global.properties` file and can also 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).

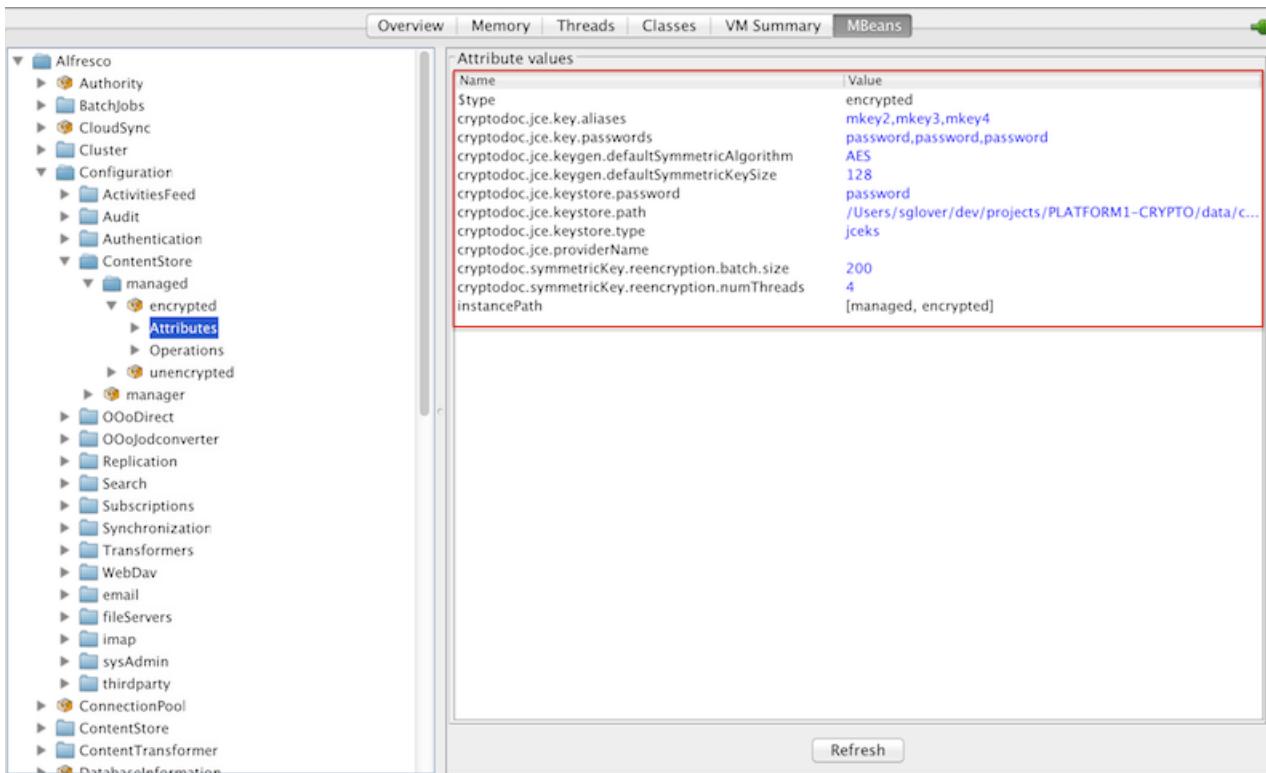
```
filecontentstore.subsystem.name=encryptedContentStore
cryptodoc.jce.keystore.path=/Users/<username>/dev/projects/PLATFORM1-CRYPTO/
data/crypto.jks
cryptodoc.jce.keystore.password=password
cryptodoc.jce.key.aliases=mkey1,mkey2
cryptodoc.jce.key.passwords=password,password
cryptodoc.jce.keygen.defaultSymmetricKeySize=128
```

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.



Encrypted content store properties

There are a number of 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.

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 keystore. 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

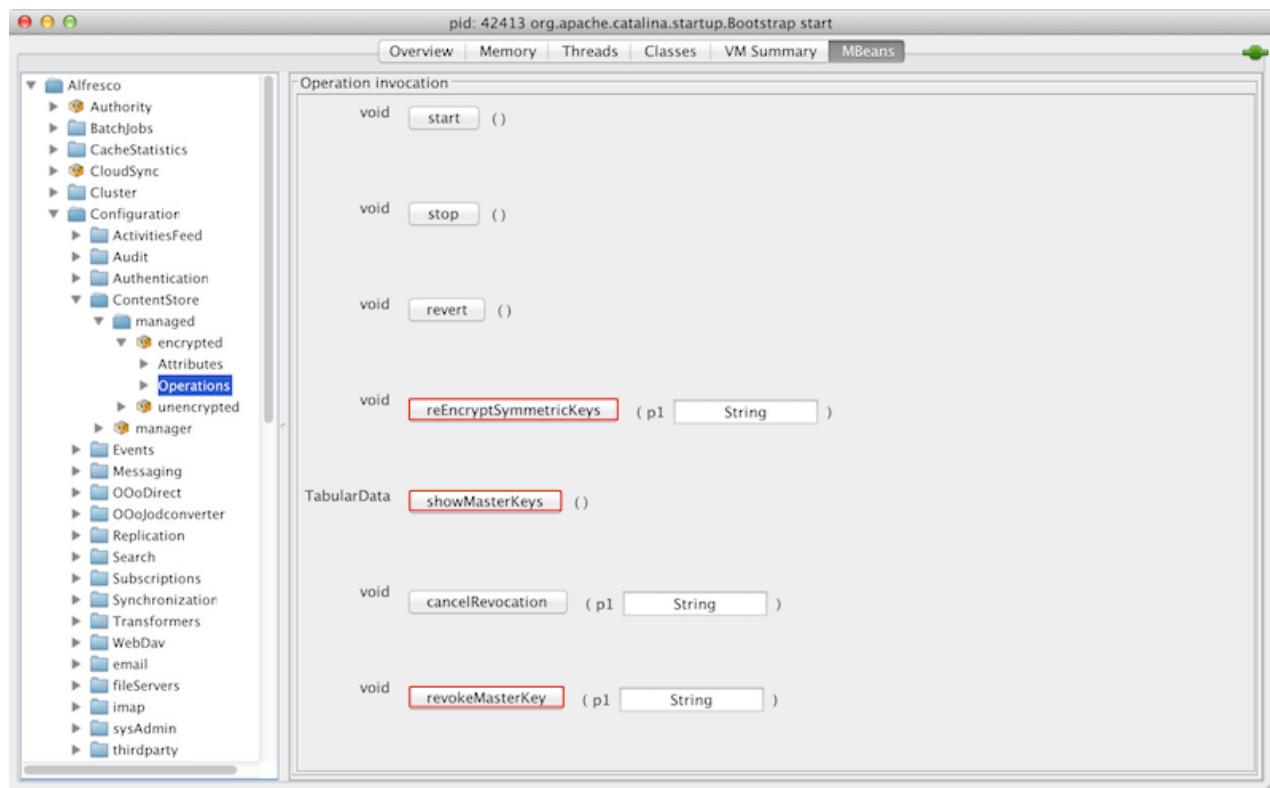
If you have installed the Oracle Java SE Development Kit (JDK), you can use JMX operations to perform some common tasks for Encrypted content store.

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.

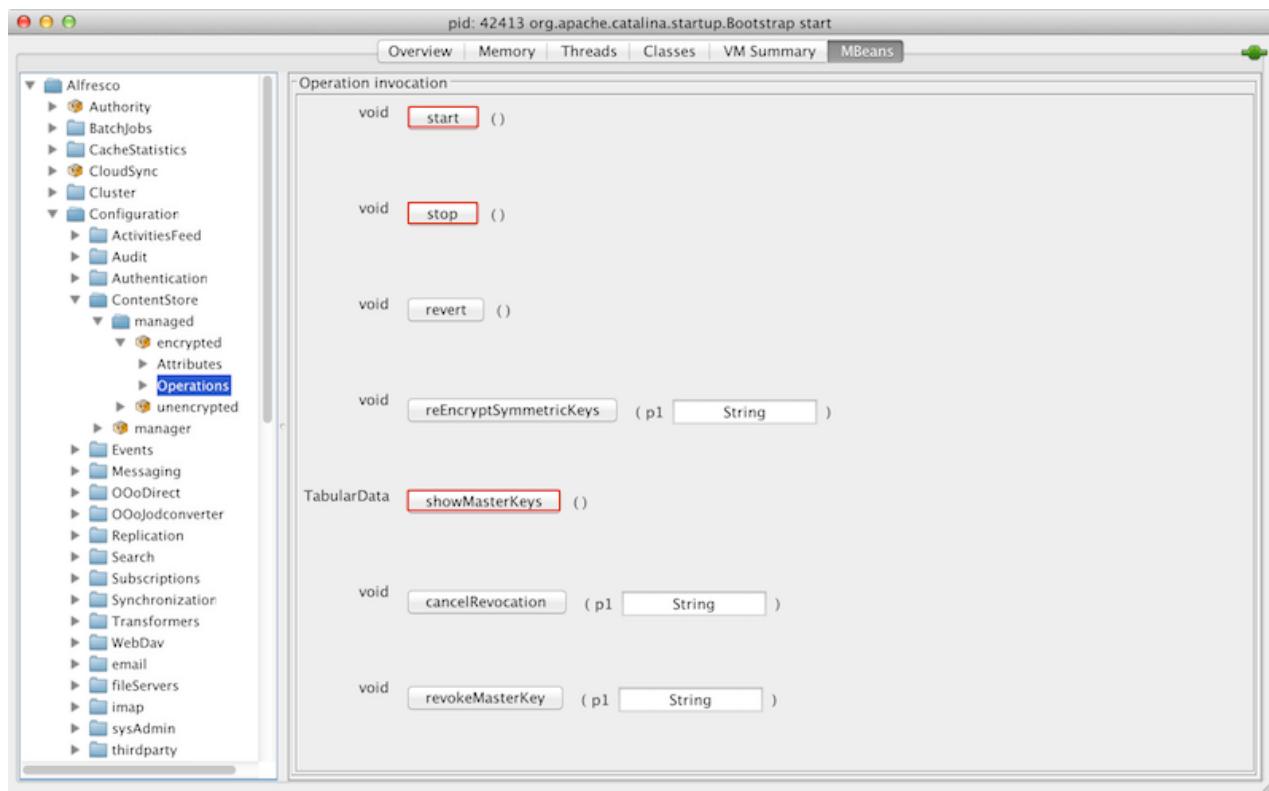


The values set on a subsystem will mean that the property values from configuration files may be overwritten or ignored. Use the JMX client to set the configuration properties.

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.alia...es	mkey2,mkey3,mkey4
cryptodoc.jce.key.p...sswords	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.

Alfresco S3 Connector

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](#).

Alfresco EMC Centera Connector

The Alfresco EMC Centera Connector module provides integration between Alfresco and Content Addressable Storage (CAS) systems.

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 the Alfresco EMC Centera Connector, and setting up `CenteraContentStore` as your main content store, see [Installing and configuring the Alfresco EMC Centera Connector](#) on page 60.

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>
```

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>
```

```

        </property>
    </bean>

</beans>
```

Customization for Alfresco Share:

The following example shows the configuration of the `cm:storeSelector` aspect using the `share-config-custom.xml` file:

```

<!-- Example config to expose the storeSelector in Share. This should be merged
     with your existing share-config-custom.xml -->

<!-- Configuring in the cm:storeSelector aspect -->
<config evaluator="node-type" condition="cm:content">
    <forms>
        <form>

            <field-visibility>
                <!-- aspect: cm:storeSelector -->
                <show id="cm:storeName" />

            </field-visibility>
            <appearance>
                <!-- Store Selector -->
                <field id="cm:storeName" label="Store Name" description="Content
Store Name" />
            </appearance>
        </form>
    </forms>
</config>

<config evaluator="string-compare" condition="DocumentLibrary"
replace="true">
    <aspects>
        <!-- Aspects that a user can see -->
        <visible>
            <aspect name="cm:storeSelector" />
        </visible>
    </aspects>
</config>
```

Managing content stores

Use this information 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 organized 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 store, 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)

You can 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.

Setting up and managing 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 replicate content between source and target repositories.

1. Shut down the Alfresco server on the source and target repositories.
2. In the source repository, open the `alfresco-global.properties` file and make the following updates:

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 jobs. You might see this error message in Share:



Alternatively, you can enable content replication from the Admin Console. See [Admin Console: Working with the replication service](#) for more information.

3. Save the file.
4. In the target repository, open the `alfresco-global.properties` file and make the following updates:
 - a. Set the `transferservice.receiver.enabled` property to `true`:

```
transferservice.receiver.enabled=true
```
5. Save the file.
6. Restart Alfresco on both source and target repositories.
7. Configure the Share URL to allow access to the source repository, as specified in [Opening locked content in the source repository](#) on page 594.
8. Create a transfer target, as specified in [Creating a new transfer target for content replication](#) on page 593.

Enabling the Replication Service

Replication Service in the 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.

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.

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. This folder contains a group called **Default Group**.

You can add transfer targets by creating a new transfers folder.

1. In the source repository, create a new folder in **Company Home > Data Dictionary > Transfers > Transfer Target Groups > Default Group**.
 - a. In the **New Folder** window, specify the name, title, and description of the new folder, for example, **Replica**.

A rule defined on the **Default Group** folder specializes the type of any folder created in 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.

- b. Click **Edit Properties** on your new folder (**Replica**).
- c. Specify the **Endpoint Host**, **Endpoint Port**, **Username** and **Password**, and click **Enabled** and **Save**.

For example; Endpoint host: `localhost`, Endpoint port: `9080`, Username: `admin`, Password: `admin`

The endpoint host and port represent the remote Alfresco host details.

- d. Enable the replication service in your `alfresco-global.properties` file:

```
replication.enabled=true
```

and restart the source repository.

2. In the target repository, enable the replication server and content receiver in the `alfresco-global.properties` file:

```
replication.enabled=true
transferservice.receiver.enabled=true
```

and restart the target repository.

3. On the source repository, create a replication job to test the target setup.

- a. From the Alfresco toolbar, click **Admin Tools** and select **Replication Jobs** from the menu.
- b. Click **Create Job**.
- c. Specify properties for **Name**, **Payload**, **Transfer Target**.

Name is a new folder name; for example, **Replication Job**. **Payload** is the source content directory, and **Transfer Target** is the folder name that you set up in [step 1 \(Replica\)](#).

- d. Click **Enabled**.
- e. Click **Create Job**.
- f. Refresh the screen after a few minutes to see a status change.

4. Verify the replication job.

Log in to Alfresco Share on the target repository, select a transferred file and click **Open in Source Repository** to check that content has replicated.

Opening locked content in the source repository

1. On the source repository, locate your current `repositoryId` in **Admin Console > General > Repository Information**:

```
http://localhost:8080/alfresco/s/enterprise/admin/admin-repositoryinfo
```

2. On the target repository, save the `<web-extension>\share-config-custom.xml.sample` file as `<web-extension>\share-config-custom.xml`.

- a. Locate the following example configuration in your `<web-extension>\share-config-custom.xml` file:

```
<config evaluator="string-compare" condition="Replication">
    <share-urls>
        Example config entry:
    </share-urls>
</config>
```

```
<share-url repositoryId="622f9533-2a1e-48fe-af4e-  
ee9e41667ea4">http://new-york-office:8080/share/</share-url>  
</share-urls>  
</config>
```

- b. Uncomment the `<share-url>` element.
 - c. Modify the `repositoryId` to match the value you located in step 1.
 - d. Change the URL to point to `http://localhost:8080/share`.
 - e. Save the `<web-extension>\share-config-custom.xml` file.
3. On the target repository, reload the configuration by refreshing the web scripts:

`http://localhost:9080/share/service/index`

Managing replication jobs

The **Replication Jobs** tool in Share Admin Tools 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 the 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 a report:

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

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

Importing and transferring files

Use this information to import files using the Bulk Import Tool, or transfer files using the File System Transfer Receiver (FSTR).

Using the Bulk Import tool

The Bulk Import tool provides a mechanism for Systems Administrators to import existing content in bulk 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).

There are two types of bulk import:

- Streaming import: This 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 bulk import feature imports files that already exist within the repository content store. As no copying is required, this gives significant performance improvements.

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 files 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 [Managing the content store](#). This allows you to select 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 set up the parameters for 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 POST 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 v5.1.0 (r122178-b10)

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:

- 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 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 set up the parameters for 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 POST 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 v5.1.0 (r122178-b10)

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 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** keyboards 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.

For more information about the fields and their meanings, see [Bulk Import tool fields and values](#) on page 604.

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 601.

For more information about the bulk import fields and their meanings, see [Bulk Import tool fields and values](#) on page 604.

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 No n/a</ResultOfLastExecution>	Yes No n/a	Result of the bulk import

Field label (from Bulk Import status web page)	Field entry (from status.xml file)	Possible values	Summary
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
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

Field label (from Bulk Import status web page)	Field entry (from status.xml file)	Possible values	Summary
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
Content Versions: # Data Written	<ContentVersionsBytesWritten>0</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

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-one-file-transfer-receiver-5.1.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

Use this information to start the File System Transfer Receiver.

1. Ensure that you have expanded the File System Transfer Receiver zip file:
`alfresco-one-file-transfer-receiver-5.1.zip`

2. To run the File System Transfer Receiver, enter the following command:

```
java -jar file-transfer-receiver.jar
```

You can navigate to `http://<FSTR-host-name>:<FSTR-port>/alfresco-ftr/service/index` to see if the FSTR is running. Information messages 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.

This file contains the Tomcat base directory and the port number to startup on.

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.

Property	Description
ftr.tomcat.portNum=	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.

This file is used to configure the operation of FSTR. It contains the settings for the root directory, staging directory, derby database connection string, username, and password.

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. The database must be on a local disk to ensure data integrity.
<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 the `log4jproperties` file. This information describes the `log4j` properties that you can set.

For example:

```
log4j.logger.org.alfresco.repo.transfer.fsr=warn
log4j.logger.org.alfresco.repo.web.scripts.transfer=warn
```

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.

Monitoring Alfresco

There are a number of methods for monitoring Alfresco.

Downloading the JMX Dump

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.

JMX monitoring and management extensions

This information 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 Oracle JMX agent and local JMX connectivity

When using Tomcat and a Oracle JVM together for monitoring, you can configure Alfresco and Tomcat to share the JVM's own platform MBean server. The pre-registered MXBeans 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.

Oracle'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 Oracle 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 Oracle JMX agent in local mode. This means the platform MBean Server is shared by Alfresco and still be available for remote connections through the RMI connector.

 Restrict JMX RMI connections to an internal administration group, due to security vulnerabilities. JMX/RMI deserializes data from a client before authentication, which means that password protection does not provide adequate security.

- To activate the Oracle 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 Oracle 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. These jobs are defined in the `<configRoot>/classes/alfresco/scheduled-jobs-context.xml` file.

Scheduled job	Description
contentStoreCleanerTrigger	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 efficiency.
nodeServiceCleanupTrigger	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 efficiency.
tempFileCleanerTrigger	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.

Scheduling cleanup of database tables

You can schedule or manually trigger the `propTablesCleanupTrigger` script to clean up audit and property values tables (`alf_audit_` and `alf_prop_` tables).

`propTablesCleanupTrigger` is a `MonitoredCronTrigger` script, and a schedule to run this script can be set in the `repository.properties` file or by using JMX. This script does not run by default.

To run this script, add the following line to the `repository.properties` file:

```
attributes.propcleaner.cronExpression=* * * * ? 2099
```

Here is an example of the output you should expect to see in the debug information:

```
2014-07-24 13:22:17,493 INFO [schema.script.ScriptExecutorImpl]
[DefaultScheduler_Worker-3] Processing from 0 to 10000 rows of 2 rows from
table alf_audit_app.
2014-07-24 13:22:17,493 INFO [schema.script.ScriptExecutorImpl]
[DefaultScheduler_Worker-3] Processing from 0 to 1 rows of 0 rows from table
alf_audit_entry.
2014-07-24 13:22:17,509 INFO [schema.script.ScriptExecutorImpl]
[DefaultScheduler_Worker-3] Processing from 0 to 10000 rows of 1 rows from
table alf_prop_unique_ctx.
2014-07-24 13:22:17,524 INFO [schema.script.ScriptExecutorImpl]
[DefaultScheduler_Worker-3] Processing from 0 to 10000 rows of 3 rows from
table alf_prop_root.
```

The default batch size is 10000 rows.

-  The `propTablesCleanupTrigger` script is designed for occasional, scheduled cleanups of the database tables. Ensure that you run this script during periods when there is minimal or no load on the server. If there is load on the server while `propTablesCleanupTrigger` script is running, you might experience database conflicts and related errors.

Backing up and restoring

This information describes the process for backing up the Alfresco content repository only. It assumes that components other than the data residing in Alfresco (operating system, database, JDK, application server, Alfresco binaries and configuration, etc.) are being backed up independently.

Your backup strategy must be tested end-to-end, including restoration of backups. Ensure that you have adequately tested your backup scripts prior to deploying Alfresco to production.

Backing up and restoring the repository

Backing up an Alfresco repository involves backing up the directory pointed to by the `dir.root` setting, the database that Alfresco is configured to use, and the Solr 4 indexes.

For backing up the Solr 4 indexes using the Share Admin Console or the `alfresco-global.properties` file, or JConsole, see [Backing up Solr 4](#).

To restore the backup successfully, the `contentstore` 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 (`contentstore` directory) and the Alfresco database from the same backup set. Otherwise, the repository may 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

By default, the `dir.root` contains both the `contentstore` and indexes. For a cold backup, back up the database and the content, and perform a full reindex when a backup is restored. A full reindex can be a time consuming process, so these steps include the indexes in the backup, removing the need to perform a reindex.

1. Stop Alfresco.
2. Back up the database Alfresco is configured to use, using your database vendor's backup tools.
3. In parallel, backup the `dir.root` directory (only the `contentstore` and `contentstore.deleted` directories).
Backing up the `contentstore.deleted` directory is optional.
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.

Performing a hot backup

The high-level procedure and order for a hot backup is:

1. Backup the Solr 4 indexes first
2. Then backup the database

3. Finally backup the `contentstore`

Solr 4 indexes have to be backed up first and before the database because if new rows are added in the database after the Solr 4 backup is done, a Solr 4 reindex (AUTO) can regenerate the missing Solr 4 indexes from the database transaction data.

Database backup should be done before backing up the `contentstore` because if your database points to a missing file, then you will not be able to retrieve content for that node. Also, if you have a file without the database data, this just means that the user has added the file too late to be included in a backup and the file will be orphaned.

It is critical to perform hot backups in the following order of steps:

1. Ensure that you have a `solr4Backup` directory under `dir.root`.
2. Backup the database Alfresco is configured to use, using your database vendor's backup tools.
3. As soon as the database backup completes, backup the specific subdirectories in `dir.root`.
4. Store both the database and `dir.root` backups together as a single unit.

For example, store the backups in the same directory or in a single compressed file. Do not store the database and `dir.root` backups independently, as that makes it difficult to reconstruct a valid backup set, if restoration becomes necessary.



By default, the Solr 4 indexes are backed up according to the cron job specified by the `solr.backup.alfresco.cronExpression` and `solr.backup.archive.cronExpression` properties, which can be set in `alfresco-global.properties`. By default, the cron job is run at 2 am for `alfrescoCore` and 4 am for `archiveCore`. You must ensure that the indexes are not backup up while these jobs are running.

Alfresco includes a background job responsible for backing up the Solr 4 indexes that (by default) is configured to run at 3am each night. The hot backup process must not run concurrently with this background job, so you should either ensure that the hot backup completes by 3am, or wait until the index backup job has completed before initiating a hot backup.

For more information on backing up Solr 4 indexes, see [Solr 4 backup and restore](#).

Backing up the database

Database hot backup requires a tool that can snapshot a consistent version of the Alfresco database (that is, it must capture a transactionally-consistent copy of all the tables in the Alfresco database). In addition, to avoid serious performance problems in the running Alfresco system while the backup is in progress, this snapshot operation should either operate without locking in the Alfresco database or it should complete quickly (within seconds).

Backup capabilities vary widely between relational database products, and you should ensure that any backup procedures that are instituted are validated by a qualified, experienced database administrator before being put into a production environment.

Backing up the file system

Follow these steps when backing up the file system.

Backup the following subdirectories of the Alfresco `dir.root` directory using whatever tools you are comfortable with (`rsync`, `xcopy`):

- `contentstore`
- `contentstore.deleted` (optional)
- `solr4Backup`

-  Do not attempt to backup the `solr4/index` subdirectory while Alfresco is running. This will cause Solr 4 index corruption. Use `solr4Backup` instead.

Troubleshooting

Help for diagnosing and resolving any Alfresco issues that you might encounter.

For additional help, refer to the following:

- Alfresco Support Portal (<http://support.alfresco.com>)
- **Alfresco Admin Console**: see [Launching the Admin Console](#) on page 130 for more information
- **Admin Tools** in Alfresco Share to view various installation and setup information
- Alfresco forums (<http://forums.alfresco.com/>)

Handling a higher rate of outbound TCP connections

1. Open the Registry.
2. Under the following registry entry:
`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TCPIP\Parameters`
3. Key in the registry of the Windows client machine.
4. Add the following registry entries:
 - TcpTimedWaitDelay**
Add this DWORD with a value of 30.
 - MaxUserPort**
Add this DWORD with a value of 32768.
5. Refer to the Windows documentation for further details on these registry entries.

Setting log levels

The `log4j.properties` file lets you configure logging levels to provide debugging information when troubleshooting. To set up logging policies, you must prepend `log4.logger` to the class name you want to log to, and set the logging level. You can set the log level dynamically using the JMX client.

When using `log4j`, you should:

- Keep local customizations and licenses outside of the web application. For example, in the extension directory:
`$TOMCAT_HOME/shared/classes/alfresco/extension/...-log4j.properties`
- The Alfresco supplied configuration files should be stored or installed within the web application. For example:
`WEB-INF/classes/alfresco/extension/...-log4j.properties`

 A `dev-log4j.properties` file should not be packaged as a part of any product.

Logging uses the Log4J `HierarchyDynamicMBean`.

-  Log levels are not cluster-aware. If needed, the log level change will need to be applied to each machine. Some consoles (for example, JManage) can provide basic facilities for accessing each machine in an application cluster.
- Editable attributes are a dynamic list of loggers with the `logLevel` attribute, which can be changed to OFF, FATAL, ERROR, WARN, INFO, DEBUG or TRACE (editable).

- `addLoggerMBean` will be impacted if it has been loaded.

The following steps provide instructions on adding loggers using JConsole:

1. Click **Alfresco** -> **Log4jHierarchy** -> **Operations** -> **addLoggerMBean**.
2. Type the full **className** in **Name** on the right hand pane.
3. Click **addLoggerMBean**.

A dialog box is displayed with the title **Operation return value**. If the operation is successful, the body of the dialog box contains the **className** you provided, preceded by **log4j:logger=**. If the operation is unsuccessful, the body of the dialog box shows **null**.

Error messages

ImageMagick

Error message on the console:

```
ERROR [AbstractImageMagickContentTransformer]
JMagickContentTransformer not available:
ERROR [AbstractImageMagickContentTransformer]
ImageMagickContentTransformer not available:
Failed to execute command: imconvert ...
```

These issues will not cause the server to fail. Alfresco is reporting that external document transformation engines are not available for use by the server. You can remove the transformation references if they are not required.

JAVA_HOME

Make sure the `JAVA_HOME` variable is set correctly for your Java installation.

FTP Socket

Error message on server startup:

```
ERROR [protocol] FTP Socket error
java.net.BindException: Address already in use:
JVM_Bind at
java.net.PlainSocketImpl.socketBind(Native Method)
```

Check to see if you have any services running against port 8080 for the Alfresco server or port 21 for the Alfresco FTP integration.

Using the Node Browser

Use **Node Browser** in the Admin Console or in Share Admin Tools as an debugging aid to browse the raw Alfresco repository structure. This feature is intended for developers responsible for customizing the application.

This is a read-only feature with basic search capability.

1. Open the Admin Console.
2. In the **Support Tools** section, click **Node Browser**. You see the **Node Browser Console** page.
3. In the **Store** section, select the store of interest:
 - `user://alfrescoUserStore`
 - `system://system`
 - `workspace://lightWeightVersionStore`

- workspace://version2Store
- archive://SpacesStore
- workspace://SpacesStore

Each store is an area of the repository and within each store, the nodes of that store are organized hierarchically. The node displayed is the root node of the selected store.

4. Click **Root List**.

The **Node Browser** page displays details of the properties, aspects, children, parents, associations, source associations, and permissions for the selected node.

5. Search the selected store, as needed:

- a. Select the search type: noderef, fts-alfresco, lucene, xpath, selectnodes, cmis-strict, cmis-alfresco, db-afts, db-cmis.
- b. Enter the search criteria in the field provided.
- c. Click **Execute**.

Using the Node Browser in Share Admin Tools

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:

- storeroot
- noderef
- xpath
- **fts-alfresco**
- cmis-strict
- cmis-alfresco
- db-afts
- db-cmis

Debugging an Alfresco installation

When developing add-ins, fixing bugs, or changing Alfresco from the source code, it is helpful to debug an instance of Alfresco running on a standard application server. You can configure Alfresco and Eclipse to provide a real-time view of the server and to troubleshoot issues by stepping through the code line by line.

To debug a running Alfresco server, you must connect to the JVM in which Alfresco is running. The following steps configure the JVM to expose an interface for this connection, and then configure Eclipse to connect to and control that JVM.

Configuring the JVM

You can configure the JVM to expose an interface for connection to the Alfresco server.

Before you start, you must:

- Have a fully installed, configured, and running instance of Alfresco. These steps assume you are using Tomcat on Windows, but the steps are similar for other application servers on other systems.
 - Have an IDE installed. These steps describe how to configure Eclipse, which must be installed first ([Eclipse](#))
 - Download and install the Alfresco source code from [Alfresco source code](#).
 - Ensure the source code is the same version as the installed Alfresco server.
1. Verify that the Alfresco server is not running.
 2. Edit the JVM options used to start the Alfresco Tomcat instance.

For example, set the following:

```
JAVA_OPTS=%JAVA_OPTS% -server -Xdebug -  
Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=8082
```

where `address` is a port for your system.

3. Save the file and close the editor.

Configuring Eclipse

This task describes how to configure Eclipse to connect to and control the JVM.

1. From the Run menu, choose the **Open Debug** dialog.
2. Right-click **Remote Java Application** and select **New**.
3. In the Name box, type `Debug Local Tomcat Alfresco`.
4. Next to Project, click **Browse**, and select **Web Client**. If this is not available as an option, ensure your source code matches that of your server.
5. In Connection Properties, enter the port number.
6. Check **Allow Termination of remote VM** if you want to be able to stop the Alfresco server from the Eclipse console.
7. Click **Apply** to save the configuration.

Troubleshooting an upgrade

Use these tips for diagnosing and resolving any issues that might arise as a result of an upgrade.

1. Immediately after starting the Alfresco server, make a copy of the `alfresco.log` file.
2. In the `alfresco.log` file, note the locations of the temporary files containing the SQL statements executed during the upgrade, and make a copy of these temporary files.
3. Submit the log file and temporary files to Alfresco Support.

Troubleshooting rules and actions

Use these troubleshooting tips when working with rules and actions.

Type specialization action problems with Mac OS/X

If you are using Mac OS/X 10.8.3 or later, the type specialization action is not performed when you save a Microsoft Word document.

To resolve this issue, edit your `alfresco-global.properties` file to set the following value:

```
policy.content.update.ignoreEmpty=false
```

Troubleshooting clustering

Use these troubleshooting tips when testing cache clustering.

- On Linux and Unix environments, you can use `netstat -ln` to check that the correct ports have been opened by the Alfresco server on the correct network adapters. You can use `telnet <hostname><port>` to check if each open port can be reached by each cluster member.
- If your cluster members are using NAT and IPv4 addresses, you might need to force the server to listen on IP V4 addresses rather than IP V6. To do this, add:

```
-Djava.net.preferIPv4Stack=true
```

to the startup options of Alfresco's JVM. In a standard Linux/Unix installation, this would require editing of the `JAVA_OPTS` variable in the script, as follow:

```
tomcat/scripts/ctl.sh
```

On a standard Windows installation, this would require adding the parameter just before `--Dalfresco.home` in:

```
tomcat/bin/service.bat
```

and then running the scripts:

```
tomcat/scripts/serviceinstall.bat REMOVE  
tomcat/scripts/serviceinstall.bat INSTALL
```

to re-register the Alfresco service with the new option.

For more information on the process of initiating clustering and the options available for configuring Alfresco clustering, see [Setting up clustering](#) on page 465.

Troubleshooting LibreOffice subsystems

Use these tips for troubleshooting the LibreOffice subsystems.

1. Enable the following log4j properties to debug:

```
log4j.logger.org.alfresco.enterprise.repo.content=DEBUG  
log4j.logger.org.artofsolving.jodconverter=DEBUG
```

For information about how to create a `log4j.properties` file, see [Key tools and files](#).



The OOoDirect debug entry is:

```
log4j.logger.org.alfresco.repo.content.transform=DEBUG.
```

2. If Tomcat is not shutdown gracefully, the `soffice.bin` process cannot be stopped. This can result in errors when starting Tomcat with port 8080 being used. If this occurs, manually kill the `soffice.bin` process.
3. You might see a failure to connect error message.

If the LibreOffice process takes more than 10 seconds to fully start up, then Alfresco fails to connect to it. If this occurs, manually kill the `soffice.bin` process before attempting to restart the Jodconverter subsystem.



The next time that you start LibreOffice, it usually starts fast enough to connect (this is due to operating system caching).

4. If the LibreOffice home location is incorrect, the Jodconverter subsystem will still start, but no LibreOffice process will be running or connected. The error is reported in the console but not in the `alfresco.log` file.

The correct value for the `jodconverter.officeHome` property varies with host operating system.

- For Mac OS X, it should be set to the directory that contains `MacOS/soffice.bin`, which is `/Applications/LibreOffice/Contents` by default.
 - For other operating systems, it should be set to the directory that contains `program/soffice.bin`.
5. When restarting the Jodconverter subsystem using JMX, you need to set the `enabled` property to true (this will also stop the JOD subsystem if it is running); then use the `start` operation to start the Jodconverter subsystem with the new property settings.
 6. The JodConverter can run a pool of multiple reusable instances of the `soffice` LibreOffice process. To use this capability, set the `jodconverter.portNumbers` property to a comma-separated list of port numbers, all of which must be available for use. For example, `2022, 2023, 2024` for a pool of three `soffice` processes.
 7. The JodConverter supports configurable restart behavior for the LibreOffice `soffice` process. To ensure that potential memory leaks in LibreOffice do not accumulate and affect performance, the JodConverter will restart an `soffice` process after a given number of tasks (transformations, metadata extractions) have been performed. The default for `jodConverter.maxTasksPerProcess` is `200`.
 8. The JodConverter allows long-running or hung tasks to be timed out. The first timeout is controlled by `jodconverter.taskQueueTimeout`, which is `30000` by default (`30000 milliseconds = 30 seconds`). If a task spends this long in a JodConverter queue awaiting execution, it will be dropped from the queue. The second timeout is controlled by `jodconverter.taskExecutionTimeout`, which is `120000` by default (`120000 milliseconds = 2 minutes`). If a task has been executing within an `soffice` process for longer than this period, that `soffice` process will be terminated and restarted.
 9. Throughput of Oo-related tasks, such as transformations, can be balanced against available hardware resources (memory, CPU) by altering the pool size and the two timeout timers.

Troubleshooting the JMX Dumper

Use this information if you need to troubleshoot the JMX Dumper.

Invoking the JMX Dumper can result in a stack trace in the log file. When you open `jmx-dumper.zip`, it is trying to find a data source defined in the `web.xml` file. (`<res-ref-name>jdbc/>` `dataSource</res-ref-name>`), but this data source is not declared in the `alfresco.xml` file.

To prevent this logging message for appearing, you can configure the data source in the `$CATALINA_BASE/conf/[enginename]/[hostname]/alfresco.xml` file.

Troubleshooting CIFS

Use this information for diagnosing and resolving any issues when configuring CIFS.

Password Error

Sometimes, when connecting to an instance of Alfresco Share, the login dialog appears several times until finally taking effect. This problem can be caused by the Share connecting to the Windows file server that is running on native SMB/port 445 rather than trying to connect by using NetBIOS.

Troubleshooting NTLM

Use this information for diagnosing and resolving any issues that might arise when configuring NTLM.

Alfresco supports NTLM v2 protocol, which is more secure than NTLM v1 protocol. However, NTLM v2 cannot be used with pass-through authentication. You will have to switch to NTLM v1 if you want to use pass-through authentication, where Alfresco passes the log on request to an Active Directory or other server to validate the login credentials. For more information, see the [Configuring pass-through](#) topic.

To authenticate using NTLM v1, set the following registry key on your client machines:

```
[HKLM\SYSTEM\CurrentControlSet\Control\Lsa]
"LMCompatibilityLevel"=dword:00000001
```

Issue:

Failure of NTLM logon on machines running Windows 7 or Internet Explorer 8.

Troubleshooting

This problem is most likely caused by enhanced security in Windows 7, Vista and Windows 2008. Previous versions of Windows (XP) would fall back to NTLM v1, if NTLM v2 failed.

1. On Windows 7 clients, navigate to **Control Panel > Administrative Tools > Local Security Policy**.
2. In the left pane, navigate to **Security Settings > Local Policies > Security Options**.
3. In the right pane, find **Network Security: LAN Manager authentication level**. By default, the value of **Network Security: LAN Manager authentication level** is set to **Send NTLMv2 response only. Refuse LM & NTLM**.
4. Set the value of **Network Security: LAN Manager authentication level** to **Send LM and NTLM - use NTLMv2 session security if negotiated**.

This setting allows Windows 7 to use the more secure NTLM v2, if available, and fall back to NTLM v1 for Alfresco. If the machines are in a domain, it is possible to change this setting on all of them by using the group policy editor on the domain controller.

Troubleshooting WebDAV

Alfresco uses two implementations of WebDAV:

- RFC-compliant WebDAV: alfresco/webdav
- Microsoft-compliant WebDAV: alfresco-aos

Microsoft WebDAV extensions (MS-DAVEXT) are only partially compatible with the WebDAV standard, therefore it is recommended that you use /alfresco-aos on Windows clients and /alfresco/webdav on Linux-based systems.

Unable to connect to locally installed Alfresco server by using WebDAV

- Check if Alfresco has finished loading. Look for a *Server startup* message in the log file

- Check if the connection works if you use the IP address instead of the host name
 - Check if you can browse folders using `https://<alfresco_ip>/alfresco/aos` in a web browser
 - Add your Alfresco server IP to the Trusted sites list in Windows Internet Explorer
 - Make sure the **WebClient** service is running. To do so, follow the steps:
 1. Start `services.msc`.
 2. Start the **WebClient** service.
-  For details on running the **WebClient** service, see [Mapping a network drive to Alfresco \(Windows users\)](#).
- If you are not using SSL, check your connection configuration for Windows and Microsoft Office. See [Setting up Alfresco Office Services using a non-SSL connection](#) on page 82 for more information.

 For details on setting the Basic Authentication Level key in the Registry Editor, see [Mapping a network drive to Alfresco \(Windows users\)](#).

 - If you can connect to the Alfresco server but cannot authenticate your login details, check if you can use the same user name and password to log in to Share.

Moving a file or folder using WebDAV on an Ubuntu client causes loss of metadata and creates a new node reference

There is a known problem where Ubuntu creates a new `nodeRef` when you move a file or a folder in WebDAV, because it uses PUT and DELETE methods instead of a MOVE method. As a result, the `nodeRef` for the file or folder changes and any associated metadata is lost. This issue applies to all versions of Ubuntu, but does not occur when using a Windows client.

Editor role cannot edit content using WebDAV and Cyberduck version 4.4+

There is a known issue when using WebDAV with Cyberduck 4.4 and later, where content cannot be edited due to insufficient permissions. To avoid this problem, you can either use a version of Cyberduck earlier than 4.4, or assign permissions to the user to allow them to create files.

OpenLDAP tips

Use these tips when working with OpenLDAP.

There are a number of things to note:

- The maximum number of results returned has been increased from the default of 500 that even applies to paged results. See the OpenLDAP documentation on limits. If you have more than 500 users or groups this would be an issue.
- Digest authentication has been configured to map from a user ID to the corresponding distinguished name. See the example data.
- Passwords are in clear text (so that any authentication mechanism can be used). It is possible they can be in the correct hashed form for the MD5 digest to work.

```
See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
include  /usr/local/etc/openldap/schema/core.schema
include  /usr/local/etc/openldap/schema/cosine.schema
include  /usr/local/etc/openldap/schema/inetorgperson.schema

# Define global ACLs to disable default read access.

# Do not enable referrals until AFTER you have a working directory
```

```

# service AND an understanding of referrals.
#referral ldap://root.openldap.org

pidfile    /usr/local/var/run/slapd.pid
argsfile   /usr/local/var/run/slapd.args

# Load dynamic backend modules:
# modulepath /usr/local/libexec/openldap
# moduleload back_bdb.la
# moduleload back_ldap.la
# moduleload back_ldbm.la
# moduleload back_passwd.la
# moduleload back_shell.la

# Sample security restrictions
# Require integrity protection (prevent hijacking)
# Require 112-bit (3DES or better) encryption for updates
# Require 63-bit encryption for simple bind
# security ssf=1 update_ssf=112 simple_bind=64

# Sample access control policy:
# Root DSE: allow anyone to read it
# Subschema (sub)entry DSE: allow anyone to read it
# Other DSEs:
#   Allow self write access
#   Allow authenticated users read access
#   Allow anonymous users to authenticate
# Directives needed to implement policy:
# access to dn.base="" by * read
# access to dn.base="cn=Subschema" by * read
# access to *
#   by self write
#   by users read
#   by anonymous auth
#
# if no access controls are present, the default policy
# allows anyone and everyone to read anything but restricts
# updates to rootdn. (e.g., "access to * by * read")
#
# rootdn can always read and write EVERYTHING!

#####
# BDB database definitions
#####

database  ldbm
suffix   "dc=company,dc=com"
rootdn  "cn=Manager,dc=company,dc=com"
# Cleartext passwords, especially for the rootdn, should
# be avoided. See slappasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
# This is secret ....
rootpw      {SSHA}u9AUUYOSVX6idlXcwyYOAG6G84oHFpvG
# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd and slap tools.
# Mode 700 recommended.
directory  /usr/local/var/openldap-data
# Indices to maintain
index objectClass eq

# Clear text to allow hashing
password-hash {CLEARTEXT}

# SASL mappings for md5 digest authentication
# Extract the user id and use as the search key

authz-regexp
  uid=([^,]*),cn=digest-md5,cn=auth

```

```

ldap:///dc=company,dc=com??one?(uid=$1)

authz-regexp
  uid=(^,]*),cn=company.com,cn=digest-md5,cn=auth
  ldap:///dc=company,dc=com??one?(uid=$1)

# Tweaks to increase the result set size and max query time

sizelimit 50000
timelimit 3600

```

The following is a very simple example LDIF file that defines People and Groups Organizational units and some example users and groups.

```

# Initial directory contents
dn: dc=company,dc=com
dc: company
objectClass: top
objectClass: domain

dn: ou=People,dc=company,dc=com
ou: People
objectClass: top
objectClass: organizationalUnit

dn: ou=Groups,dc=company,dc=com
ou: Groups
objectClass: top
objectClass: organizationalUnit

dn: uid=fullname,ou=People,dc=company,dc=com
objectclass: inetOrgPerson
sn: Name
cn: Full Name
userPassword: inClearText
telephoneNumber: 1234567890
uid: fullname
givenName: Full
mail: full.name@company.com
o: Company Software Inc.

dn: uid=walrus,ou=People,dc=company,dc=com
objectclass: inetOrgPerson
sn: Rus
cn: Wal Rus
userPassword: inClearText
telephoneNumber: 1234567890
uid: walrus
givenName: Wal
mail: wal.rus@company.com
o: Company Software Inc.

dn: cn=Group One,ou=Groups,dc=company,dc=com
objectclass: groupOfNames
cn: Group One
member: uid=fullname,ou=People,dc=company,dc=com

dn: cn=Group Two,ou=Groups,dc=company,dc=com
objectclass: groupOfNames
cn: Group Two
member: cn=Group One,ou=Groups,dc=company,dc=com
member: uid=walrus,ou=People,dc=company,dc=com

```

Active Directory tips

Tips for using Active Directory with the LDAP synchronization.

- You might need to give special permissions in the Active Directory to the account that you are using to do the LDAP bind (as configured in `ldap.synchronization.java.naming.security.principal`). To do this, open Active Directory Users and Computers, right click on the domain, and select "Delegate Control..." Click "Next", then select the user that you are using for the LDAP bind and click "Next". The permission that they will need is on the next screen "Read all inetOrgPerson information."
- The example URL in `ldap.authentication.java.naming.provider.url` does not use SSL. SSL is recommended for production systems. You'll need to switch the port from 389 (below, non-SSL) to 636 for SSL.
- It is often helpful to screen out non-user accounts and disabled accounts. The default user queries in the `ldap-ad` subsystem type do this by checking bit fields on the `userAccountControl` attribute. For example:

```
userAccountControl:1.2.840.113556.1.4.803:=512
```

Troubleshooting SMTP inbound email using StartTLS

For StartTLS support to work for inbound email, you must configure SSL for Java.

To identify whether you are having this problem, enable `DEBUG` logging for the class `org.subethamail` in your `log4j.properties` file.

```
startTLS() failed: no cipher suites in common
```

Also, to enable efficient inbound mail server logging in debug mode, you need a `log4j` option that allows you to track mails, including the sender details, recipient details, subject and the reason for rejection/acceptance. To do so, enable `DEBUG` logging for the class `org.subethamail.smtp.server.ConnectionHandler` as shown:

```
log4j.logger.org.subethamail.smtp.server.ConnectionHandler=debug
```

The following process outlines one method for creating a self-signed certificate. However, this can differ between JVM vendors, so see the JVM documentation for more information.

1. Create a suitable key and certificate:

```
keytool -genkey -keystore mySrvKeystore -keyalg RSA
```

2. Add the following somewhere in your Tomcat configuration. For example, `/etc/tomcat5/tomcat7.conf`.

```
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.keyStore=mySrvKeystore - Djavax.net.ssl.keyStorePassword=123456"
```

 This method explains how to create a self-signed certificate only. SSL vendors can provide certificates signed by an authority and might be more suitable for production use.

Troubleshooting IMAP

IMAP scale limitations

If you mount more than 5000 folders or mailbox folders, depending on the IMAP client that you are using, you might not be able to view more than the first 5000 folders.

In order to avoid this situation, you should limit the number of folders that are being mounted. For example:

- Do not mount from the company root space if you know that you have a very large folder structure. Choose a specific site to reduce the number of folders being mounted.
- Do not extract attachments to a separate folder (`imap.attachments.mode=SEPARATE`), particularly for large repositories. When you specify `imap.attachments.mode`, choose one of the following settings:
 - `imap.attachments.mode=COMMON`: all attachments for all emails are extracted to one folder
 - `imap.attachments.mode=SAME`: attachments are extracted to the same folder as the original message

IMAP server error message

```
Exception in thread "Thread-53" java.lang.RuntimeException:  
java.net.BindException: Cannot assign requested address:  
JVM_Bind at com.icegreen.greenmail imap.ImapServer.run(ImapServer.java:53)  
Caused by: java.net.BindException:  
Cannot assign requested address: JVM_Bind
```

This error message is related to the IP address or hostname that has been provided for binding. To resolve this issue:

- Check that the IP address or hostname you provided is correct for your `imap.server.host` setting.
 - Check that the port you are using is not blocked. The default port to use is 143.
 - Check that firewalls are not blocking this IP address or hostname.
 - Use the command line tool Netstat to check your network connections.
-  You should not use localhost as the `imap.server.host` - update this value with the IP address (or corresponding DNS address) of your external IP interface. A value of 0.0.0.0 in Unix will make it listen on the specified port on all IP interfaces.

Troubleshooting schema-related problems

The Schema Difference Tool provides a way of identifying and troubleshooting problems in Alfresco database schemas.

Such problems can sometimes arise when performing certain version upgrades or customized installations.

Background

The Schema Difference Tool can be used when troubleshooting or examining the database schema for an Alfresco repository. The tool has two main functions:

1. Producing schema dumps as XML files.
2. Validating a database schema.

Schema dumps were available in previous versions of Alfresco. However, prior to the introduction of the Schema Difference Tool, the only way to judge the validity of the schema was to examine the file manually and compare schemas with simple text tools such as the Unix diff command. The Schema Difference Tool performs a certain amount of automatic comparison that removes much of the effort needed in making these comparisons.

If any changes are made to the database schema during server start-up (such as when installing Alfresco afresh) then the tool performs both schema dumping and validation as described. The

dumps and validation are made both pre-upgrade (that is before the schema changes) and post-upgrade.

Definition of terms used

The terms given are used throughout the rest of this document.

Database object

A schema, sequence, table, column, index, primary key or foreign key.

Reference schema

The definitive representation of an Alfresco repository schema for a given schema version on a vendor specific RDBMS. The reference schema is a model for what should be present in the database after installing or upgrading an Alfresco repository to a particular version. A reference schema is presented in the same XML format as a schema dump. For example, a schema reference can be produced for MySQL on version 5025 of the Alfresco repository schema.

Target schema

The database schema that will be compared and validated with respect to a reference schema. For example, if installing an Alfresco repository from scratch, then the newly created schema will be a target schema for comparison against the appropriate reference schema.

Performing schema dumps

Schema dumps are XML representations of the database schema.

Schema dumps can take place in two situations:

1. The dump is triggered automatically on startup due to a difference being found between the reference and actual database schema.
2. The dump is manually triggered by using a JMX client.

Each of these scenarios is described in the following sections.

Automatic dumps

Schema dumps are performed automatically on Alfresco server startup, if changes in database schema are detected.

Schema dumps are XML representations of the RDBMS schema. They should conform to the XSD: <http://www.alfresco.org/repo/db-schema/db-schema.xsd>. The XSD file is embedded in the repository.

A schema dump is performed automatically during repository server startup if there were changes made to the database schema. The Alfresco log will indicate if any dumps were performed - entries such as these will be present:

```
2012-01-30 17:46:58,517  INFO [domain.schema.SchemaBootstrap] [main]
Normalized schema dumped to file
/tomcat/temp/Alfresco/Alfresco-schema-PostgreSQLDialect-pre-upgrade-
alf_-5548956643327704619.xml.
2012-01-30 17:46:58,518  INFO [domain.schema.SchemaBootstrap] [main]
Normalized schema dumped to file
/tomcat/temp/Alfresco/Alfresco-schema-PostgreSQLDialect-pre-upgrade-
avm_-2166257481854030130.xml.
2012-01-30 17:46:58,518  INFO [domain.schema.SchemaBootstrap] [main]
Normalized schema dumped to file
/tomcat/temp/Alfresco/Alfresco-schema-PostgreSQLDialect-pre-upgrade-
jbpm_-2230905975269998715.xml.
2012-01-30 17:46:58,519  INFO [domain.schema.SchemaBootstrap] [main]
Normalized schema dumped to file
```

```
/tomcat/temp/Alfresco/Alfresco-schema-PostgreSQLDialect-pre-upgrade-
act_-8103448407472298481.xml.
```

Similar entries for the post-upgrade files will also be present.

-  The legacy tool is still included and will create dumps of its own - the log messages look similar but should not be confused with the new format dumps.
-  JBPM has been removed from Alfresco. Schema dump will ignore any JBPM tables and not treat their presence or absence as an error.

Triggering dumps by using JMX

Schema dumps can also be triggered manually by using a JMX client.

In addition to automatic dumping, dumps can be manually invoked by use of the JMX interface.

-  This is an Enterprise only feature.

The JMX category **Alfresco, DatabaseInformation, SchemaExport** contains two operations:

1. `java.util.List dumpSchemaToXML()`
2. `java.util.List dumpSchemaToXML(String prefixList)`

The first operation takes no parameters and when invoked will create three dump files one for each prefix 'alf_', 'act_', and 'avm_'. The prefix means that only tables and sequences whose names begin with the prefix will be included in the dump. Related items, such as the indexes belonging to a particular table, will be dumped regardless of name.

The second variation takes a single String parameter and is a comma-separated list of prefixes that you wish to dump. If this operation were invoked with the parameter "alf_acl_, alf_node_" for example, then two files would be created (one for each prefix). The tables dumped in the first file would include `alf_acl_change_set` and `alf_acl_member`. Tables in the second file would include `alf_node_aspects` and `alf_node_assoc`. Neither file would include `alf_locale` or `alf_permission` since they do not carry one of the supplied prefixes.

Both of these calls will result in the log showing the location of the dumped files, but they also return a `List` of path names. JConsole will helpfully display these lists in a copy/paste friendly manner.

Performing schema validation

Schema validation of schema dumps can happen either due to a schema change during repository startup, or can be triggered manually by using JMX.

Schema validation is performed with differencing and validation.

Differencing

Differencing produces similar information to that obtained by using the Unix tool `diff` against a known 'good' reference schema dump and a potentially problematic target schema dump.

Differencing produces similar information to that obtained by using the Unix tool `diff` against a known 'good' reference schema dump and a potentially problematic target schema dump. However, since the tool is designed for performing a comparison between two database schemas, rather than arbitrary text, the output is more specific about the types of difference. The types of difference that can be reported are:

- A database object appears in both the reference and target schemas, but has differences in its properties. For example if an index appears in both schemas but has a different name.

- A database object appears in the reference schema but no corresponding object has been identified in the target database.
- A database object appears in the target schema but no corresponding object has been identified in the reference database.

One advantage of the Schema Differencing Tool differencing over traditional diff tool comparisons is that an index is not recognised by the exact text appearing in a dump. Instead it is identified by which table the index belongs to, which columns are indexed and in what order. If an index has the expected name and belongs to the correct table but has the wrong columns, or the correct columns in the wrong order, then differences will be reported. Or conversely, if the correct table has an index with the correct columns in the correct order, but has the wrong index name, then this will be reported. The name can be ignored during comparisons (useful for auto-generated index names) or can be taken into account. Part of the task of producing reference schema files is to specify this behaviour using `DbValidator` objects, which are explained in the following sections.

Index related example

Supposing we have the following index defined in the reference schema:

Index name	<code>permission_id</code>
Parent table	<code>alf_access_control_entry</code>
Columns	<code>permission_id, authority_id, allowed, applies</code>

This index is specified in the schema reference file in this way (parts omitted for brevity):

```
<table name="alf_access_control_entry">
  <!-- column definitions, primary keys and foreign keys ommitted -->
  <indexes>
    <index name="permission_id" unique="true">
      <columnnames>
        <columnname>permission_id</columnname>
        <columnname>authority_id</columnname>
        <columnname>allowed</columnname>
        <columnname>applies</columnname>
      </columnnames>
    </index>
    <!-- further index definitions ommitted -->
  </indexes>
</table>
```

When the target schema's index is compared against this reference then firstly a list of candidate matches are produced. There can be more than one matching index in the target schema, in which case a redundant database object warning is issued.

Candidate matches are produced dependent on object type. For indexes:

1. If the parent table is the same and the index name is the same, then it is considered the same index.
2. If the name is different but the parent table is the same and the columns indexed are the same, and in the same order, then it is considered to be the same index.

Taking the first scenario for matching and using the `permission_id` index defined in the example, then if the `permission_id` index in the target database has the `allowed` and `applies` columns in the reverse order than is expected, the log file would notify us of validation problems:

```
2012-01-31 11:24:24,280  WARN  [domain.schema.SchemaBootstrap] [RMI TCP
Connection(11)-10.244.50.71]
Schema validation found 2 potential problems, results written to:
/tomcat/temp/Alfresco/Alfresco-PostgreSQLDialect-Validation-
alf_-5903917616348258838.txt
```

The contents of the report file would look similar to the following:

```
Difference: expected
  index .alf_access_control_entry.permission_id.columnNames[2]="allowed",
but was .alf_access_control_entry.permission_id.columnNames[2]="applies"
Difference: expected
  index .alf_access_control_entry.permission_id.columnNames[3]="applies",
but was .alf_access_control_entry.permission_id.columnNames[3]="allowed"
```

Each line shows a problem with a particular database property. Here it indicates that the property at the path `.alf_access_control_entry.permission_id.columnNames[2]` has the value `applies` but according to the reference schema should be `allowed`. The leading dot of the path can be ignored (the schema name would be present before the leading dot in the case of Oracle for example), then there is the table name `alf_access_control_entry`, the index name `permission_id` within that, and a zero-indexed list property within that. The third item (index 2) is the property at fault: `columnNames[2]`.

Similarly, the next line indicates that the next item in the column name list, `columnNames[3]`, has the value `allowed` but was expected to be `applies`.

Validation

The Schema Difference tool can use schema reference XML files to perform validation in addition to that performed by simple differencing.

Validation allows the application of more complex rules than whether there is a difference between two property values. Validation is performed by `DbValidator` objects. A chain of `DbValidator` objects is associated with each database object in the reference schema. Each of these is executed in turn and given the chance to create validation errors based on the corresponding object in the target schema.

If an index has not been given a specific name then the RDBMS will auto-generate one at creation time. This means that the reference schema cannot specify the exact name that the index in the target database will have. This would lead to schema differences being reported if it were not for the use of validators. A `NameValidator` can be specified for such an index:

```
<index name="SQL120116153558430" unique="true">
  <validators>
    <validator class="org.alfresco.util.schemacomp.validator.NameValidator">
      <properties>
        <property name="pattern">SQL[0-9]+</property>
      </properties>
    </validator>
  </validators>
  <columnnames>
    <columnname>ID</columnname>
  </columnnames>
</index>
```

This example is from a DB2 schema reference file `Schema-Reference-ALF.xml` and indicates that although in the original reference schema the index was named `SQL120116153558430` any index having the appropriate parent table, column names (and column order) is valid as long as the name matches the regular expression `SQL[0-9]+`.

When the validator is invoked, it checks that the name property of the index matches the supplied regular expression. In addition to this, the validator reports, when configured to, that it takes responsibility for the name property of the index. This stops the Schema Difference Tool from applying the differencing logic to the property. A `DbValidator` can choose to apply its validation in addition to the differencing logic by not taking sole responsibility for any properties. Conversely a validator can also take sole responsibility for an entire database object in which case no differencing logic is applied to any part of the object.

A similar problem to the auto-generated name problem is when a database object is created automatically. DB2 creates indexes on the fly under certain circumstances. It is not known whether these indexes will exist at the time the Schema Difference Tool will be run. Furthermore, the indexes are an implementation detail for DB2 rather than an explicit declaration on how the Alfresco schema should appear. To suppress such errors an `IgnoreObjectValidator` can be used - it takes responsibility for validation of the associated database object, but performs no actual validation.

Another index related example

Supposing an index is expected to be auto-generated and is defined in the schema reference file as:

```
<index name="SQL120116153558430" unique="true">
  <validators>
    <validator class="org.alfresco.util.schemacomp.validator.NameValidator">
      <properties>
        <property name="pattern">SQL[0-9]+</property>
      </properties>
    </validator>
  </validators>
  <columnnames>
    <columnname>ID</columnname>
  </columnnames>
</index>
```

Perhaps a specific unsupported upgrade path has introduced an unexpected schema change - it might not be a problem, but it is important that differences are highlighted so that a decision can be made on whether the difference represents a problem and whether a fix will need to be made. On running the Schema Difference Tool, the following might be observed in the log files:

```
2012-01-31 14:28:50,697  WARN  [domain.schema.SchemaBootstrap] [main] Schema validation found 1 potential problems, results written to: /tomcat/temp/Alfresco/Alfresco-DB2Dialect-Validation-Post-Upgrade-alf_-4048062354335481885.txt
2012-01-31 14:28:51,440  INFO  [domain.schema.SchemaBootstrap] [main] Compared database schema with reference schema (all OK):
class path resource [alfresco/dbscripts/create/
org.hibernate.dialect.DB2Dialect/Schema-Reference-AVM.xml]
2012-01-31 14:28:54,682  INFO  [domain.schema.SchemaBootstrap] [main] Compared database schema with reference schema (all OK):
class path resource [alfresco/dbscripts/create/
org.hibernate.dialect.DB2Dialect/Schema-Reference-ACT.xml]
```

The AVM and ACT database objects are all as expected, but there is a difference between the target schema and the ALF (alf_ prefixed database objects) schema reference. Looking at that file it can be seen that an index that is expected to have been auto-generated has been created with an explicit name:

```
Validation: index
ALFUSER.ALF_ACCESS_CONTROL_ENTRY.SQL120131142718040.name="idx_alf_ace_auth"
fails to match rule: name must match pattern 'SQL[0-9]+'
```

Specifically, the error report is stating that the index defined in the schema reference having the name SQL120131142718040 belonging to the table ALF_ACCESS_CONTROL_ENTRY is expected to be named in the same way: prefixed with SQL then a string of one or more digits.

In addition to automatic validation, validation can be manually invoked by use of the JMX interface.

 This is an enterprise only feature.

The JMX category **Alfresco, DatabaseInformation, SchemaValidator** contains one operation:

```
void validateSchema()
```

The operation takes no parameters and returns nothing. However, if the operation is invoked then validation will be performed and the Alfresco log will show the results:

```
2012-01-31 14:51:46,770  INFO  [domain.schema.SchemaBootstrap] [RMI TCP Connection(13)-10.244.50.71] Compared database schema  
with reference schema (all OK): class path resource  
[alfresco/dbscripts/create/org.hibernate.dialect.PostgreSQLDialect/Schema-  
Reference-ALF.xml]  
2012-01-31 14:51:47,360  INFO  [domain.schema.SchemaBootstrap] [RMI TCP Connection(13)-10.244.50.71] Compared database schema  
with reference schema (all OK): class path resource  
[alfresco/dbscripts/create/org.hibernate.dialect.PostgreSQLDialect/Schema-  
Reference-AVM.xml]  
2012-01-31 14:51:50,910  INFO  [domain.schema.SchemaBootstrap] [RMI TCP Connection(13)-10.244.50.71] Compared database schema  
with reference schema (all OK): class path resource  
[alfresco/dbscripts/create/org.hibernate.dialect.PostgreSQLDialect/Schema-  
Reference-ACT.xml]
```

In the example there were no problems found in the target schema.

Reference

Properties available in a JMX client

This information gives a summary of the properties that can be viewed and changed in a JMX client.

alfresco.authentication.allowGuestLogin

Specifies whether to allow guest access to Alfresco.

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.

ntlm.authentication.mapUnknownUserToGuest

Specifies whether unknown users are automatically logged on as the Alfresco guest user during Single Sign-On (SSO).

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.

authentication.chain

Specifies the authentication chain.

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

synchronization.import.cron

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

synchronization.loggingInterval

Specifies the number of user or group entries the synchronization subsystem will process before logging progress at INFO level. If you have the following default entry in log4j.properties:

log4j.logger.org.alfresco.repo.security.sync=info. The default is 100.

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 is successfully authenticated who does not yet exist in Alfresco. The default is true.

synchronization.synchronizeChangesOnly

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

synchronization.workerThreads

Specifies the number of worker threads. For example, 2.

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.bindto

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

cifs.disableNIO

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

cifs.disableNativeCode

When true, switches off the use of any JNI calls and JNI-based CIFS implementations.

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.enabled

Enables or disables the CIFS server.

cifs.hostannounce

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

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.datagramPort

Controls the NetBIOS datagram port. The default is 138.

cifs.netBIOSMB.namePort

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

cifs.netBIOSMB.sessionPort

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

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. The special token {localname} can be used in place of the local server's host name and a unique name can be generated by prepending/appending to it.

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.

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.urlfile.prefix

An absolute URL against which all desktop actions and URL files resolve their folder URL. The special token {localname} can be used in place of the local server's host name.

filesystem.acl.global.defaultAccessLevel

Specifies the default access level. Directly names the access control level (None, Read or Write) that applies to requests that are not in scope of any other access control. Note that it is not valid to use the value None without defining other access controls.

filesystem.acl.global.domainAccessControls

Specifies the set of access controls with domain scope. This is a composite property whose value should be a comma-separated list of domain names. To define the access level for one of the listed domains, use the property filesystem.acl.global.domainAccessControls.value.Domain.accessType.

filesystem.acl.global.protocolAccessControls

Specifies the set of access controls with protocol scope. This is a composite property whose value should be a comma-separated list of access control names.

filesystem.acl.global.userAccessControls

Specifies the set of access controls with user scope. This is a composite property whose value should be a comma-separated list of user names.

filesystem.domainMappings

Specifies the domain mapping rules that are used when the client does not supply its domain in the NTLM request.

filesystem.name

Specifies the name given to the repository file system mount exposed through the CIFS server. For example, Alfresco.

ftp.enabled

Enables or disables the FTP server.

ftp.ipv6.enabled

Enables or disables the IPv6 FTP server.

ftp.port

Specifies the port that the FTP server listens for incoming connections on. Defaults to port 21.

imap.config.home.folderPath

Specifies the default locations for the IMAP mount point. For example, Imap Home.

imap.config.home.rootPath

Specifies the default location for the IMAP mount point. For example, / \${spaces.company_home.childname}.

imap.config.home.store

Specifies the default location for the IMAP mount point. For example, \${spaces.store}.

imap.config.ignore.extraction

Defines whether or not attachments are extracted.

imap.config.server.mountPoints

Specifies the list of mount points. For example, AlfrescoIMAP.

imap.server.enabled

Enables or disables the IMAP server. This is set to false, by default.

imap.server.host
 Specifies the host for the IMAP server.

imap.server.port
 Specifies the port number for the IMAP server. For example, 143.

imap.config.server.mountPoints.value.AlfrescoIMAP.modeName
 Specifies the AlfrescoIMAP mount point access mode name. For example, MIXED.

imap.config.server.mountPoints.default.rootPath
 Specifies the root path for the mount point.

imap.config.server.mountPoints.value.AlfrescoIMAP.mountPointName
 Specifies the mount point name.

imap.config.server.mountPoints.default.store
 Specifies the default store for the mount point.

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.maxusers
 The maximum number of users who are allowed to log in or -1 if there is no limit.

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.

img.dyn
 Points to the directory containing the ImageMagick shared library (Unix) or DLL files (Windows). For example, (Windows) img.dyn=\${img.root}; (Linux) img.dyn=\${img.root}/lib.

img.exe
 Points to the ImageMagick executable file name.

img.root
 Points to the ImageMagick root directory.

JMX bean categories reference

This information explains the individual bean types exported by Alfresco. The heading for each bean type provides the JMX object naming scheme, where possible. Each section lists the individual properties for the bean type.

JMX read-only monitoring beans

JMX values (Managed Bean or MBean attributes) are exposed in the Alfresco Admin Console and with internal tools (Alfresco JMX Dump) or external tools like JConsole. The read-only beans are described here with example values.

Alfresco:Name=Authority, Object Type=org.alfresco.enterprise.repo.management.Authority

Exposes the number of groups and users known to the Authority Service.

Attribute name	Example value
NumberOfGroups	7
NumberOfUsers	4

Alfresco:Name=BatchJobs, Object

Type=org.alfresco.enterprise.repo.management.BatchMonitor and org.alfresco.enterprise.repo.management.SyncMonitorMBean

Exposes the settings for the last run batch job, including the start and end times, number of errors and synchronization settings. There are two types in this bean: FeedNotifier and Synchronization.

Attribute name	Example value
EndTime	Format: Thu Jul 03 00:00:00 BST 2014
StartTime	Format: Thu Jul 03 00:00:00 BST 2014
LastError	blank
TotalErrors	blank
SuccessfullyProcessedEntries	blank
TotalResults	blank
ProcessName	FeedNotifier
CurrentEntryID	Person admin
LastErrorEntryID	blank
PercentComplete	blank
LastErrorMessage	<null>
LastRunOnServer	127.0.0.1:8080
SyncEndTime	Format: Thu Jun 26 13:49:45 BST 2014
SyncStartTime	Format: Thu Jun 26 13:49:45 BST 2014
SynchronizationStatus	COMPLETE

Alfresco:Name=CacheStatistics, Object

Type=org.alfresco.enterprise.repo.management.CacheStatisticsMBeanImpl

Exposes statistics for the caches that are available for Alfresco. Cache name follows the pattern org.alfresco.*TransactionalCache. The following attributes are available for each cache:

Attribute name	Example value
ClearTime	NaN
Clears	Numeric
Gets	Numeric
HitMissRatio	Numeric with decimals
HitTime	Numeric with decimals
Hits	Numeric
MissTime	Numeric with decimals
Misses	Numeric
PutTime	Numeric with decimals
Puts	Numeric
RemoveTime	NaN
Removes	Numeric

Additional information about each attribute:

ClearTime

The mean time, in nanoseconds, for the cache to complete clearing (that is, to empty or drop the entire cache contents).

Clears

The number of times that the cache has been cleared (that is, emptied, or *dropped*).

Gets

The number of times that the cache has had a value requested from it. This includes cache hits, where the cache contains a value, but omits where the cache reports that there is no value corresponding to a particular key.

HitMissRatio

The hit ratio for the given cache. A value of 1.0 is the maximum indicating that every request has been honored (that is, all GET requests are *hits*) and 0.0 represents a cache that has never successfully returned a previously cached value (that is, all GET requests are *misses*).

HitTime

The mean time, in nanoseconds, for GET operations to complete, where a value has been found in the cache.

Hits

The number of times that a GET request is successful.

MissTime

The mean time, in nanoseconds, for GET requests, where a result is not returned (a *miss*).

Misses

The number of times that a GET request is not successful.

PutTime

The mean time, in nanoseconds, for inserting a value into the cache.

Puts

The number of times a value is inserted into the cache (a PUT operation).

RemoveTime

The mean time, in nanoseconds, for removing a value from cache.

Removes

The number of removal operations applied to the cache, where a value is successfully removed from the cache.

Alfresco:Name=CloudSync, Object

Type=org.alfresco.enterprise.repo.web.scripts.sync.transport.CloudSyncMonitorJMXBean

Exposes the settings for cloud synchronization, specifically pull and push request statistics.

Attribute name	Example value
AveragePullRequestTime	0
AveragePushRequestTime	0
CurrentPullRequests	0
CurrentPushRequests	0
PullRequestFailureCount	0
PullRequestSuccessCount	0
PushRequestFailureCount	0
PushRequestSuccessCount	0
TotalPullRequestCount	0
TotalPushRequestCount	0

Alfresco:Name=Cluster, Object

Types=org.alfresco.enterprise.repo.management.ClusterAdmin and org.alfresco.enterprise.repo.management.ClusterInfo

Exposes information about repository server clustering in Alfresco.

See the Alfresco Admin Console **Repository Services > Repository Server Clustering** for information about these attributes: <http://<hostname>:<portnumber>/alfresco/service/enterprise/admin/admin-clustering>.

Alfresco:Name=ConnectionPool, Object

Type=org.apache.commons.dbcp.BasicDataSource

Allows monitoring of the Apache Commons DBCP database connection pool and its configuration.

Attribute name	Example value
DefaultTransactionIsolation	-1
DriverClassName	org.postgresql.Driver
InitialSize	10
MaxActive	40
MaxIdle	-1
MaxWait	-1
MinEvictableIdleTimeMillis	1800000
MinIdle	0
NumActive	2
NumIdle	8
RemoveAbandoned	false
RemoveAbandonedTimeout	300
TestOnBorrow	false
TestOnReturn	false
TestWhileIdle	false
TimeBetweenEvictionRunsMillis	-1
Url	jdbc:postgresql://localhost:5432/alfresco
Username	alfresco
ValidationQuery	<null>

Additional information about each attribute:

DefaultTransactionIsolation

The JDBC code number for the transaction isolation level, corresponding to those in the `java.sql.Connection` class. The special value of -1 indicates that the database's default transaction isolation level is in use and this is the most common setting. For the Microsoft SQL Server JDBC driver, the special value of 4096 indicates snapshot isolation.

DriverClassName

The fully-qualified name of the JDBC driver class.

InitialSize

The number of connections opened when the pool is initialized.

MaxActive

The maximum number of connections in the pool.

MaxIdle

The maximum number of connections that are not in use kept open.

MaxWait

The maximum number of milliseconds to wait for a connection to be returned before throwing an exception (when connections are unavailable) or -1 to wait indefinitely.

MinEvictableIdleTimeMillis

The minimum number of milliseconds that a connection can sit idle before it is eligible for eviction.

MinIdle

The minimum number of connections in the pool.

NumActive

The number connections in use; a useful monitoring metric.

NumIdle

The number of connections that are not in use; another useful monitoring metric.

RemoveAbandoned

A Boolean that when true indicates that a connection is considered abandoned and eligible for removal if it has been idle longer than the RemoveAbandonedTimeout.

RemoveAbandonedTimeout

The time in seconds before an abandoned connection can be removed.

TestOnBorrow

A boolean that when true indicates that connections will be validated before being borrowed from the pool.

TestOnReturn

A boolean that when true indicates that connections will be validated before being returned to the pool.

TestWhileIdle

A boolean that when true indicates that connections will be validated while they are idle.

TimeBetweenEvictionRunsMillis

The number of milliseconds to sleep between eviction runs, when greater than zero.

Url

The JDBC URL to the database connection.

Username

The name used to authenticate with the database.

ValidationQuery

The SQL query that will be used to validate connections before returning them.

Alfresco:Name=ContentStore, Object**Type=org.alfresco.enterprise.repo.management.ContentStore**

Allows monitoring of each Alfresco content store. When `Type=FileContentStore`, the Root attribute of the name holds the file system path to the store. Specific attributes exposed are the total space (and also free space) in the content store, in bytes, and whether the store allows write operations.

Attribute name	Example value
SpaceFree	33434923008
SpaceTotal	64422408192
WriteSupported	true

Alfresco:Name=ContentTransformer, Object**Type=org.alfresco.repo.content.transform.ContentTransformer***

Expose key information about the transformation utilities relied upon by Alfresco. There are two instances:

- `Alfresco:Name=ContentTransformer, Type=Configuration`

- Alfresco:Name=ContentTransformer,Type=ImageMagick

The following properties are exposed for Type=Configuration:

Attribute name	Example value
ContextNames	[, doclib , index , webpreview , syncRule , asyncRule]
CustomPropertyNames	content.transformer.PdfBox.extensions.pdf.txt.maxSourceSizeKBytes

Attribute name	Example value
ExtensionsAndMimetypes	[3fr - image/x-raw-hasselblad , 3g2 - video/3gpp2 , 3gp - video/3gpp , acp - application/acp , aep - application/vnd.adobe.aftereffects.project , aet - application/vnd.adobe.aftereffects.template , ai - application/illustrator , aiff - audio/x-aiff , air - application/vnd.adobe.air-application-installer-package+zip , apk - application/vnd.android.package-archive , arw - image/x-raw-sony , asf - video/x-ms-asf , asnd - audio/vnd.adobe.soundbooth , au - audio/basic , avi - video/x-msvideo , avx - video/x-rad-screenplay , bcpio - application/x-bcpio , bin - application/octet-stream , bmp - image/bmp , cdf - application/x-netcdf , cer - application/x-x509-ca-cert , cgm - image/cgm , class - application/java , cpio - application/x-cpio , cr2 - image/x-raw-canon , csh - application/x-csh , css - text/css , csv - text/csv , dita - application/dita+xml , dng - image/x-raw-adobe , doc - application/msword , docm - application/vnd.ms-word.document.macroenabled.12 , docx - application/vnd.openxmlformats-officedocument.wordprocessingml.document , dotm - application/vnd.ms-word.template.macroenabled.12 , dotx - application/vnd.openxmlformats-officedocument.wordprocessingml.template , dvi - application/x-dvi , dwg - image/vnd.dwg , dwt - image/x-dwt , eml - message/rfc822 , eps - application/eps , etx - text/x-setext , fla - application/x-fla , flac - audio/x-flac , flv - video/x-flv , fm - application/framemaker , fpx - application/x-zip , gif - image/gif , gml - application/sgml , gtar - application/x-gtar , gzip - application/x-gzip , hdf - application/x-hdf , hqx - application/mac-binhex40 , html - text/html , ics - text/calendar , ief - image/ief , indd - application/x-indesign , jp2 - image/jp2 , jpg - image/jpeg , js - application/x-javascript , json - application/json , k25 - image/x-raw-kodak , key - application/vnd.apple.keynote , latex - application/x-latex , m4a - audio/mp4 , m4v - video/x-m4v , man - application/x-troff-man , md - text/x-markdown , me - application/x-troff-me , mif - application/x-mif , mov - video/quicktime , movie - video/x-sgi-movie , mp3 - audio/mpeg , mp4 - video/mp4 , mpeg2 - video/mpeg2 , mpg - video/mpeg , mpp - application/vnd.ms-project , mrw - image/x-raw-minolta , ms - application/x-troff-mes , msg - application/vnd.ms-outlook , mw - text/mediawiki , nef - image/x-raw-nikon , numbers - application/vnd.apple.numbers , oda - application/oda , odb - application/vnd.oasis.opendocument.database , odc - application/vnd.oasis.opendocument.chart , odf - application/vnd.oasis.opendocument.formula , odg - application/vnd.oasis.opendocument.graphics , odi - application/vnd.oasis.opendocument.image , odm - application/vnd.oasis.opendocument.text-master , odp - application/vnd.oasis.opendocument.presentation , ods - application/vnd.oasis.opendocument.spreadsheet , odt - application/vnd.oasis.opendocument.text , oga - audio/ogg , ogg - audio/vorbis , ogv - video/ogg , ogx - application/ogg , orf - image/x-raw-olympus , otg - application/vnd.oasis.opendocument.graphics-template , oth - application/vnd.oasis.opendocument.text-web , otp - application/vnd.oasis.opendocument.presentation-template , ots - application/vnd.oasis.opendocument.spreadsheet-template , ott - application/vnd.oasis.opendocument.text-template , pages - application/vnd.apple.pages , pbm - image/x-portable-bitmap , pdf - application/pdf , pef - image/x-raw-pentax , pgm - image/x-portable-graymap , pmd - application/pagemaker , png - image/png , pnm - image/x-portable-anymap , potm - application/vnd.ms-powerpoint.template.macroenabled.12 , potx - application/vnd.openxmlformats-officedocument.presentationml.template , ppam - application/vnd.ms-powerpoint.addin.macroenabled.12 ,

Attribute name	Example value
TestFileExtensionsAndMimetypes	<pre>[3g2 - video/3gpp2 , 3gp - video/3gpp , acp - application/acp , asf - video/x-ms-asf , avi - video/x-msvideo , bmp - image/bmp , doc - application/msword , docx - application/vnd.openxmlformats-officedocument.wordprocessingml.document , eml - message/rfc822 , eps - application/eps , flv - video/x-flv , gif - image/gif , html - text/html , jpg - image/jpeg , m4a - audio/mp4 , m4v - video/x-m4v , mov - video/quicktime , mp3 - audio/mpeg , mp4 - video/mp4 , mpg - video/mpeg , msg - application/vnd.ms-outlook , odf - application/vnd.oasis.opendocument.formula , odg - application/vnd.oasis.opendocument.graphics , odp - application/vnd.oasis.opendocument.presentation , ods - application/vnd.oasis.opendocument.spreadsheet , odt - application/vnd.oasis.opendocument.text , ogg - audio/vorbis , ogv - video/ogg , otg - application/vnd.oasis.opendocument.graphics-template , otp - application/vnd.oasis.opendocument.presentation-template , ots - application/vnd.oasis.opendocument.spreadsheet-template , ott - application/vnd.oasis.opendocument.text-template , pdf - application/pdf , png - image/png , ppt - application/vnd.ms-powerpoint , pptx - application/vnd.openxmlformats-officedocument.presentationml.presentation , sda - application/vnd.stardivision.draw , sdc - application/vnd.stardivision.calc , sdd - application/vnd.stardivision.impress , sdw - application/vnd.stardivision.writer , smf - application/vnd.stardivision.math , sxc - application/vnd.sun.xml.calc , sxd - application/vnd.sun.xml.draw , sxi - application/vnd.sun.xml.impress , sxw - application/vnd.sun.xml.writer , tar - application/x-tar , tiff - image/tiff , txt - text/plain , vsd - application/vnd.visio , webm - video/webm , wma - audio/x-ms-wma , wmv - video/x-ms-wmv , wpd - application/wordperfect , xls - application/vnd.ms-excel , xlsx - application/vnd.openxmlformats-officedocument.spreadsheetml.sheet , xml - text/xml , zip - application/zip]</pre>
TransformerNames	<pre>[Archive , BinaryPassThrough , HtmlParser , ImageMagick , JodConverter , JodConverter.2Pdf , JodConverter.Html2Pdf , MediaWikiParser , OOXML , OOXMLThumbnail , Office , OutlookMsg , PdfBox , PdfBox.TextToPdf , Poi , RFC822 , StringExtractor , TextMining , TikaAuto , complex.Any.Image , complex.JodConverter.Image , complex.JodConverter.PdfBox , complex.OutlookMsg2Image , complex.PDF.Image , complex.Text.Image , complex.iWorks.Image , double.ImageMagick , iWorksQuicklooks]</pre>

The following properties are exposed for Type=ImageMagick. Available indicates whether the utility is installed correctly and was found when Alfresco started up, and VersionString indicates the version information returned by the utility:

Attribute name	Example value
Available	true
VersionString	Version: ImageMagick 6.8.6-6 2013-07-16 Q8 http://www.imagemagick.org

**Alfresco:Name=CustomModel,
org.alfresco.enterprise.repo.management.CustomModelsInfoMBeanImpl**

Exposes information about the number of active aspects, models, and types.

Attribute name	Example value
CustomModelsStatistics	[]

Alfresco:Name=DatabaseInformation, org.alfresco.enterprise.repo.management.Database

Exposes metadata about the database itself, including database, driver and JDBC information, and the name used to authenticate with the database.

Attribute name	Example value
DatabaseMajorVersion	9
DatabaseMinorVersion	2
DatabaseProductName	PostgreSQL
DatabaseProductVersion	9.2.4
DriverMajorVersion	9
DriverMinorVersion	0
DriverName	PostgreSQL Native Driver
DriverVersion	PostgreSQL 9.0 JDBC4 (build 802)
JDBCMajorVersion	4
JDBCMajorVersion	0
StoresLowerCaselIdentifiers	true
StoresLowerCaseQuotedIdentifiers	false
StoresMixedCaselIdentifiers	false
StoresMixedCaseQuotedIdentifiers	false
StoresUpperCaselIdentifiers	false
StoresUpperCaseQuotedIdentifiers	false
URL	jdbc:postgresql://localhost:5432/alfresco
UserName	alfresco

**Alfresco:Name=Encryption,
org.alfresco.enterprise.encryption.management.AlfrescoKeyStoreBean**

Exposes information about the location and backup location of encryption methods used by Alfresco. There are three types of keystore:

- Alfresco:Name=Encryption,Type=Key Store
- Alfresco:Name=ContentTransformer,Type=SSL Key Store
- Alfresco:Name=ContentTransformer,Type=SSL Trust Store

The following properties are exposed for Type=Key Store:

Attribute name	Example value
BackupLocation	C:/Alfresco/alf_data/keystore/backup-keystore
Location	C:/Alfresco/alf_data/keystore/keystore

The following properties are exposed for Type=SSL Key Store:

Attribute name	Example value
BackupLocation	<not readable>

Attribute name	Example value
Location	C:/Alfresco/alf_data/keystore/ssl.keystore

The following properties are exposed for Type=SSL Trust Store:

Attribute name	Example value
BackupLocation	<not readable>
Location	C:/Alfresco/alf_data/keystore/ssl.truststore

Alfresco:Name=FileServerConfig, Object Type=com.sun.proxy.\$Proxy108

Allows management and monitoring of the CIFS and FTP servers configured in Alfresco. See the Alfresco Admin Console **Virtual File Systems - File Servers** for information about these attributes: `http://<hostname>:<portnumber>/alfresco/service/enterprise/admin/admin-fileservers`.

Alfresco:Name=GlobalProperties, Object Type=org.alfresco.enterprise.repo.management.PropertiesDynamicMBean

Exposes the default configuration settings for Alfresco that are present in the `alfresco-global.properties` file.

Attribute name	Example value
;	default filesystem target configuration
:Where	should the root of the web project be stored, by default /www/avm_webapps
V2.1-A.fixes.to.schema	0
activities.feed.fetchBatchSize	150
activities.feed.generator.jsonFormatOnly	true
activities.feed.max.ageMins	44640
activities.feed.max.idRange	1000000
activities.feed.max.size	200
activities.feedNotifier.batchSize	200
activities.feedNotifier.numThreads	2
alfresco.authentication.gateway.bufferSize	2048
alfresco.authentication.gateway.connectTimeout	10000
alfresco.authentication.gateway.host	blank
alfresco.authentication.gateway.httpConnectionStalecheck	true
alfresco.authentication.gateway.httpTcpNodelay	true
alfresco.authentication.gateway.inboundHeaders	X-Alfresco-Authenticator-Key,X-Alfresco-Remote-User
alfresco.authentication.gateway.outboundHeaders	Authorization,key
alfresco.authentication.gateway.port	443
alfresco.authentication.gateway.prefixUrl	/publicapi
alfresco.authentication.gateway.protocol	https
alfresco.authentication.gateway.readTimeout	120000
alfresco.cluster.enabled	true
alfresco.cluster.hostname	\${localname}
alfresco.cluster.interface	blank
alfresco.cluster.max.init.retries	50

Attribute name	Example value
alfresco.cluster.nodetype	"Repository server"
alfresco.clusterCheck.timeout	4000
alfresco.context	alfresco
alfresco.hazelcast.autoinc.port	false
alfresco.hazelcast.configLocation	classpath:alfresco/hazelcast/ hazelcast-tcp.xml
alfresco.hazelcast.ec2.accesskey	my-access-key
alfresco.hazelcast.ec2.region	us-east-1
alfresco.hazelcast.ec2.secretkey	my-secret-key
alfresco.hazelcast.ec2.securitygroup	blank
alfresco.hazelcast.ec2.tagkey	type
alfresco.hazelcast.ec2.tagvalue	hz-nodes
alfresco.hazelcast.mancenter.enabled	false
alfresco.hazelcast.mancenter.url	http://localhost:8080/mancenter
alfresco.hazelcast.password	alfrescocluster
alfresco.hazelcast.port	5701
alfresco.hazelcast.tcp.config	<members></members>
alfresco.host	127.0.0.1
alfresco.port	8080
alfresco.protocol	http
alfresco.rmi.services.external.host	0.0.0.0
alfresco.rmi.services.port	50500
alfresco.rmi.services.retries	4
alfresco_user_store.adminpassword	209c6174da490caeb422f3fa5a7ae634
alfresco_user_store.adminusername	admin
alfresco_user_store.guestusername	guest
alfresco_user_store.store	user://alfrescoUserStore
alfresco_user_store.system_container.childname	sys:system
alfresco_user_store.user_container.childname	sys:people
audit.alfresco-access.enabled	false
audit.alfresco-access.sub-actions.enabled	false
audit.cmischangelog.enabled	false
audit.config.strict	false
audit.dod5015.enabled	false
audit.enabled	true
audit.filter.alfresco-access.default.enabled	false
audit.filter.alfresco-access.transaction.path	~/sys:archivedItem;~/ver:;.*
audit.filter.alfresco-access.transaction.type	cm:folder;cm:content;st:site
audit.filter.alfresco-access.transaction.user	~System;~null;.*
audit.tagging.enabled	true
authentication.chain	alfrescoNtlm1:alfrescoNtlm
authentication.ticket.expiryMode	AFTER_INACTIVITY
authentication.ticket.ticketsExpire	true
authentication.ticket.useSingleTicketPerUser	true
authentication.ticket.validDuration	PT1H
authority.useBridgeTable	true
bulkImport.batch batchSize	20

Attribute name	Example value
bulkImport.batch.numThreads	4
cache.aclEntitySharedCache.backup-count	1
cache.aclEntitySharedCache.cluster.type	fully-distributed
cache.aclEntitySharedCache.eviction-percentage	25
cache.aclEntitySharedCache.eviction-policy	LRU
cache.aclEntitySharedCache.maxIdleSeconds	0
cache.aclEntitySharedCache.maxItems	50000
cache.aclEntitySharedCache.merge-policy	hz.ADD_NEW_ENTRY
cache.aclEntitySharedCache.timeToLiveSeconds	0
cache.aclEntitySharedCache.tx.maxItems	50000
cache.aclSharedCache.backup-count	1
cache.aclSharedCache.cluster.type	fully-distributed
cache.aclSharedCache.eviction-percentage	25
cache.aclSharedCache.eviction-policy	LRU
cache.aclSharedCache.maxIdleSeconds	0
cache.aclSharedCache.maxItems	50000
cache.aclSharedCache.merge-policy	hz.ADD_NEW_ENTRY
cache.aclSharedCache.timeToLiveSeconds	0
cache.aclSharedCache.tx.maxItems	20000
cache.authenticationSharedCache.backup-count	1
cache.authenticationSharedCache.cluster.type	fully-distributed
cache.authenticationSharedCache.eviction-percentage	25
cache.authenticationSharedCache.eviction-policy	LRU
cache.authenticationSharedCache.maxIdleSeconds	0
cache.authenticationSharedCache.maxItems	5000
cache.authenticationSharedCache.merge-policy	hz.ADD_NEW_ENTRY
cache.authenticationSharedCache.timeToLiveSeconds	0
cache.authenticationSharedCache.tx.maxItems	100
cache.authorityEntitySharedCache.tx.maxItems	50000
cache.authoritySharedCache.backup-count	1
cache.authoritySharedCache.cluster.type	invalidating
cache.authoritySharedCache.eviction-percentage	25
cache.authoritySharedCache.eviction-policy	LRU
cache.authoritySharedCache.maxIdleSeconds	0
cache.authoritySharedCache.maxItems	10000
cache.authoritySharedCache.merge-policy	hz.ADD_NEW_ENTRY
cache.authoritySharedCache.timeToLiveSeconds	0
cache.authoritySharedCache.tx.maxItems	10000
cache.authorityToChildAuthoritySharedCache.backup-count	1
cache.authorityToChildAuthoritySharedCache.cluster.type	invalidating
cache.authorityToChildAuthoritySharedCache.eviction-percentage	25
cache.authorityToChildAuthoritySharedCache.eviction-policy	LRU
cache.authorityToChildAuthoritySharedCache.maxIdleSeconds	0
cache.authorityToChildAuthoritySharedCache.maxItems	40000

Attribute name	Example value
cache.authorityToChildAuthoritySharedCache.merge-policy	hz . ADD_NEW_ENTRY
cache.authorityToChildAuthoritySharedCache.timeToLiveSeconds	0
cache.authorityToChildAuthoritySharedCache.tx.maxItems	40000
cache.avmEntitySharedCache.backup-count	1
cache.avmEntitySharedCache.cluster.type	fully-distributed
cache.avmEntitySharedCache.eviction-percentage	25
cache.avmEntitySharedCache.eviction-policy	LRU
cache.avmEntitySharedCache.maxIdleSeconds	0
cache.avmEntitySharedCache.maxItems	5000
cache.avmEntitySharedCache.merge-policy	hz . ADD_NEW_ENTRY
cache.avmEntitySharedCache.timeToLiveSeconds	0
cache.avmEntitySharedCache.tx.maxItems	5000
cache.avmNodeAspectsSharedCache.backup-count	1
cache.avmNodeAspectsSharedCache.cluster.type	fully-distributed
cache.avmNodeAspectsSharedCache.eviction-percentage	25
cache.avmNodeAspectsSharedCache.eviction-policy	LRU
cache.avmNodeAspectsSharedCache.maxIdleSeconds	0
cache.avmNodeAspectsSharedCache.maxItems	5000
cache.avmNodeAspectsSharedCache.merge-policy	hz . ADD_NEW_ENTRY
cache.avmNodeAspectsSharedCache.timeToLiveSeconds	0
cache.avmNodeAspectsSharedCache.tx.maxItems	5000
cache.avmNodeSharedCache.backup-count	1
cache.avmNodeSharedCache.cluster.type	fully-distributed
cache.avmNodeSharedCache.eviction-percentage	25
cache.avmNodeSharedCache.eviction-policy	LRU
cache.avmNodeSharedCache.maxIdleSeconds	0
cache.avmNodeSharedCache.maxItems	5000
cache.avmNodeSharedCache.merge-policy	hz . ADD_NEW_ENTRY
cache.avmNodeSharedCache.timeToLiveSeconds	0
cache.avmNodeSharedCache.tx.maxItems	5000
cache.avmStoreSharedCache.tx.maxItems	1000
cache.avmVersionRootEntityCache.tx.maxItems1	100
cache.avmVersionRootEntitySharedCache.backup-count	1
cache.avmVersionRootEntitySharedCache.cluster.type	fully-distributed
cache.avmVersionRootEntitySharedCache.eviction-percentage	25
cache.avmVersionRootEntitySharedCache.eviction-policy	LRU
cache.avmVersionRootEntitySharedCache.maxIdleSeconds	0
cache.avmVersionRootEntitySharedCache.maxItems	1000
cache.avmVersionRootEntitySharedCache.merge-policy	hz . ADD_NEW_ENTRY
cache.avmVersionRootEntitySharedCache.timeToLiveSeconds	0
cache.cachingContentStoreCache.backup-count	1
cache.cachingContentStoreCache.cluster.type	local
cache.cachingContentStoreCache.eviction-percentage	25
cache.cachingContentStoreCache.eviction-policy	LRU

Attribute name	Example value
cache.cachingContentStoreCache.maxIdleSeconds	86400
cache.cachingContentStoreCache.maxItems	5000
cache.cachingContentStoreCache.merge-policy	hz.ADD_NEW_ENTRY
cache.cachingContentStoreCache.timeToLiveSeconds	0
cache.caveatConfigCache.backup-count	1
cache.caveatConfigCache.cluster.type	invalidating
cache.caveatConfigCache.eviction-percentage	25
cache.caveatConfigCache.eviction-policy	LRU
cache.caveatConfigCache.maxIdleSeconds	0
cache.caveatConfigCache.maxItems	5000
cache.caveatConfigCache.merge-policy	hz.ADD_NEW_ENTRY
cache.caveatConfigCache.timeToLiveSeconds	0
cache.caveatConfigCache.tx.maxItems	100
cache.compiledModelsSharedCache.backup-count	1
cache.compiledModelsSharedCache.cluster.type	invalidating
cache.compiledModelsSharedCache.eviction-percentage	25
cache.compiledModelsSharedCache.eviction-policy	LRU
cache.compiledModelsSharedCache.maxIdleSeconds	0
cache.compiledModelsSharedCache.maxItems	1000
cache.compiledModelsSharedCache.merge-policy	hz.ADD_NEW_ENTRY
cache.compiledModelsSharedCache.timeToLiveSeconds	0
cache.contentDataSharedCache.backup-count	1
cache.contentDataSharedCache.cluster.type	fully-distributed
cache.contentDataSharedCache.eviction-percentage	25
cache.contentDataSharedCache.eviction-policy	LRU
cache.contentDataSharedCache.maxIdleSeconds	0
cache.contentDataSharedCache.maxItems	130000
cache.contentDataSharedCache.merge-policy	hz.ADD_NEW_ENTRY
cache.contentDataSharedCache.timeToLiveSeconds	0
cache.contentDataSharedCache.tx.maxItems	65000
cache.contentDiskDriver fileInfoCache.backup-count	1
cache.contentDiskDriver fileInfoCache.cluster.type	local
cache.contentDiskDriver fileInfoCache.eviction-percentage	25
cache.contentDiskDriver fileInfoCache.eviction-policy	LRU
cache.contentDiskDriver fileInfoCache.maxIdleSeconds	0
cache.contentDiskDriver fileInfoCache.maxItems	1000
cache.contentDiskDriver fileInfoCache.merge-policy	hz.ADD_NEW_ENTRY
cache.contentDiskDriver fileInfoCache.timeToLiveSeconds	0
cache.executingActionsCache.backup-count	1
cache.executingActionsCache.cluster.type	fully-distributed
cache.executingActionsCache.eviction-percentage	25
cache.executingActionsCache.eviction-policy	LRU
cache.executingActionsCache.maxIdleSeconds	0
cache.executingActionsCache.maxItems	1000
cache.executingActionsCache.merge-policy	hz.ADD_NEW_ENTRY
cache.executingActionsCache.timeToLiveSeconds	0
cache.globalConfigSharedCache.backup-count	1

Attribute name	Example value
cache.globalConfigSharedCache.cluster.type	invalidating
cache.globalConfigSharedCache.eviction-percentage	25
cache.globalConfigSharedCache.eviction-policy	LRU
cache.globalConfigSharedCache.maxIdleSeconds	0
cache.globalConfigSharedCache.maxItems	1000
cache.globalConfigSharedCache.merge-policy	hz.ADD_NEW_ENTRY
cache.globalConfigSharedCache.timeToLiveSeconds	0
cache imapMessageSharedCache.backup-count	1
cache imapMessageSharedCache.cluster.type	invalidating
cache imapMessageSharedCache.eviction-percentage	25
cache imapMessageSharedCache.eviction-policy	LRU
cache imapMessageSharedCache.maxIdleSeconds	0
cache imapMessageSharedCache.maxItems	2000
cache imapMessageSharedCache.merge-policy	hz.ADD_NEW_ENTRY
cache imapMessageSharedCache.timeToLiveSeconds	0
cache imapMessageSharedCache.tx.maxItems	1000
cache immutableEntitySharedCache.backup-count	1
cache immutableEntitySharedCache.cluster.type	invalidating
cache immutableEntitySharedCache.eviction-percentage	25
cache immutableEntitySharedCache.eviction-policy	LRU
cache immutableEntitySharedCache.maxIdleSeconds	0
cache immutableEntitySharedCache.maxItems	50000
cache immutableEntitySharedCache.merge-policy	hz.ADD_NEW_ENTRY
cache immutableEntitySharedCache.timeToLiveSeconds	0
cache immutableEntitySharedCache.tx.maxItems	10000
cache immutableSingletonSharedCache.backup-count	1
cache immutableSingletonSharedCache.cluster.type	invalidating
cache immutableSingletonSharedCache.eviction-percentage	25
cache immutableSingletonSharedCache.eviction-policy	LRU
cache immutableSingletonSharedCache.maxIdleSeconds	0
cache immutableSingletonSharedCache.maxItems	12000
cache immutableSingletonSharedCache.merge-policy	hz.ADD_NEW_ENTRY
cache immutableSingletonSharedCache.timeToLiveSeconds	0
cache immutableSingletonSharedCache.tx.maxItems	12000
cache loadedResourceBundlesSharedCache.backup-count	1
cache loadedResourceBundlesSharedCache.cluster.type	fully-distributed
cache loadedResourceBundlesSharedCache.eviction-percentage	25
cache loadedResourceBundlesSharedCache.eviction-policy	LRU
cache loadedResourceBundlesSharedCache.maxIdleSeconds	0
cache loadedResourceBundlesSharedCache.maxItems	1000
cache loadedResourceBundlesSharedCache.merge-policy	hz.ADD_NEW_ENTRY
cache loadedResourceBundlesSharedCache.timeToLiveSeconds	0
cache loadedResourceBundlesSharedCache.tx.maxItems	1000

Attribute name	Example value
cache.messagesSharedCache.backup-count	1
cache.messagesSharedCache.cluster.type	fully-distributed
cache.messagesSharedCache.eviction-percentage	25
cache.messagesSharedCache.eviction-policy	LRU
cache.messagesSharedCache.maxIdleSeconds	0
cache.messagesSharedCache.maxItems	1000
cache.messagesSharedCache.merge-policy	hz . ADD_NEW_ENTRY
cache.messagesSharedCache.timeToLiveSeconds	0
cache.messagesSharedCache.tx.maxItems	1000
cache.node.allRootNodesSharedCache.backup-count	1
cache.node.allRootNodesSharedCache.cluster.type	invalidating
cache.node.allRootNodesSharedCache.eviction-percentage	25
cache.node.allRootNodesSharedCache.eviction-policy	LRU
cache.node.allRootNodesSharedCache.maxIdleSeconds	0
cache.node.allRootNodesSharedCache.maxItems	1000
cache.node.allRootNodesSharedCache.merge-policy	hz . ADD_NEW_ENTRY
cache.node.allRootNodesSharedCache.timeToLiveSeconds	0
cache.node.allRootNodesSharedCache.tx.maxItems	500
cache.node.aspectsSharedCache.backup-count	1
cache.node.aspectsSharedCache.cluster.type	local
cache.node.aspectsSharedCache.eviction-percentage	25
cache.node.aspectsSharedCache.eviction-policy	LRU
cache.node.aspectsSharedCache.maxIdleSeconds	0
cache.node.aspectsSharedCache.maxItems	130000
cache.node.aspectsSharedCache.merge-policy	hz . ADD_NEW_ENTRY
cache.node.aspectsSharedCache.timeToLiveSeconds	0
cache.node.aspectsSharedCache.tx.maxItems	65000
cache.node.childByNameSharedCache.backup-count	1
cache.node.childByNameSharedCache.cluster.type	local
cache.node.childByNameSharedCache.eviction-percentage	25
cache.node.childByNameSharedCache.eviction-policy	LRU
cache.node.childByNameSharedCache.maxIdleSeconds	0
cache.node.childByNameSharedCache.maxItems	130000
cache.node.childByNameSharedCache.merge-policy	hz . ADD_NEW_ENTRY
cache.node.childByNameSharedCache.timeToLiveSeconds	0
cache.node.childByNameSharedCache.tx.maxItems	65000
cache.node.nodesSharedCache.backup-count	1
cache.node.nodesSharedCache.cluster.type	invalidating
cache.node.nodesSharedCache.eviction-percentage	25
cache.node.nodesSharedCache.eviction-policy	LRU
cache.node.nodesSharedCache.maxIdleSeconds	0
cache.node.nodesSharedCache.maxItems	250000
cache.node.nodesSharedCache.merge-policy	hz . ADD_NEW_ENTRY
cache.node.nodesSharedCache.timeToLiveSeconds	0
cache.node.nodesSharedCache.tx.maxItems	125000
cache.node.parentAssocsSharedCache.backup-count	1
cache.node.parentAssocsSharedCache.cluster.type	fully-distributed

Attribute name	Example value
cache.node.parentAssocsSharedCache.eviction-percentage	25
cache.node.parentAssocsSharedCache.eviction-policy	LRU
cache.node.parentAssocsSharedCache.maxIdleSeconds	0
cache.node.parentAssocsSharedCache.maxItems	130000
cache.node.parentAssocsSharedCache.merge-policy	hz . ADD_NEW_ENTRY
cache.node.parentAssocsSharedCache.timeToLiveSeconds	0
cache.node.propertiesSharedCache.backup-count	1
cache.node.propertiesSharedCache.cluster.type	local
cache.node.propertiesSharedCache.eviction-percentage	25
cache.node.propertiesSharedCache.eviction-policy	LRU
cache.node.propertiesSharedCache.maxIdleSeconds	0
cache.node.propertiesSharedCache.maxItems	130000
cache.node.propertiesSharedCache.merge-policy	hz . ADD_NEW_ENTRY
cache.node.propertiesSharedCache.timeToLiveSeconds	0
cache.node.propertiesSharedCache.tx.maxItems	65000
cache.node.rootNodesSharedCache.backup-count	1
cache.node.rootNodesSharedCache.cluster.type	invalidating
cache.node.rootNodesSharedCache.eviction-percentage	25
cache.node.rootNodesSharedCache.eviction-policy	LRU
cache.node.rootNodesSharedCache.maxIdleSeconds	0
cache.node.rootNodesSharedCache.maxItems	1000
cache.node.rootNodesSharedCache.merge-policy	hz . ADD_NEW_ENTRY
cache.node.rootNodesSharedCache.timeToLiveSeconds	0
cache.node.rootNodesSharedCache.tx.maxItems	1000
cache.nodeOwnerSharedCache.backup-count	1
cache.nodeOwnerSharedCache.cluster.type	fully-distributed
cache.nodeOwnerSharedCache.eviction-percentage	25
cache.nodeOwnerSharedCache.eviction-policy	LRU
cache.nodeOwnerSharedCache.maxIdleSeconds	0
cache.nodeOwnerSharedCache.maxItems	40000
cache.nodeOwnerSharedCache.merge-policy	hz . ADD_NEW_ENTRY
cache.nodeOwnerSharedCache.timeToLiveSeconds	0
cache.nodeOwnerSharedCache.tx.maxItems	40000
cache.nodeRulesSharedCache.tx.maxItems	2000
cache.permissionEntitySharedCache.tx.maxItems	50000
cache.permissionsAccessSharedCache.backup-count	1
cache.permissionsAccessSharedCache.cluster.type	fully-distributed
cache.permissionsAccessSharedCache.eviction-percentage	25
cache.permissionsAccessSharedCache.eviction-policy	LRU
cache.permissionsAccessSharedCache.maxIdleSeconds	0
cache.permissionsAccessSharedCache.maxItems	50000
cache.permissionsAccessSharedCache.merge-policy	hz . ADD_NEW_ENTRY
cache.permissionsAccessSharedCache.timeToLiveSeconds	0
cache.permissionsAccessSharedCache.tx.maxItems	10000
cache.personSharedCache.backup-count	1
cache.personSharedCache.cluster.type	fully-distributed
cache.personSharedCache.eviction-percentage	25

Attribute name	Example value
cache.personSharedCache.evacuation-policy	LRU
cache.personSharedCache.maxIdleSeconds	0
cache.personSharedCache.maxItems	1000
cache.personSharedCache.merge-policy	hz.ADD_NEW_ENTRY
cache.personSharedCache.timeToLiveSeconds	0
cache.personSharedCache.tx.maxItems	1000
cache.propertyUniqueContextSharedCache.backup-count	1
cache.propertyUniqueContextSharedCache.cluster.type	invalidating
cache.propertyUniqueContextSharedCache.evacuation-percentage	25
cache.propertyUniqueContextSharedCache.evacuation-policy	LRU
cache.propertyUniqueContextSharedCache.maxIdleSeconds	0
cache.propertyUniqueContextSharedCache.maxItems	10000
cache.propertyUniqueContextSharedCache.merge-policy	hz.ADD_NEW_ENTRY
cache.propertyUniqueContextSharedCache.timeToLiveSeconds	0
cache.propertyUniqueContextSharedCache.tx.maxItems	10000
cache.propertyValueCache.backup-count	1
cache.propertyValueCache.cluster.type	invalidating
cache.propertyValueCache.evacuation-percentage	25
cache.propertyValueCache.evacuation-policy	LRU
cache.propertyValueCache.maxIdleSeconds	0
cache.propertyValueCache.maxItems	10000
cache.propertyValueCache.merge-policy	hz.ADD_NEW_ENTRY
cache.propertyValueCache.timeToLiveSeconds	0
cache.propertyValueCache.tx.maxItems	1000
cache.publicapi.webScriptsRegistryCache.backup-count	1
cache.publicapi.webScriptsRegistryCache.cluster.type	invalidating
cache.publicapi.webScriptsRegistryCache.evacuation-percentage	25
cache.publicapi.webScriptsRegistryCache.evacuation-policy	LRU
cache.publicapi.webScriptsRegistryCache.maxIdleSeconds	0
cache.publicapi.webScriptsRegistryCache.maxItems	1000
cache.publicapi.webScriptsRegistryCache.merge-policy	hz.ADD_NEW_ENTRY
cache.publicapi.webScriptsRegistryCache.timeToLiveSeconds	0
cache.readersDeniedSharedCache.backup-count	1
cache.readersDeniedSharedCache.cluster.type	fully-distributed
cache.readersDeniedSharedCache.evacuation-percentage	25
cache.readersDeniedSharedCache.evacuation-policy	LRU
cache.readersDeniedSharedCache.maxIdleSeconds	0
cache.readersDeniedSharedCache.maxItems	10000
cache.readersDeniedSharedCache.merge-policy	hz.ADD_NEW_ENTRY
cache.readersDeniedSharedCache.timeToLiveSeconds	0
cache.readersDeniedSharedCache.tx.maxItems	10000
cache.readersSharedCache.backup-count	1
cache.readersSharedCache.cluster.type	fully-distributed

Attribute name	Example value
cache.readersSharedCache.eviction-percentage	25
cache.readersSharedCache.eviction-policy	LRU
cache.readersSharedCache.maxIdleSeconds	0
cache.readersSharedCache.maxItems	10000
cache.readersSharedCache.merge-policy	hz.ADD_NEW_ENTRY
cache.readersSharedCache.timeToLiveSeconds	0
cache.readersSharedCache.tx.maxItems	10000
cache.remoteAlfrescoTicketService.ticketsCache.backup-count	1
cache.remoteAlfrescoTicketService.ticketsCache.cluster.type	fully-distributed
cache.remoteAlfrescoTicketService.ticketsCache.eviction-percentage	25
cache.remoteAlfrescoTicketService.ticketsCache.eviction-policy	LRU
cache.remoteAlfrescoTicketService.ticketsCache.maxIdleSeconds	0
cache.remoteAlfrescoTicketService.ticketsCache.maxItems	1000
cache.remoteAlfrescoTicketService.ticketsCache.merge-policy	hz.ADD_NEW_ENTRY
cache.remoteAlfrescoTicketService.ticketsCache.timeToLiveSeconds	0
cache.resourceBundleBaseNamesSharedCache.backup-count	1
cache.resourceBundleBaseNamesSharedCache.cluster.type	fully-distributed
cache.resourceBundleBaseNamesSharedCache.eviction-percentage	25
cache.resourceBundleBaseNamesSharedCache.eviction-policy	LRU
cache.resourceBundleBaseNamesSharedCache.maxIdleSeconds	
cache.resourceBundleBaseNamesSharedCache.maxItems	1000
cache.resourceBundleBaseNamesSharedCache.merge-policy	hz.ADD_NEW_ENTRY
cache.resourceBundleBaseNamesSharedCache.timeToLiveSeconds	
cache.resourceBundleBaseNamesSharedCache.tx.maxItems	1000
cache.routingContentStoreSharedCache.backup-count	1
cache.routingContentStoreSharedCache.cluster.type	local
cache.routingContentStoreSharedCache.eviction-percentage	25
cache.routingContentStoreSharedCache.eviction-policy	LRU
cache.routingContentStoreSharedCache.maxIdleSeconds	0
cache.routingContentStoreSharedCache.maxItems	10000
cache.routingContentStoreSharedCache.merge-policy	hz.ADD_NEW_ENTRY
cache.routingContentStoreSharedCache.timeToLiveSeconds	0
cache.routingContentStoreSharedCache.tx.maxItems	10000
cache.samITrustEngineSharedCache.backup-count	1
cache.samITrustEngineSharedCache.cluster.type	fully-distributed
cache.samITrustEngineSharedCache.eviction-percentage	25
cache.samITrustEngineSharedCache.eviction-policy	LRU

Attribute name	Example value
cache.samlTrustEngineSharedCache.maxIdleSeconds	0
cache.samlTrustEngineSharedCache.maxItems	5000
cache.samlTrustEngineSharedCache.merge-policy	hz.ADD_NEW_ENTRY
cache.samlTrustEngineSharedCache.timeToLiveSeconds	0
cache.samlTrustEngineSharedCache.tx.maxItems	5000
cache.siteNodeRefSharedCache.backup-count	1
cache.siteNodeRefSharedCache.cluster.type	fully-distributed
cache.siteNodeRefSharedCache.eviction-percentage	25
cache.siteNodeRefSharedCache.eviction-policy	LRU
cache.siteNodeRefSharedCache.maxIdleSeconds	0
cache.siteNodeRefSharedCache.maxItems	5000
cache.siteNodeRefSharedCache.merge-policy	hz.ADD_NEW_ENTRY
cache.siteNodeRefSharedCache.timeToLiveSeconds	0
cache.siteNodeRefSharedCache.tx.maxItems	5000
cache.siteNodeRefSharedCache.merge-policy	hz.ADD_NEW_ENTRY
cache.siteNodeRefSharedCache.timeToLiveSeconds	0
cache.siteNodeRefSharedCache.tx.maxItems	5000
cache.tagscopeSummarySharedCache.backup-count	1
cache.tagscopeSummarySharedCache.cluster.type	fully-distributed
cache.tagscopeSummarySharedCache.eviction-percentage	25
cache.tagscopeSummarySharedCache.eviction-policy	LRU
cache.tagscopeSummarySharedCache.maxIdleSeconds	0
cache.tagscopeSummarySharedCache.maxItems	1000
cache.tagscopeSummarySharedCache.merge-policy	hz.ADD_NEW_ENTRY
cache.tagscopeSummarySharedCache.timeToLiveSeconds	0
cache.tagscopeSummarySharedCache.tx.maxItems	1000
cache.tenantEntitySharedCache.backup-count	1
cache.tenantEntitySharedCache.cluster.type	fully-distributed
cache.tenantEntitySharedCache.eviction-percentage	25
cache.tenantEntitySharedCache.eviction-policy	LRU
cache.tenantEntitySharedCache.maxIdleSeconds	0
cache.tenantEntitySharedCache.maxItems	1000
cache.tenantEntitySharedCache.merge-policy	hz.ADD_NEW_ENTRY
cache.tenantEntitySharedCache.timeToLiveSeconds	0
cache.tenantEntitySharedCache.tx.maxItems	1000
cache.ticketsCache.backup-count	1
cache.ticketsCache.cluster.type	fully-distributed
cache.ticketsCache.eviction-percentage	25
cache.ticketsCache.eviction-policy	LRU
cache.ticketsCache.maxIdleSeconds	0
cache.ticketsCache.maxItems	1000
cache.ticketsCache.merge-policy	hz.ADD_NEW_ENTRY
cache.ticketsCache.timeToLiveSeconds	0
cache.userToAuthoritySharedCache.backup-count	1
cache.userToAuthoritySharedCache.cluster.type	invalidating
cache.userToAuthoritySharedCache.eviction-percentage	25
cache.userToAuthoritySharedCache.eviction-policy	LRU

Attribute name	Example value
cache.userToAuthoritySharedCache.maxIdleSeconds	0
cache.userToAuthoritySharedCache.maxItems	5000
cache.userToAuthoritySharedCache.merge-policy	hz.ADD_NEW_ENTRY
cache.userToAuthoritySharedCache.timeToLiveSeconds	0
cache.userToAuthoritySharedCache.tx.maxItems	100
cache.webScriptsRegistrySharedCache.backup-count	1
cache.webScriptsRegistrySharedCache.cluster.type	invalidating
cache.webScriptsRegistrySharedCache.eviction-percentage	25
cache.webScriptsRegistrySharedCache.eviction-policy	LRU
cache.webScriptsRegistrySharedCache.maxIdleSeconds	0
cache.webScriptsRegistrySharedCache.maxItems	1000
cache.webScriptsRegistrySharedCache.merge-policy	hz.ADD_NEW_ENTRY
cache.webScriptsRegistrySharedCache.timeToLiveSeconds	0
cache.webServicesQuerySessionSharedCache.backup-count	1
cache.webServicesQuerySessionSharedCache.cluster.type	fully-distributed
cache.webServicesQuerySessionSharedCache.eviction-percentage	25
cache.webServicesQuerySessionSharedCache.eviction-policy	LRU
cache.webServicesQuerySessionSharedCache.maxIdleSeconds	0
cache.webServicesQuerySessionSharedCache.maxItems	1000
cache.webServicesQuerySessionSharedCache.merge-policy	hz.ADD_NEW_ENTRY
cache.webServicesQuerySessionSharedCache.timeToLiveSeconds	0
cache.webServicesQuerySessionSharedCache.tx.maxItems	50
cache.zoneToAuthoritySharedCache.backup-count	1
cache.zoneToAuthoritySharedCache.cluster.type	invalidating
cache.zoneToAuthoritySharedCache.eviction-percentage	25
cache.zoneToAuthoritySharedCache.eviction-policy	LRU
cache.zoneToAuthoritySharedCache.maxIdleSeconds	0
cache.zoneToAuthoritySharedCache.maxItems	500
cache.zoneToAuthoritySharedCache.merge-policy	hz.ADD_NEW_ENTRY
cache.zoneToAuthoritySharedCache.timeToLiveSeconds	0
cache.zoneToAuthoritySharedCache.tx.maxItems	500
content.metadataExtractor.default.timeoutMs	20000
create.missing.people	<code> \${server.transaction.allow-writes}</code>
db.driver	org.postgresql.Driver
db.name	alfresco
db.password	admin
db.pool.abandoned.detect	false
db.pool.abandoned.log	false
db.pool.abandoned.time	300
db.pool.evict.idle.min	1800000
db.pool.evict.interval	600000

Attribute name	Example value
db.pool.evict.num.tests	-1
db.pool.evict.validate	false
db.pool.idle	10
db.pool.initial	10
db.pool.max	275
db.pool.min	10
db.pool.statements.enable	true
db.pool.statements.max	40
db.pool.validate.borrow	true
db.pool.validate.query	blank
db.pool.validate.return	false
db.pool.wait.max	-1
db.schema.name	blank
db.schema.stopAfterSchemaBootstrap	false
db.schema.update	true
db.schema.update.lockRetryCount	24
db.schema.update.lockRetryWaitSeconds	5
db.txn.isolation	-1
db.url	jdbc:postgresql://localhost:5432/\${db.name}
db.username	alfresco
default.async.action.corePoolSize	8
default.async.action.maximumPoolSize	20
default.async.action.threadPriority	1
deployment.avm.name	avm
deployment.avm.rootPath	/www/avm_webapps
deployment.avm.storeNamePattern	%storeName%-live
deployment.dmr.consolidate	true
deployment.dmr.name	alfresco
deployment.filesystem.autofix	true
deployment.filesystem.datadir	\${deployment.filesystem.rootdir}/depdata
deployment.filesystem.default.metadatadir	\${deployment.filesystem.metadatadir}/default
deployment.filesystem.default.name	filesystem
deployment.filesystem.default.rootdir	./www
deployment.filesystem.errorOnOverwrite	false
deployment.filesystem.logdir	\${deployment.filesystem.rootdir}/deploy
deployment.filesystem.metadatadir	\${deployment.filesystem.rootdir}/depmetadata
deployment.filesystem.rootdir	./wcm
deployment.service.corePoolSize	2
deployment.service.maximumPoolSize	3
deployment.service.numberOfSendingThreads	5
deployment.service.targetLockRefreshTime	60000
deployment.service.targetLockTimeout	3600000
deployment.service.threadPriority	5

Attribute name	Example value
dir.auditcontentstore	<code> \${dir.root}/audit.contentstore</code>
dir.cachedcontent	<code> \${dir.root}/cachedcontent</code>
dir.contentstore	<code> \${dir.root}/contentstore</code>
dir.contentstore.deleted	<code> \${dir.root}/contentstore.deleted</code>
dir.contentstore.tenants	blank
dir.indexes	<code> \${dir.root}/lucene-indexes</code>
dir.indexes.backup	<code> \${dir.root}/backup-lucene-indexes</code>
dir.indexes.lock	<code> \${dir.indexes}/locks</code>
dir.keystore	<code> \${dir.root}/keystore</code>
dir.license.external	C:/Alfresco
dir.root	C:/Alfresco/alf_data
domain.name.caseSensitive	false
domain.separator	blank
download.cleaner.maxAgeMins	60
download.cleaner.repeatIntervalMins	60
download.cleaner.startDelayMins	60
download.maxContentSize	2152852358
encryption.bootstrap.reencrypt	false
encryption.cipherAlgorithm	DESede/CBC/PKCS5Padding
encryption.keyAlgorithm	DESede
encryption.keySpec.class	org.alfresco.encryption. DESEDEKeyGenerator
encryption.keystore.backup.keyMetaData.location	<code> \${dir.keystore}/backup-keystore- passwords.properties</code>
encryption.keystore.backup.location	<code> \${dir.keystore}/backup-keystore</code>
encryption.keystore.backup.provider	blank
encryption.keystore.backup.type	JCEKS
encryption.keystore.keyMetaData.location	<code> \${dir.keystore}/keystore- passwords.properties</code>
encryption.keystore.location	<code> \${dir.keystore}/keystore</code>
encryption.keystore.provider	blank
encryption.keystore.type	JCEKS
encryption.mac.algorithm	HmacSHA1
encryption.mac.messageTimeout	30000
encryption.reencryptor.chunkSize	100
encryption.reencryptor.numThreads	2
encryption.ssl.keystore.keyMetaData.location	<code> \${dir.keystore}/ssl-keystore- passwords.properties</code>
encryption.ssl.keystore.location	<code> \${dir.keystore}/ssl.keystore</code>
encryption.ssl.keystore.provider	blank
encryption.ssl.keystore.type	JCEKS
encryption.ssl.truststore.keyMetaData.location	<code> \${dir.keystore}/ssl-truststore- passwords.properties</code>
encryption.ssl.truststore.location	<code> \${dir.keystore}/ssl.truststore</code>
encryption.ssl.truststore.provider	blank
encryption.ssl.truststore.type	JCEKS
fileFolderService.checkHidden.enabled	true

Attribute name	Example value
ftp.enabled	false
ftp.port	21
fts.indexer.batchSize	1000
hibernate.jdbc.use_get_generated_keys	false
home.folder.creation.disabled	false
home.folder.creation.eager	true
home_folder_provider_synchronizer.enabled	false
home_folder_provider_synchronizer.keep_empty_parents	false
home_folder_provider_synchronizer.override_provider	blank
hybridworkflow.enabled	false
imap.attachments.folder.folderPath	<code> \${spaces imap_attachments.childname}</code>
imap.attachments.folder.rootPath	<code> /\${spaces.company_home.childname}</code>
imap.attachments.folder.store	<code> \${spaces.store}</code>
imap.attachments.mode	SEPARATE
imap.config.home.folderPath	<code> \${spaces imap_home.childname}</code>
imap.config.home.rootPath	<code> /\${spaces.company_home.childname}</code>
imap.config.home.store	<code> \${spaces.store}</code>
imap.config.server.mountPoints	AlfrescoIMAP
imap.config.server.mountPoints.default.modeName	ARCHIVE
imap.config.server.mountPoints.default.mountPointName	IMAP
imap.config.server.mountPoints.default.rootPath	<code> \${protocols.rootPath}</code>
imap.config.server.mountPoints.default.store	<code> \${spaces.store}</code>
imap.config.server.mountPoints.value.AlfrescoIMAP.modeName	MIXED
imap.config.server.mountPoints.value.AlfrescoIMAP.mountPointName	Alfresco IMAP
imap.server.attachments.extraction.enabled	true
imap.server.enabled	false
imap.server.port	143
img.coders	<code> \${img.root}\modules\coders</code>
img.config	<code> \${img.root}\config</code>
img.dyn	<code> \${img.root}/lib</code>
img.exe	<code> \${img.root}\convert.exe</code>
img.gslib	<code> \${img.root}\lib</code>
img.root	C:\Alfresco\imagemagick
index.backup.cronExpression	0 0 3 * * ?
index.recovery.maximumPoolSize	5
index.recovery.mode	VALIDATE
index.recovery.stopOnError	false
index.reindexMissingContent.cronExpression	* * * * * ? 2099
index.subsystem.name	solr
index.tracking.adm.cronExpression	<code> \${index.tracking.cronExpression}</code>
index.tracking.avm.cronExpression	<code> \${index.tracking.cronExpression}</code>
index.tracking.cronExpression	0/5 * * * * ?
index.tracking.disableInTransactionIndexing	false
index.tracking.maxRecordSetSize	1000
index.tracking.maxTxnDurationMinutes	10

Attribute name	Example value
index.tracking.minRecordPurgeAgeDays	30
index.tracking.purgeSize	7200000
index.tracking.reindexLagMs	1000
jodconverter.enabled	true
jodconverter.officeHome	C:/Alfresco/libreoffice/App/ libreoffice
jodconverter.portNumbers	8100
kerberos.authentication.cifs.enableTicketCracking	false
location.license.embedded	/WEB-INF/alfresco/license/*.lic
location.license.external	file://\${dir.license.external}/*.lic
location.license.shared	classpath*:alfresco/extension/ license/*.lic
mail.service.corePoolSize	8
mail.service.maximumPoolSize	20
mbean.server.locateExistingServerIfPossible	true
monitor.rmi.service.enabled	true
monitor.rmi.service.port	50508
mybatis.useLocalCaches	false
nodes.bulkLoad.cachingThreshold	10
notification.email.siteinvite	false
ooo.enabled	false
ooo.exe	C:/Alfresco/libreoffice/App/ libreoffice/program/soffice.exe
ooo.port	8100
ooo.user	\${dir.root}/ouuser
opencmis.activities.enabled	true
opencmis.connector.default.objectsDefaultDepth	100
opencmis.connector.default.objectsDefaultMaxItems	10000
opencmis.connector.default.openHttpSession	false
opencmis.connector.default.rootPath	/\${spaces.company_home.childname} \${spaces.store}
opencmis.connector.default.store	-1
opencmis.connector.default.typesDefaultDepth	500
opencmis.connector.default.typesDefaultMaxItems	
opencmis.context.override	false
opencmis.context.value	false
opencmis.server.override	false
opencmis.server.value	blank
opencmis.servletpath.override	false
opencmis.servletpath.value	blank
orphanReaper.lockRefreshTime	60000
orphanReaper.lockTimeOut	3600000
people.search.honor_hint.useCQ	true
policy.content.update.ignoreEmpty	true
protocols.rootPath	/\${spaces.company_home.childname} \${spaces.store}
protocols.storeName	
publishing.root	\${publishing.root.path}/ \${spaces.publishing.root.childname}

Attribute name	Example value
publishing.root.path	/app:company_home/app:dictionary
replication.enabled	false
repo.remote.endpoint	/service
repository.name	Main Repository
sample.site.disabled	false
security.anyDenyDenies	true
server.maxusers	-1
server.setup.transaction.max-retries	40
server.setup.transaction.max-retry-wait-ms	15000
server.setup.transaction.min-retry-wait-ms	15000
server.setup.transaction.wait-increment-ms	10
server.transaction.allow-writes	true
server.transaction.max-retries	40
server.transaction.max-retry-wait-ms	2000
server.transaction.min-retry-wait-ms	100
server.transaction.mode.default	PROPAGATION_REQUIRED
server.transaction.mode.readOnly	PROPAGATION_REQUIRED, readOnly
server.transaction.wait-increment-ms	100
server.web.transaction.max-duration-ms	0
share.context	share
share.host	127.0.0.1
share.port	8080
share.protocol	http
shutdown.backstop.enabled	false
shutdown.backstop.timeout	10000
solr.cmis.alternativeDictionary	DEFAULT_DICTIONARY
solr.host	localhost
solr.max.host.connections	40
solr.max.total.connections	40
solr.port	8080
solr.port.ssl	8443
solr.secureComms	https
solr.solrConnectTimeout	5000
solr.solrPassword	solr
solr.solrPingCronExpression	0 0/5 * * * ? *
solr.solrUser	solr
solr.store.mappings	solrMappingAlfresco, solrMappingArchive
solr.store.mappings.value.solrMappingAlfresco.baseUrl	/solr/alfresco
solr.store.mappings.value.solrMappingAlfresco. httpClientFactory	solrHttpClientFactory
solr.store.mappings.value.solrMappingAlfresco.identifier	SpacesStore
solr.store.mappings.value.solrMappingAlfresco.protocol	workspace
solr.store.mappings.value.solrMappingArchive.baseUrl	/solr/archive
solr.store.mappings.value.solrMappingArchive. httpClientFactory	solrHttpClientFactory
solr.store.mappings.value.solrMappingArchive.identifier	SpacesStore

Attribute name	Example value
solr.store.mappings.value.solrMappingArchive.protocol	archive
spaces.archive.store	archive://SpacesStore
spaces.company_home.childname	app:company_home
spaces.content_forms.childname	app:forms
spaces.dictionary.childname	app:dictionary
spaces.emailActions.childname	app:email_actions
spaces.extension_webscripts.childname	cm:extensionwebscripts
spaces.guest_home.childname	app:guest_home
spaces imapConfig.childname	app:imap_configs
spaces imap_attachments.childname	cm:Imap Attachments
spaces imap_home.childname	cm:Imap Home
spaces imap_templates.childname	app:imap_templates
spaces inbound_transfer_records.childname	app:inbound_transfer_records
spaces.models.childname	app:models
spaces.nodetemplates.childname	app:node_templates
spaces.publishing.root.childname	app:publishing_root
spaces.rendition.rendering_actions.childname	app:rendering_actions
spaces.replication.replication_actions.childname	app:replication_actions
spaces.savedsearches.childname	app:saved_searches
spaces.scheduled_actions.childname	cm:Scheduled Actions
spaces.scripts.childname	app:scripts
spaces.searchAction.childname	cm:search
spaces.shared.childname	app:shared
spaces.sites.childname	st:sites
spaces.store	workspace://SpacesStore
spaces.templates.childname	app:space_templates
spaces.templates.content.childname	app:content_templates
spaces.templates.email.activities.childname	cm:activities
spaces.templates.email.childname	app:email_templates
spaces.templates.email.following.childname	app:following
spaces.templates.email.invite.childname	cm:invite
spaces.templates.email.invite1.childname	app:invite_email_templates
spaces.templates.email.notify.childname	app:notify_email_templates
spaces.templates.email.workflowemailnotification.childname	cm:workfownotification
spaces.templates.rss.childname	app:rss_templates
spaces.transfer_groups.childname	app:transfer_groups
spaces.transfer_temp.childname	app:temp
spaces.transfers.childname	app:transfers
spaces.user_homes.childname	app:user_homes
spaces.user_homes.regex.group_order	blank
spaces.user_homes.regex.key	userName
spaces.user_homes.regex.pattern	blank
spaces.wcm.childname	app:wcm
spaces.wcm_content_forms.childname	app:wcm_forms
spaces.wcm_deployed.childname	cm:wcm_deployed
spaces.webscripts.childname	cm:webscripts
spaces.workflow.definitions.childname	app:workflow_defs

Attribute name	Example value
subsystems.test.beanProp	inst1,inst2,inst3
subsystems.test.beanProp.default.anotherStringProperty	Global Default
subsystems.test.beanProp.default.longProperty	123456789123456789
subsystems.test.beanProp.value.inst2.boolProperty	true
subsystems.test.beanProp.value.inst3.anotherStringProperty	Global Instance Default
subsystems.test.simpleProp2	true
subsystems.test.simpleProp3	Global Default3
sync.checkLicenseForSyncMode	true
sync.cloud.url	https://a.alfresco.me/alfresco/a/{network}/
sync.mode	ON_PREMISE
sync.pullJob.enabled	true
sync.pushJob.enabled	true
system.acl.maxPermissionCheckTimeMillis	10000
system.acl.maxPermissionChecks	1000
system.authorities_container.childname	sys:authorities
system.bootstrap.config_check.strict	true
system.cache.disableImmutableSharedCaches	false
system.cache.disableMutableSharedCaches	false
system.cache.parentAssocs.limitFactor	8
system.cache.parentAssocs.maxSize	130000
system.certificate_container.childname	sys:samlcertificate
system.content.caching.cacheOnInbound	true
system.content.caching.contentCleanup.cronExpression	0 0 3 * * ?
system.content.caching.maxDeleteWatchCount	1
system.content.caching.maxFileSizeMB	0
system.content.caching.maxUsageMB	4096
system.content.caching.minLengthMillis	60000
system.content.contentUrlConverter batchSize	500
system.content.contentUrlConverter.cronExpression	* * * * * ? 2099
system.content.contentUrlConverter.runAsScheduledJob	false
system.content.contentUrlConverter.threadCount	2
system.content.deletionFailureAction	IGNORE
system.content.eagerOrphanCleanup	false
system.content.maximumFileSizeLimit	blank
system.content.orphanCleanup.cronExpression	0 0 4 * * ?
system.content.orphanProtectDays	14
system.descriptor.childname	sys:descriptor
system.descriptor.current.childname	sys:descriptor-current
system.downloads_container.childname	sys:downloads
system.enableTimestampPropagation	true
system.folderService.defaultListMaxResults	5000
system.hibernateMaxExecutions	20000
system.integrity.enabled	true
system.integrity.failOnViolation	true
system.integrity.maxErrorsPerTransaction	5
system.integrity.trace	false

Attribute name	Example value
system.maximumStringLength	-1
system.metadata-query-indexes.ignored	true
system.patch.sharedFolder.cronExpression	0 0 0 ? 1 1 2030
system.patch.sharedFolder.deferred	false
system.people_container.childname	sys:people
system.quickshare.enabled	true
system.readpermissions.bulkfetchsize	1000
system.readpermissions.optimise	true
system.remote_credentials_container.childname	sys:remote_credentials
system.store	system://system
system.syncset_definition_container.childname	sys:syncset_definitions
system.system_container.childname	sys:system
system.thumbnail.definition.default.maxPages	-1
system.thumbnail.definition.default.maxSourceSizeKBytes	-1
system.thumbnail.definition.default.pageLimit	1
system.thumbnail.definition.default.readLimitKBytes	-1
system.thumbnail.definition.default.readLimitTimeMs	-1
system.thumbnail.definition.default.timeoutMs	-1
system.thumbnail.generate	true
system.thumbnail.mimetype.maxSourceSizeKBytes.docx	-1
system.thumbnail.mimetype.maxSourceSizeKBytes.odp	-1
system.thumbnail.mimetype.maxSourceSizeKBytes.ods	-1
system.thumbnail.mimetype.maxSourceSizeKBytes.odt	-1
system.thumbnail.mimetype.maxSourceSizeKBytes.pdf	-1
system.thumbnail.mimetype.maxSourceSizeKBytes.pptx	-1
system.thumbnail.mimetype.maxSourceSizeKBytes.txt	-1
system.thumbnail.mimetype.maxSourceSizeKBytes.xlsx	-1
system.thumbnail.quietPeriod	604800
system.thumbnail.quietPeriodRetriesEnabled	true
system.thumbnail.redeployStaticDefsOnStartup	true
system.thumbnail.retryCount	2
system.thumbnail.retryPeriod	60
system.usages.clearBatchSize	0
system.usages.enabled	false
system.usages.updateBatchSize	50
system.webdav.activities.enabled	false
system.webdav.renameShufflePattern	(.*[\\..*]) (.*[a-f0-9]{8}+\$) (.*\\.tmp\$) (.*\\.wbk\$) (.*\\.bak\$) (.*\\~\$) (.*backup.*\\.do[ct]{1}[x]?[m]?\$)
system.webdav.rootPath	\${protocols.rootPath}
system.webdav.servlet.enabled	true
system.webdav.storeName	\${protocols.storeName}
system.webdav.url.path.prefix	blank
system.workflow.deployWorkflowsInTenant	true
system.workflow.deploy servlet.enabled	false
system.workflow.engine.activitidefinitions.visible	true
system.workflow.engine.activiti.idblocksize	100

Attribute name	Example value
system.workflow.maxAuthoritiesForPooledTasks	500
system.workflow.maxGroupReviewers	0
system.workflow.maxPooledTasks	-1
system.workflow_container.childname	sys:workflow
system.zones_container.childname	sys:zones
ticket.cleanup.cronExpression	0 0 * * * ?
transferservice.receiver.enabled	false
transferservice.receiver.lockRefreshTime	60000
transferservice.receiver.lockRetryCount	3
transferservice.receiver.lockRetryWait	100
transferservice.receiver.lockTimeOut	300000
transferservice.receiver.stagingDir	<code> \${java.io.tmpdir}/alfresco-transfer-staging</code>
transformer.Archive.includeContents	false
trashcan.MaxValue	1000
urlshortening.bitly.api.key	R_ca15c6c89e9b25cccd170baf209a0d4f
urlshortening.bitly.url.length	20
urlshortening.bitly.username	brianalfresco
user.name.caseSensitive	false
version.store.deprecated.lightWeightVersionStore	workspace://lightWeightVersionStore
version.store.enableAutoVersioning	true
version.store.migrateCleanupJob.batchSize	1
version.store.migrateCleanupJob.threadCount	3
version.store.migrateVersionStore.batchSize	1
version.store.migrateVersionStore.cronExpression	* * * * * ? 2099
version.store.migrateVersionStore.limitPerJobCycle	-1
version.store.migrateVersionStore.runAsScheduledJob	false
version.store.migrateVersionStore.threadCount	3
version.store.onlyUseDeprecatedV1	false
version.store.version2Store	workspace://version2Store
version.store.versionComparatorClass	blank
wcm.rename.max.time.milliseconds	2000
webscripts.encryptTempFiles	false
webscripts.memoryThreshold	4194304
webscripts.setMaxContentSize	4294967296
webscripts.tempDirectoryName	WebScripts
xforms.formatCaption	true



From Alfresco One 5.1 onwards, the `system.workflow.engine.activiti.enabled` property is no longer available.

Alfresco:Name=License, Object

Type=`org.alfresco.enterprise.repo.management.LicenseDescriptor`

Exposes the parameters of the Alfresco Enterprise license.

See the Alfresco Admin Console **General > License** for information about these attributes:

`http://<hostname>:<portnumber>/alfresco/service/enterprise/admin/admin-license.`

**Alfresco:Name=Log4jHierarchy, Object
Type=org.apache.log4j.jmx.HierarchyDynamicMBean**

Log4jHierarchy is an instance of the HierarchyDynamicMBean class provided with log4j that allows adjustments to be made to the level of detail included in the Alfresco server logs.

All read only attributes are prefixed with `logger=org.alfresco.` and the default value is the same as the attribute name, for example, `logger=org.alfresco.cmis` is the name and default value of the CMIS logger.

The editable `threshold` attribute is discussed in [JMX editable management beans](#) on page 670.

**Alfresco:Name=MetadataQueryIndexesCheck, Object
Type=org.alfresco.enterprise.repo.management.PatchCheck**

Exposes the metadata query index patch number and whether the patch has been applied:

Attribute name	Example value
Applied	true
PatchId	patch.db-V4.2-metadata-query-indexes

**Alfresco:Name=ModuleService, Object
Type=org.alfresco.enterprise.repo.management.ModuleService**

Allows monitoring of installed modules, listing modules that have been applied, and any missing modules:

Attribute name	Example value
AllModules	Format (a composite table containing the details of all modules currently installed): <pre>[[Attribute Name Attribute Value -----] module.description Alfresco Docs Integration module.id org.alfresco.integrations.google.docs module.installDate 2014-06-20T09:43:17.773+01:00 module.installState INSTALLED module.repo.version.max 999 module.repo.version.min 4.2.0 module.title Alfresco Google Docs Integration module.version 2.0.6]]</pre>
MissingModules	[]

**Alfresco:Name=PatchService, Object
Type=org.alfresco.enterprise.repo.management.PatchService**

Allows monitoring of installed patches.

Attribute name	Example value
AppliedPatches	Format (a composite table containing the details of all patches currently installed): <pre>[[Attribute Name Attribute Value ----- appliedOnDate Fri Jun 20 09:47:59 BST 2014 appliedToSchema 6052 appliedToServer 4.2.1 (r63452-b50) - Enterprise description Migrate old Tenant attributes fixesFromSchema 0 fixesToSchema 0 id patch.migrateAttrTenants report Not relevant to schema 6,052 succeeded true targetSchema 1 wasExecuted false]]</pre>

Alfresco:Name=RepoServerMgmt, Object Type=org.alfresco.repo.admin.RepoServerMgmt

Exposes information about the repository server, including the maximum number of users, user and ticket counts:

Attribute name	Example value
MaxUsers	-1
ReadOnly	false
TicketCountAll	2
TicketCountNonExpired	2
UserCountAll	2
UserCountNonExpired	2

Alfresco:Name=RepositoryDescriptor, object

Type=org.alfresco.enterprise.repo.management.RepositoryDescriptor, Type=*

Exposes metadata about the Alfresco repository. There are three types:

- Alfresco:Name=RepositoryDescriptor, Type=Current: exposes information about the current repository installation.
- Alfresco:Name=RepositoryDescriptor, Type=Initially Installed: exposes information about the initial repository installation, before any patches or upgrades were installed.
- Alfresco:Name=RepositoryDescriptor, Type=Server: exposes information about the current server version, as contained in the Alfresco war file. This instance should be used to determine the properties of the server.

See the Alfresco Admin Console **General > Repository Information** for information about these attributes: <http://<hostname>:<portnumber>/alfresco/service/enterprise/admin/admin-repositoryinfo>.

Alfresco:Name=RunningActions, Object

Type=org.alfresco.enterprise.repo.management.ActionsImpl

Exposes information about any actions that are in progress, including the number and action statistics:

Attribute name	Example value
ActionStatistics	[]
RunningActionCount	0
RunningActions	[]

Alfresco:Name=Runtime, Object Type=org.alfresco.enterprise.repo.management.Runtime

Exposes basic properties about the memory available to the JVM, including the amount of free memory and the maximum and total amount of memory in bytes.

 A Oracle JVM exposes much more detailed information through its platform MX Beans.

Attribute name	Example value
AvailableProcessors	8
AvailableProcessorsFreeMemory	391222616
MaxMemory	778502144
TotalMemory	778502144

Alfresco:Name=Schedule, Object

Type=org.alfresco.enterprise.scheduler.MonitoredRAMJobStore\$MonitoredCronTrigger, Group=*, Type=*, Trigger=*

Allows monitoring of the individual triggers, i.e. scheduled jobs, running in the Quartz scheduler. The attributes have various default settings but share the following meanings:

Group

The name of the schedule group that owns the trigger. Typically DEFAULT.

Type

The type of trigger, typically MonitoredCronTrigger or MonitoredSimpleTrigger. Triggers of different types have different properties.

Trigger

The name of the trigger itself. Must be unique within the group.

All instances have the following properties:

CalendarName

The name of the scheduling Calendar associated with the trigger, or null if there is not one.

Description

An optional textual description of the trigger.

EndTime

The time after which the trigger will stop repeating, if set.

FinalFireTime

The time at which the last execution of the trigger is scheduled, if applicable.

Group

The name of the schedule group that owns the trigger.

JobGroup

The name of the schedule group that owns the job executed by the trigger.

JobName

The name of the job executed by the trigger.

MayFireAgain

A Boolean that when true indicates that it is possible for the trigger to fire again.

Name

The name of the trigger.

NextFireTime

The next time at which the trigger will fire.

PreviousFireTime

The previous time at which the trigger fired.

Priority

A numeric priority that decides which trigger is executed before another in the event of a 'tie' in their scheduled times.

StartTime

The time at which the trigger should start.

State

The current state of the trigger.

Volatile

A Boolean that when true indicates that the trigger will not be remembered when the JVM is restarted.

When `Type=MonitoredCronTrigger`, the following additional properties are available:

CronExpression

A unix-like expression, using the same syntax as the cron command, that expresses when the job should be scheduled.

TimeZone

The name of the time zone to be used to interpret times.

If `Type=MonitoredSimpleTrigger`, the following additional properties are available:

RepeatCount

The number of times the job should repeat, after which it will be removed from the schedule. A value of -1 means repeat indefinitely.

RepeatInterval

The time interval in milliseconds between job executions.

TimesTriggered

The number of times the job has been run.

For specific information about the `propTablesCleanupTrigger` that you can scheduled to run database cleanup activities, see [Scheduling cleanup of database tables](#) on page 611.

Alfresco:Name=SolrIndexes, Object**Type=org.alfresco.enterprise.repo.management.SOLRIndex**

Allows monitoring of each searchable index. There are two types:

- `Alfresco:Name=SolrIndexes, Core=alfresco`: exposes information about the current index directory.
- `Alfresco:Name=SolrIndexes, Core=archive`: exposes information about the archive index directory.

Attribute name	Example value
Current	true
DataDirectory	C:\Alfresco\alf_data\solr\archive\SpacesStore\
HasDeletions	false
IndexInstanceDirectory	org.apache.lucene.store.SimpleFSDirectory:org.apache.lucene.store.SimpleFSDirectory@C:\Alfresco\alf_data\solr\archive\SpacesStore\index
InstanceDirectory	C:\Alfresco\alf_data\solr\archive-SpacesStore\
LastModified	Format: Thu Sep 24 14:32:30 BST 2015
MaxDocument	940
NumDocuments	940

Attribute name	Example value
Optimized	true
StartTime	Format: Mon Oct 12 10:36:20 BST 2015
Uptime	364362
Version	1403253989735

Alfresco:Name=SystemProperties, Object**Name=org.alfresco.enterprise.repo.management.PropertiesDynamicMBean**

Exposes all the system properties of the JVM. The set of standard system properties is documented on the Apache website.

See the Alfresco Admin Console **System Summary** for information about these attributes:

<http://<hostname>:<portnumber>/alfresco/service/enterprise/admin/admin-systemsummary>.

Alfresco:Name=WorkflowInformation, Object**Type=org.alfresco.enterprise.repo.management.Workflow**

Exposes information about the workflow management interface for Activiti definitions and tasks. Only the read-only attributes for `Alfresco:Name=WorkflowInformation` are shown in this table. For information about the editable attributes, see [JMX editable management beans](#) on page 670.

Attribute name	Example value
ActivitiEngineEnabled	true
ActivitiWorkflowDefinitionsVisible	true
NumberOfActivitiTaskInstances	0
NumberOfActivitiWorkflowDefinitionsDeployed	9
NumberOfActivitiWorkflowInstances	0

 The `ActivitiEngineEnabled` property is enabled by default. It is recommended that you do not change (or disable) this property via the JMX client.

JMX editable management beans

JMX values (Managed Bean or MBean attributes) are exposed in the Alfresco Admin Console and with internal tools (Alfresco JMX Dump) or external tools like JConsole. The editable management beans are described here with their default values where attributes are not already explained in the Alfresco Admin Console.

The default values given are the defaults for an installer-installed instance of Alfresco on Windows. These values can differ if you are using a different install method or operating system.

 Be aware that any changes you make to attributes in the live system are written to the database. The next time that Alfresco starts, these values will take precedence over any values specified in properties files, for example, `alfresco-global.properties`.

Alfresco:Type=Configuration, Category=ActivitiesFeed, Object Type=ActivitiesFeed \$default

This MBean provides information about the Activities Feed configuration in Alfresco, including whether the feed is enabled and how often users will receive Activities Feed emails.

See the Alfresco Admin Console **Repository Services - Activities Feed** for information about these editable attributes: <http://<hostname>:<portnumber>/alfresco/service/enterprise/admin/admin-activitiesfeed>.

Alfresco:Type=Configuration, Category=Audit, Object Type=Audit\$default

This MBean provides information about the audit configuration in Alfresco, namely which audit capabilities are enabled.

Attribute name	Example value
audit.alfresco-access.enabled	false
audit.cmischangelog.enabled	false
audit.enabled	true
audit.sync.enabled	true
audit.tagging.enabled	true

Alfresco:Type=Configuration, Category=Authentication, Object Type=Authentication \$managed\$alfrescoNtIml1

This MBean provides information about the authentication configuration in Alfresco.

See the Alfresco Admin Console **Directories - Directory Management** for information about these editable attributes: <http://<hostname>:<portnumber>/alfresco/service/enterprise/admin/admin-directorymanagement>.

Alfresco:Type=Configuration, Category=ContentStore, Object Type=ContentStore \$managed\$encrypted or Object Type=ContentStore\$managed\$unencrypted

This MBean provides information about the type of ContentStore used and its configuration in Alfresco. For more information, see [Set up encryption properties using JMX client](#) and [Encrypted Content Store properties](#).

Alfresco:Type=Configuration, Category=Events, Object Type=Events\$default

This MBean provides information about the type of events that are used in Alfresco for monitoring.

Attribute name	Values
alfresco.events.include	CONTENTPUT, NODEADDED, NODEREMOVED, NODEMOVED, NODERENAMED, NODECHECKOUTCANCELLED, NODECHECKEDOUT, NODECHECKEDIN

Alfresco:Type=Configuration, Category=Messaging, Object Type=Messaging\$default

This MBean provides information about the messages that are used in Alfresco Analytics.

Attribute name	Example value
messaging.activiti.enrichers.endpoint	vm:alfresco.activiti.events.enrichers
messaging.activiti.process.amqp	amqp:topic:alfresco.activiti.process? jmsMessageType=Text
messaging.activiti.sourceQueue.endpoint	direct-vm:alfresco.activiti.events
messaging.activiti.step.amqp	amqp:topic:alfresco.activiti.step? jmsMessageType=Text
messaging.activiti.variable.amqp	amqp:topic:alfresco.activiti.variable? jmsMessageType=Text
messaging.broker.connections.max	8
messaging.broker.url	failover:(tcp://localhost:61616)?timeout=3000
messaging.camel.context.id	alfrescoCamelContext
messaging.events.enrichers.endpoint	vm:alfresco.events.raw.enrichers
messaging.events.published.amqp	amqp:topic:alfresco.repo.events.activities? jmsMessageType=Text

Attribute name	Example value
messaging.events.published.endpoint	direct-vm:alfresco.events.raw
messaging.events.repo.node.sourceQueue.endpoint	direct-vm:alfresco.events
messaging.events.repo.node.targetTopic.endpoint	mqp:topic:alfresco.repo.events.nodes?
	jmsMessageType=Text
messaging.routing.numThreads	10
messaging.transacted	true

Alfresco:Type=Configuration, Category=OOoDirect, Object Type=OOoDirect\$default

This MBean provides information about the Open Office configuration in Alfresco.

Attribute name	Example value
ooo.enabled	false
ooo.exe	C:/Alfresco/libreoffice/App/libreoffice/program/soffice.exe
ooo.host	localhost
ooo.port	8100

Alfresco:Type=Configuration, Category=OOoJodconverter, Object Type=OOoJodconverter\$default

This MBean provides information about JODConverter configuration, which automates conversions between office document formats in Alfresco.

Attribute name	Example value
jodconverter.enabled	true
jodconverter.maxTasksPerProcess	200
jodconverter.officeHome	C:/Alfresco/libreoffice/App/libreoffice
jodconverter.portNumbers	8100
jodconverter.taskExecutionTimeout	120000
jodconverter.taskQueueTimeout	30000
jodconverter.templateProfileDir	

Alfresco:Type=Configuration, Category=Replication, Object Type=Replication\$default

This MBean provides information about the replication settings between repositories, and the replication configuration in Alfresco.

See the Alfresco Admin Console **Repository Services - Replication Service** for information about these editable attributes: <http://<hostname>:<portnumber>/alfresco/service/enterprise/admin/admin-replicationservice>.

Alfresco:Type=Configuration, Category=Search, Object Types=Search\$*

This MBean provides information about the search service in use (Solr, Solr 4 or no index) and the configuration for that search in Alfresco.

See the Alfresco Admin Console **Repository Services - Search Service** for information about these editable attributes: <http://<hostname>:<portnumber>/alfresco/service/enterprise/admin/admin-searchservice>.

Alfresco:Type=Configuration, Category=Subscriptions, Object Type=Subscriptions\$default

This MBean provides information about settings for subscriptions in Alfresco, including whether subscriptions are enabled and any template paths.

See the Alfresco Admin Console **Repository Services - Subscription Service** for information about these editable attributes: `http://<hostname>:<portnumber>/alfresco/service/enterprise/admin/admin-subscriptionservice`.

Alfresco:Type=Configuration, Category=Synchronization, Object Type=Synchronization\$default

This MBean provides information about the synchronization settings in Alfresco, including when to synchronize and logging intervals.

Attribute name	Example value
synchronization.allowDeletions	true
synchronization.autoCreatePeopleOnLogin	true
synchronization.import.cron	0 0 0 * * ?
synchronization.loggingInterval	100
synchronization.syncOnStartup	true
synchronization.syncWhenMissingPeopleLogin	true
synchronization.synchronizeChangesOnly	true
synchronization.workerThreads	1

Alfresco:Type=Configuration, Category=Transformers, Object Type=Transformers\$default

This MBean provides information about the transformation service setting for converting between different file formats.

See the Alfresco Admin Console **Repository Services - Transformation Services** for information about these editable attributes: `http://<hostname>:<portnumber>/alfresco/service/enterprise/admin/admin-transformations`.

Alfresco:Type=Configuration, Category=WebDav, Object Type=org.alfresco.enterprise.repo.management.WebDav

This MBean provides information about whether WebDav is enabled or disabled for Alfresco.

Attribute name	Example value
Enabled	true

Alfresco:Type=Configuration, Category=email, Object Types=email\$inbound, email\$outbound

This MBean provides information about the inbound and outbound email configuration in Alfresco, including server, protocol and encoding information.

See the Alfresco Admin Console **Email Services - Inbound Email and Email Services - Outbound Email** for information about these editable attributes: `http://<hostname>:<portnumber>/alfresco/service/enterprise/admin/admin-inboundemail` and `http://<hostname>:<portnumber>/alfresco/service/enterprise/admin/admin-outboundemail`.

Alfresco:Type=Configuration, Category=fileServers, Object Type=fileServers\$default

This MBean provides information about the CIFS and FTP servers configured in Alfresco.

See the Alfresco Admin Console **Virtual File Systems - File Servers** for information about these editable attributes: `http://<hostname>:<portnumber>/alfresco/service/enterprise/admin/admin-filesystems`.

Alfresco:Type=Configuration, Category=googledocs, Object Type=googledocs\$v2

This MBean provides information about the Google Docs configuration in Alfresco.

Attribute name	Example value
googledocs.enabled	true
googledocs.idleThresholdSeconds	600
googledocs.version	2.0.6-12ent

Alfresco:Type=Configuration, Category=imap, Object Types=imap\$default, imap\$default \$imap.config.server.mountPoints\$AlfrescoIMAP

This MBean provides information about the IMAP mail and server configuration in Alfresco.

See the Alfresco Admin Console **Virtual File Systems - IMAP Service** for information about these editable attributes: <http://<hostname>:<portnumber>/alfresco/service/enterprise/admin/admin-imap>.

Alfresco:Type=Configuration, Category=sysAdmin, Object Type=sysAdmin\$default

This MBean provides information about the administration settings for the Alfresco Share.

Attribute name	Example value
alfresco.context	alfresco
alfresco.host	127.0.0.1
alfresco.port	8080
alfresco.protocol	http
server.allowWrite	true
server.allowedusers	
server.maxusers	-1
share.context	share
share.host	127.0.0.1
share.port	8080
share.protocol	http
site.public.group	GROUP_EVERYONE

Alfresco:Type=Configuration, Category=thirdParty, Object Type=thirdparty\$default

This MBean provides information about the third party modules configured in Alfresco.

Attribute name	Example value
img.coders	C:\Alfresco\imagemagick\modules\coders
img.config	C:\Alfresco\imagemagick\config
img.dyn	C:\Alfresco\imagemagick\lib
img.exe	C:\Alfresco\imagemagick\convert.exe
img.gslib	C:\Alfresco\imagemagick\lib
img.root	C:\Alfresco\imagemagick

Alfresco:Name=FileServerConfig, Object Type=com.sun.proxy\$Proxy108

This MBean allows management and monitoring of the CIFS and FTP servers configured in Alfresco. See the Alfresco Admin Console **Virtual File Systems - File Servers** for information about these editable attributes: <http://<hostname>:<portnumber>/alfresco/service/enterprise/admin/admin-fileservers>.

Alfresco:Name=Log4jHierarchy, Object Type=org.apache.log4j.jmx.HierarchyDynamicMBean

This MBean is an instance of the HierarchyDynamicMBean class provided with log4j that allows adjustments to be made to the level of detail included in the Alfresco server logs. Not all attributes

for Alfresco:Name=Log4jHierarchy are editable; only those that are editable are shown in this table.

Attribute name	Example value
threshold	ALL

Threshold is a special attribute that controls the server-wide logging threshold. It is not cluster aware. Its value must be the name of one of the log4j logging levels.

- ⚠ Any messages logged with a priority lower than this threshold will be filtered from the logs. The default value is ALL, which means no messages are filtered, and the highest level of filtering is OFF which turns off logging altogether (not recommended).

You can add a new logger to Log4jHierarchy by selecting the **Operations > addLoggerMBean** operation in JConsole and specify the string and priority for the new logger MBean. The bean will be given an additional read-only property for that logger and a new MBean will be registered in the #log4j:logger=* tree, allowing management of that logger.

It is not normally necessary to use this operation, because the Alfresco server pre-registers all loggers initialized during startup. However, if the logger you are interested in was not initialized at this point, you will have to add a logger. Add the fully qualified name of the logger as an argument and if successful, the object name of the newly registered MBean for managing that logger is returned. The logger is then displayed in the attribute list of log4j loggers.

For example, if in Java class `org.alfresco.repo.admin.patch.PatchExecuter` the logger is initialized as follows:

```
private static Log logger = LogFactory.getLog(PatchExecuter.class);
```

Then the logger name would be `org.alfresco.repo.admin.patch.PatchExecuter`.

Alfresco:Name=Schedule, Object

Type=`org.alfresco.enterprise.scheduler.MonitoredRAMJobStore$MonitoredCronTrigger`

This MBean provides information on the triggers that are configured for Alfresco, for example, those that are started by cron jobs or other events. You can fire a trigger by selecting the required MBean trigger and **Operations > executeNow**. The selected trigger is started.

Alfresco:Name=WorkflowInformation, Object

Type=`org.alfresco.enterprise.repo.management.Workflow`

Exposes information about the workflow management interface for Activiti definitions and tasks. The attributes for Alfresco:Name=WorkflowInformation are shown in this table.

Attribute name	Example value
ActivitiEngineEnabled	true
ActivitiWorkflowDefinitionsVisible	true
NumberOfActivitiTaskInstances	0
NumberOfActivitiWorkflowDefinitionsDeployed	9
NumberOfActivitiWorkflowInstances	0

- ⚠ The ActivitiEngineEnabled attribute is enabled by default. It is recommended that you do not change (or disable) this property via the JMX client.

See the Alfresco Admin Console **Repository Services - Process Engines** for information about these editable attributes: <http://<hostname>:<portnumber>/alfresco/service/enterprise/admin/admin-processengines>.

log4j:logger=*

An instance of the LoggerDynamicMBean class provided with log4j that allows adjustments to be made to the level of detail included in the logs from an individual logger.

Not all attributes for `log4j:logger=*` are editable; only those that are editable are shown in this table.

Attribute name	Example value
priority	WARN

Priority is a special attribute that specifies the minimum log4j logging level of messages from this logger to include in the logs. For example, a value of `ERROR` would mean that messages logged at lower levels such as `WARN` and `INFO` would not be included.

You can change the priority of any log4j attribute by selecting the required MBean and **Attributes > priority**. The new value does not prevail after a shutdown of Alfresco. For a list of possible priority values, see [Log4j priority settings](#).

Alfresco Full Text Search Reference

The following sections describe the Alfresco Full Text Search (FTS) syntax.

The Alfresco Full Text Search (FTS) query text can be used standalone or it can be embedded in CMIS-SQL using the `contains()` predicate function. The CMIS specification supports a subset of Alfresco FTS. The full power of Alfresco FTS can not be used and, at the same time, maintain portability between CMIS repositories.

Alfresco FTS is exposed directly by the interface, which adds its own template, and is also used as its default field. The default template is:

```
%({cm:name cm:title cm:description ia:whatEvent ia:descriptionEvent lnk:title  
lnk:description TEXT)
```

When Alfresco FTS is embedded in CMIS-SQL, only the CMIS-SQL-style property identifiers (`cmis:name`) and aliases, CMIS-SQL column aliases, and the special fields listed can be used to identify fields. The SQL query defines tables and table aliases after `from` and `join` clauses. If the SQL query references more than one table, the `contains()` function must specify a single table to use by its alias. All properties in the embedded FTS query are added to this table and all column aliases used in the FTS query must refer to the same table. For a single table, the table alias is not required as part of the `contains()` function.

When Alfresco FTS is used standalone, fields can also be identified using `prefix:local-name` and `{uri}local-name` styles.

Search for a single term

Single terms are tokenized before the search according to the appropriate data dictionary definition(s).

If you do not specify a field, it will search in the content and properties. This is a shortcut for searching all properties of type content.

```
banana  
TEXT:banana
```

Both of these queries will find any nodes with the word "banana" in any property of type `d:content`.

If the appropriate data dictionary definition(s) for the field supports both FTS and untokenized search, then FTS search will be used. FTS will include synonyms if the analyzer generates them. Terms cannot contain whitespace.

Search for a phrase

Phrases are enclosed in double quotes. Any embedded quotes can be escaped using "`\`". If no field is specified then the default `TEXT` field will be used, as with searches for a single term.

The whole phrase will be tokenized before the search according to the appropriate data dictionary definition(s).

```
"big yellow banana"
```

Search for an exact term

To search for an exact term, prefix the term with "`=`". This ensures that the term will not be tokenized, therefore you can search for stop words.

If both FTS and ID base search are supported for a specified or implied property, then exact matching will be used where possible.

```
=running
```

Will match "running" but will not be tokenized. If you are using stemming it might not match anything.

For the `cm:name` field, which is in the index as both tokenized and untokenized, it will use the untokenized field. For example, `=part` will only match the exact term "part". If you use `=part*` it will match additional terms, like "partners". If there is no untokenized field in the index, it will fall back to use the tokenized field, and then, with stemming/plurals, it would match.

Search for term expansion

To force tokenization and term expansion, prefix the term with "`~`".

For a property with both ID and FTS indexes, where the ID index is the default, force the use of the FTS index.

```
~running
```

Search for conjunctions

Single terms, phrases, and so on can be combined using "AND" in upper, lower, or mixed case.

```
big AND yellow AND banana
TEXT:big and TEXT:yellow and TEXT:banana
```

These queries search for nodes that contain the terms "big", "yellow", and "banana" in any content.

Search for disjunctions

Single terms, phrases, and so on can be combined using "OR" in upper, lower, or mixed case.

If not otherwise specified, by default search fragments will be ORed together.

```
big yellow banana
big OR yellow OR banana
TEXT:big TEXT:yellow TEXT:banana
TEXT:big OR TEXT:yellow OR TEXT:banana
```

These queries search for nodes that contain the terms "big", "yellow", or "banana" in any content.

Search for negation

Single terms, phrases, and so on can be combined using "NOT" in upper, lower, or mixed case, or prefixed with "!" or "-".

```
yellow NOT banana
yellow !banana
yellow -banana
NOT yellow banana
-yellow banana
!yellow banana
```

Search for optional, mandatory, and excluded elements of a query

Sometimes AND and OR are not enough. If you want to find documents that must contain the term "car", score those with the term "red" higher, but do not match those just containing "red".

Operator	Description
" "	The field, phrase, group is optional; a match increases the score.
The field, phrase, group is mandatory (Note: this differs from Google - see "=")	
"-", "!"	The field, phrase, group must not match.

The following example finds documents that contain the term "car", score those with the term "red" higher, but does not match those just containing "red":

```
+car | red
```

 At least one element of a query must match (or not match) for there to be any results.

All AND and OR constructs can be expressed with these operators.

Search for fields

Search specific fields rather than the default. Terms, phrases, etc. can all be preceded by a field. If not the default field TEXT is used.

```
field:term
field:"phrase"
=field:exact
~field:expand
```

Fields fall into three types: property fields, special fields, and fields for data types.

Property fields evaluate the search term against a particular property, special fields are described in the following table, and data type fields evaluate the search term against all properties of the given type.

Description	Type	Example
Fully qualified property	Property	{http://www.alfresco.org/model/content/1.0}name:apple
Fully qualified property	Property	@{http://www.alfresco.org/model/content/1.0}name:apple
CMIS style property	Property	cm_name:apple
Prefix style property	Property	cm:name:apple
Prefix style property	Property	@cm:name:apple
TEXT	Special	TEXT:apple
ID	Special	ID:"NodeRef"
ISROOT	Special	ISROOT:T
TX	Special	TX:"TX"
PARENT	Special	PARENT:"NodeRef"
PRIMARYPARENT	Special	PRIMARYPARENT:"NodeRef"

Description	Type	Example
QNAME	Special	QNAME:"app:company_home"
CLASS	Special	CLASS:"qname"
EXACTCLASS	Special	EXACTCLASS:"qname"
TYPE	Special	TYPE:"qname"
EXACTTYPE	Special	EXACTTYPE:"qname"
ASPECT	Special	ASPECT:"qname"
EXACTASPECT	Special	EXACTASPECT:"qname"
ISUNSET	Special	ISUNSET:"property-qname"
ISNULL	Special	ISNULL:"property-qname"
ISNOTNULL	Special	ISNOTNULL:"property-qname"
EXISTS	Special	EXISTS:"name of the property"
SITE	Special	SITE:"shortname of the site"
TAG	Special	TAG: "name of the tag"
Fully qualified data type	Data Type	{http://www.alfresco.org/model/dictionary/1.0}content:apple
prefixed data type	Data Type	d:content:apple

Search for wildcards

Wildcards are supported in terms, phrases, and exact phrases using "*" to match zero, one, or more characters and "?" to match a single character. The "*" wildcard character can appear on its own and implies Google-style. The "anywhere after" wildcard pattern can be combined with the "=" prefix for identifier based pattern matching.

The following will all find the term apple.

```
TEXT:app?e
TEXT:app*
TEXT:*pple
appl?
*ppe
=pple
"ap*le"
"***le"
"?????"
```

When performing a search that includes a wildcard character, it is best to wrap your search term in double quotation marks. This ensures all metadata and content are searched.

Search for ranges

Inclusive ranges can be specified in Google-style. There is an extended syntax for more complex ranges. Unbounded ranges can be defined using MIN and MAX for numeric and date types and "\u0000" and "\xFFFF" for text (anything that is invalid).

Lucene	Google	Description	Example
[#1 TO #2]	#1..#2	The range #1 to #2 inclusive #1 <= x <= #2	0..5 [0 TO 5]

Lucene	Google	Description	Example
<#1 TO #2]		The range #1 to #2 including #2 but not #1. #1 < x <= #2	<0 TO 5]
[#1 TO #2>		The range #1 to #2 including #1 but not #2. #1 <= x < #2	[0 TO 5>
<#1 TO #2>		The range #1 to #2 exclusive. #1 < x < #2	<0 TO 5>

```
TEXT:apple..banana
my:int:[0 TO 10]
my:float:2.5..3.5
my:float:0..MAX
mt:text:[1 TO "\uFFFF"]
```

Search for fuzzy matching

Fuzzy matching is not implemented. The default Lucene implementation is Levenshtein Distance, which is expensive to evaluate.

Postfix terms with "float"

```
apple~0.8
```

Search for proximity

Google-style proximity is supported.

To specify proximity for fields, use grouping.

```
big * apple
TEXT:(big * apple)
big *(3) apple
TEXT:(big *(3) apple)
```

Search for boosts

Query time boosts allow matches on certain parts of the query to influence the score more than others.

All query elements can be boosted: terms, phrases, exact terms, expanded terms, proximity (only in field groups), ranges, and groups.

```
term^2.4
"phrase" ^3
term~0.8^4
=term^3
~term^4
cm:name:(big * yellow)^4
1..2^2
[1 TO 2]^2
yellow AND (car OR bus)^3
```

Search for grouping

Groupings of terms are made using "(" and ")". Groupings of all query elements are supported in general. Groupings are also supported after a field - field group.

The query elements in field groups all apply to the same field and cannot include a field.

```
(big OR large) AND banana
```

```
title:((big OR large) AND banana)
```

Search for spans and positions

Spans and positions are not implemented. Positions will depend on tokenization.

Anything more detailed than one *(2) two are arbitrarily dependent on the tokenization. An identifier and pattern matching, or dual FTS and ID tokenization, might be the answer in these cases.

```
term[^] - start
term[$] - end
term[position]
```

These are of possible use but excluded for now. Lucene surround extensions:

```
and(terms etc)
99w(terms etc)
97n(terms etc)
```

Escaping characters

Any character can be escaped using the backslash "\\" in terms, IDs (field identifiers), and phrases. Java unicode escape sequences are supported. Whitespace can be escaped in terms and IDs.

For example:

```
cm:my\ content:my\ name
```

Mixed FTS ID behavior

This relates to the priority defined on properties in the data dictionary, which can be both tokenized or untokenized.

Explicit priority is set by prefixing the query with "=" for identifier pattern matches.

The tilde "~" can be used to force tokenization.

Search for order precedence

Operator precedence is SQL-like (not Java-like). When there is more than one logical operator in a statement, and they are not explicitly grouped using parentheses, NOT is evaluated first, then AND, and finally OR.

The following shows the operator precedence from highest to lowest:

```
""
[ , ] , < , >
()
~ (prefix and postfix) , =
^
+ , | , -
NOT,
AND
OR
```

AND and OR can be combined with +, |, - with the following meanings:

AND (no prefix is the same as +)	Explanation
big AND dog	big and dog must occur
+big AND +dog	big and dog must occur
big AND +dog	big and dog must occur

AND (no prefix is the same as +)	Explanation
+big AND dog	big and dog must occur
big AND dog	big must occur and dog should occur
big AND dog	big should occur and dog must occur
big AND dog	both big and dog should occur, and at least one must match
big AND -dog	big must occur and dog must not occur
-big AND dog	big must not occur and dog must occur
-big AND -dog	both big and dog must not occur
big AND -dog	big should occur and dog must not occur

OR (no prefix is the same as +)	Explanation
dog OR wolf	dog and wolf should occur, and at least one must match
+dog OR +wolf	dog and wolf should occur, and at least one must match
dog OR +wolf	dog and wolf should occur, and at least one must match
+dog OR wolf	dog and wolf should occur, and at least one must match
dog OR wolf	dog and wolf should occur, and at least one must match
dog OR wolf	dog and wolf should occur, and at least one must match
dog OR wolf	dog and wolf should occur, and at least one must match
dog OR -wolf	dog should occur and wolf should not occur, one of the clauses must be valid for any result
-dog OR wolf	dog should not occur and wolf should occur, one of the clauses must be valid for any result
-dog OR -wolf	dog and wolf should not occur, one of the clauses must be valid for any result

Search query syntax APIs

These examples show how to embed queries in CMIS.

Embedded in CMIS contains()

```

- strict queries
SELECT * FROM Document WHERE CONTAINS('\'zebra\'')
SELECT * FROM Document WHERE CONTAINS('\'quick\'')

- Alfresco extensions
SELECT * FROM Document D WHERE CONTAINS(D, 'cmis:name:\'Tutorial\'')
SELECT cmis:name as BOO FROM Document D WHERE CONTAINS('BOO:\'Tutorial\'')

```

Search Service

```

ResultSet results = searchService.query(storeRef,
    SearchService.LANGUAGE_FTS_ALFRESCO, "quick");

SearchService.LANGUAGE_FTS_ALFRESCO = "fts-alfresco"

```

Node Browser

Alfresco FTS is supported in the node browser.

JavaScript

```

search
{
  query: string,           mandatory, in appropriate format and encoded for the
  given language
  store: string,           optional, defaults to 'workspace://SpacesStore'
  language: string,        optional, one of: lucene, xpath, jcr-xpath, fts-
alfresco - defaults to 'lucene'
  templates: [],           optional, Array of query language template objects
  (see below) - if supported by the language
  sort: [],                optional, Array of sort column objects (see below) -
if supported by the language
  page: object,            optional, paging information object (see below) - if
supported by the language
  namespace: string,       optional, the default namespace for properties
  defaultField: string,   optional, the default field for query elements when
not explicit in the query
  onerror: string          optional, result on error - one of: exception, no-
results - defaults to 'exception'
}

sort
{
  column: string,          mandatory, sort column in appropriate format for the
  language
  ascending: boolean       optional, defaults to false
}

page
{
  maxItems: int,            optional, max number of items to return in result
  set
    skipCount: int         optional, number of items to skip over before
  returning results
}

template
{
  field: string,            mandatory, custom field name for the template
  template: string          mandatory, query template replacement for the
  template
}

```

For example:

```

var def =
{
  query: "cm:name:test*",
  language: "fts-alfresco"
};
var results = search.query(def);

```

Templates

Alfresco FTS is not supported in FreeMarker.

Search query templates

The FTS query language supports query templates. These are intended to help when building application specific searches.

A template is a query but with additional support to specify template substitution.

%field

Insert the parse tree for the current `ftstest` and replace all references to fields in the current parse tree with the supplied field.

%{field1, field2}**%{field1 field2}**

(The comma is optional.) Create a disjunction, and for each field, add the parse tree for the current `ftstest` to the disjunction, and then replace all references to fields in the current parse tree with the current field from the list.

Name	Template	Example Query	Expanded Query
t1	%cm:name	t1:n1	cm:name:n1
t1	%cm:name	t1:"n1"	cm:name:"n1"
t1	%cm:name	~t1:n1^4	~cm:name:n1^4
t2	%{cm:name, cm:title}	t2:"woof"	(cm:name:"woof" OR cm:title:"woof")
t2	%{cm:name, cm:title}	~t2:woof^4	(~cm:name:woof OR ~cm:title:woof)^4
t3	%cm:name AND my:boolean:true	t3:banana	(cm:name:banana AND my:boolean:true)

Templates can refer to other templates.

```
nameAndTitle -> %{cm:name, cm:title}
nameAndTitleAndDescription -> %{nameAndTitle, cm:description}
```

Search query literals

When you search, entries are generally a term or a phrase. The string representation you type in will be transformed to the appropriate type for each property when executing the query. For convenience, there are numeric literals but string literals can also be used.

String literals for phrases can be enclosed in double quotes or single quotes. Java single character and \uXXXX-based escaping are supported within these literals.

Integer and decimal literals conform to the Java definitions.

Dates as any other literal can be expressed as a term or phrase. Dates are in the format Any or all of the time can be truncated. All of the date must be present.

The date type also supports `NOW` syntax.

In range queries, strings, term, and phrases that do not parse to valid type instance for the property are treated as open ended.

```
test:integer[ 0 TO MAX] matches anything positive
```

Date ranges do not respect the truncated resolution that can be presented in range queries.

Forms reference

This reference contains detailed information for forms controls and the configuration syntax.

Form controls

Controls are represented by a FreeMarker template snippet, and each field has a control and an optional set of parameters.

The following controls are available.

association.ftl

The `association` control is used to allow objects in the repository to be picked and ultimately associated with the node being edited. The control uses the JavaScript `Alfresco.ObjectPicker` component to allow the user to browse the repository and pick objects.

The following parameters are available:

- `compactMode`: Determines whether the picker will be shown in compact mode.
- `showTargetLink`: Determines whether a link to the document details page will be rendered to content items.

category.ftl

The `category` control is used to allow the user to select categories for the node being edited. The control uses the JavaScript `Alfresco.ObjectPicker` component to allow the user to browse the category hierarchy.

The following parameters are available:

- `compactMode`: Determines whether the picker will be shown in compact mode.

checkbox.ftl

The `checkbox` control renders a standard HTML check box control.

The following parameters are available:

- `styleClass`: Allows a custom CSS class to be applied to the check box.

date.ftl

The `date` control renders a date field allowing free form entry of dates, as well as a calendar widget allowing dates to be selected visually. If appropriate a time field is also rendered.

The following parameters are available:

- `showTime`: Determines whether the time entry field should be displayed.

encoding.ftl

The `encoding` control renders a selectable list of encodings.

The following parameters are available:

- `property`: The name of a content property to retrieve the current encoding from; if omitted the `field.value` value is used.
- `styleClass`: Allows a custom CSS class to be applied to the select list.

invisible.ftl

The `invisible` control renders nothing at all; it can be used when a form definition needs to be requested and returned but not displayed. This control has no parameters.

mimetype.ftl

The `mimetype` control renders a selectable list of mime types.

The following parameters are available:

- `property`: The name of a content property to retrieve the current mime type from, if omitted the `field.value` value is used.
- `styleClass`: Allows a custom CSS class to be applied to the select list.

period.ftl

The `period` control renders a selectable list of periods and an expression entry field.

The following parameters are available:

- `dataTypeParameters`: A JSON object representing the period definitions to show in the list.

selectone.ftl

The `selectone` control renders a standard HTML select list.

The following parameters are available:

- `options`: A comma separated list of options to display, for example "First,Second,Third". If a value for an option also needs to be specified, use the "First|1,Second|2,Third|3" format.
- `size`: The size of the list, that is, how many options are always visible.
- `styleClass`: Allows a custom CSS class to be applied to the select list.

selectmany.ftl

The `selectmany` control renders a standard HTML select list allowing multiple selections.

The following parameters are available:

- `options` (mandatory, comma separated string): A comma separated list of options to display, for example "First,Second,Third". If a value for an option also needs to be specified the "First|1,Second|2,Third|3" format can be used.
- `size` (optional, int): The size of the list i.e. how many options are always visible, the default is 5.
- `styleClass` (optional, string): Allows a custom CSS class to be applied to the select list.
- `style` (optional, string): Allows CSS rules to applied directly to the select list.
- `forceEditable` (optional, boolean): Forces the control to be editable, default is false.

size.ftl

The `size` control renders a read only human readable representation of the content size.

The following parameters are available:

- `property`: The name of a content property to retrieve the current content size from; if omitted the `field.value` value is used.

textarea.ftl

The `textarea` control renders a standard HTML text area field.

The following parameters are available:

- `rows`: The number of rows the text area will have
- `columns`: The number of columns the text area will have
- `styleClass`: Allows a custom CSS class to be applied to the text area

textfield.ftl

The `textfield` control renders a standard HTML text field.

The following parameters are available:

- `styleClass`: Allows a custom CSS class to be applied to the text field
- `maxLength`: Defines the maximum number of characters the user can enter
- `size`: Defines the size of the text field

Forms configuration syntax

The `share-config-custom.xml` file uses an XML configuration syntax.

The XML syntax is described as follows:

default-controls

The type element defines what control to use, by default, for each type defined in the Alfresco content model. The name attribute contains the prefix form of the data type, for example `d:text`. The template attribute specifies the path to the template snippet to use to represent the field. If the path value should be a relative path, it is relative from the `alfresco` package. If the path value is absolute, it is looked up relative to the `alfresco/web-extension/site-webscripts` package, normally found in the application server shared classes location. The control-param element provides a mechanism to pass parameters to control templates, meaning that control templates can be re-used.

constraint-handlers

The constraint element defines what JavaScript function to use to check that fields with constraints are valid before being submitted. The `id` attribute is the unique identifier given to the model constraint in the Alfresco content model, for example `LIST`. The validation-handler attribute represents the name of a JavaScript function that gets called when the field value needs to be validated. The event attribute defines what event will cause the validation handler to get called. This will be a standard DOM event, that is, `keyup`, `blur`, and so on. The validation handler called usually has a default message to display when validation fails, the message and message-id attributes provide a way to override this message. However, the validation messages are not shown (the **Submit** button is enabled/disabled).

dependencies

The dependencies element defines the list of JavaScript and CSS files required by any custom controls being used in the application. In order for valid XHTML code to be generated, the dependencies need to be known ahead of time so the relevant links can be generated in the HTML head section. The `src` attribute of both the JavaScript and CSS elements contains the path to the resource, the path should be an absolute path from the root of the web application (but not including the web application context).

form

The `form` element represents a form to display. If the form element exists within a config element that provides an evaluator and condition, the form will only be found if the item being requested matches the condition. If the form element exists within a config element without an evaluator and condition, the form is always found. The optional `id` attribute allows an identifier to be associated with the form, thus allowing multiple forms to be defined for the same item. The `submission-url` allows the action attribute of the generated form to be overridden so that the contents of the form can be submitted to any arbitrary URL.

view-form

The `view-form` element allows the default template that auto generates the form UI to be overridden. The template attribute specifies the path to the template to be used when the form is in view mode. The value is usually an absolute path, which is relative to the `alfresco/web-extension/site-webscripts` package, normally found in the application server shared classes location. If this element is present, the `field-visibility` element is effectively ignored and therefore does not have to be present.

edit-form

The `edit-form` element allows the default template that auto generates the form UI to be overridden. The template attribute specifies the path to the template to be used when the form is in edit mode. The value is usually an absolute path, which is relative to the `alfresco/web-extension/site-webscripts` package, normally found in the application server shared classes location. If this element is present, the `field-visibility` element is effectively ignored and therefore does not have to be present.

create-form

The `create-form` element allows the default template that auto generates the form UI to be overridden. The `template` attribute specifies the path to the template to be used when the form is in create mode. The value is usually an absolute path, which is relative to the `alfresco/web-extension/site-webscripts` package, normally found in the application server shared classes location. If this element is present, the `field-visibility` element is effectively ignored and therefore does not have to be present.

field-visibility

The `field-visibility` element defines which fields are going to appear on the form, unless a custom template is used.

show

The `show` element specifies a field that should appear on the form. The `id` attribute represents the unique identifier for a field, for example, `cm:name`. The optional `for-mode` attribute indicates when the field should appear. Valid values for the attribute are `view`, `edit`, and `create`. If the attribute is not specified, the field will appear in all modes. If present, the field will only appear for the modes listed. For example, to only show a field in view and edit modes, the `for-mode` attribute would contain `view,edit`.

There are fields that can be optional for an item, and by default they cannot be returned by the server. The `force` attribute can be used to indicate to the form service that it should do everything it can to find and return a definition for the field. An example might be a property defined on an aspect, if the aspect is not applied to the node, a field definition for the property will not be returned. If `force` is `true`, it would indicate that server needs to try and find the property on an aspect in the content model.

hide

The `hide` element normally comes into play when multiple configuration files are combined as it can be used to hide fields previously configured to be shown. The `id` attribute represents the unique identifier for a field, for example `cm:name` that should not be displayed. The optional `for-mode` attribute indicates in which modes the field should not appear. Valid values for the attribute are `view`, `edit`, and `create`. If the attribute is not specified, the field will never appear. If present, the field will be hidden for the modes listed. For example, to hide a field in view and edit modes, the `for-mode` attribute would contain `view,edit`.

The algorithm for determining whether a particular field will be shown or hidden works, as follows:

1. If there is no `field-visibility` configuration (show or hide tags) then all fields are visible in all modes.
2. If there are one or more hide tags then the specified field(s) will be hidden in the specified modes. All other fields remain visible as before.
3. As soon as a single `show` tag appears in the configuration XML, this is taken as a signal that all field visibility is to be manually configured. At that point, all fields default to hidden and only those explicitly configured to be shown (with a `show` tag) will be shown.
4. Show and hide rules will be applied in sequence, with later rules potentially invalidating previous rules.
5. Show or hide rules, which only apply for specified modes, have an implicit element. For example, `<show id="name" for-mode="view" />` would show the name field in view mode and by implication, hide it in other modes.

appearance

The optional `appearance` element controls the look and feel of the controls that make up the form. Unlike the `field-visibility` element, this element will be processed and the information available to custom templates defined with the `view-form`, `edit-form` and `create-form` elements, it is up to those templates whether they use the available data. The configuration of what fields are present and how they appear has been separated to provide the maximum flexibility, and although it might be slightly more verbose, the separation allows the appearance to be defined for fields that are not explicitly mentioned within the `field-visibility` element.

set

The optional `set` element provides the basis of creating groups of fields. The `id` attribute gives the set a unique identifier that other set definitions and fields can refer to. The `parent` attribute allows sets to be nested, and the value should reference a valid set definition, previously defined. The `appearance` attribute specifies how the set will be rendered. The only supported and allowed values are `fieldset` and `panel`. If an `appearance` attribute is not supplied, the set will not be rendered. The `label` and `label-id` attributes provide the title for the set when it is rendered. If neither are supplied, the set identifier is used.

A default set with an identifier of "" (empty string) is always present, and any fields without an explicit set membership automatically belong to the default set. The default set will be displayed with a label of Default.

field

The `field` element allows most aspects of a field's appearance to be controlled from the label to the control that should be used. The only mandatory attribute is `id`, which specifies the field to be customized. However, the field identifier does not have to be present within the `field-visibility` element.

The `label` and `label-id` attributes define the label to be used for the form. If neither attribute is present, the field label returned from the Form Service is used. The `description` and `description-id` attributes are used to display a tool tip for the field. If neither is present, the description returned from the Form Service is used (this could also be empty).

The `read-only` attribute indicates to the form UI generation template that the field should never be shown in an editable form. Finally, the optional `set` attribute contains the identifier of a previously defined set. If the attribute is omitted, the field belongs to the default set.

control

The `control` element allows the control being used for the field to be configured or customized. If present, the `template` attribute specifies the path to the template snippet to use to represent the field overriding the `default-control` template. If the path value is relative, it is relative from the `alfresco` package. If the path value is absolute, it is looked up relative to the `<web-extension>/site-webscripts` package, normally found in the application server shared classes location.

The `control-param` sub-elements provide a mechanism to pass parameters to control templates. This template could either be the one defined locally or the template defined in the `default-control` element for the data type of the field.

constraint-handlers

The `constraint` sub-elements define the JavaScript function to use for checking that fields with constraints are valid before being submitted. The main purpose of this element is to allow aspects of the constraint to be overridden for a particular field. Each attribute effectively overrides the equivalent attribute.

Administrator best practices

Best practice guidelines for Alfresco administrators.

Tips for getting the most out of Alfresco

1. Allow sufficient time to plan your project and identify the most optimal path for you.
2. Benchmark the system you want to use to ensure you can tune it for best performance and high availability before you go live.
3. Ensure customizations occur using the `<extensions>` and `<web-extensions>` directories, and/or `.AMP` files to help smooth upgrade and debugging processes.
4. Discover more about FreeMarker templates. You can create custom views for your spaces, and email templates to fit your organization, among other things.
5. Discover more about web scripts. This requires some, but not extensive, technical knowledge, and is very powerful.
6. Use a space template to create reusable components and enable business processes.
7. Leverage the CIFS interface to easily integrate with existing applications using drag and drop.
8. For Microsoft shops, Microsoft Office integration makes adoption of Alfresco seamless.
9. Email integration provides simple and safe way to store emails inside the Alfresco repository.
10. Coordinate with Alfresco on short-term consulting. This allows you and/or your System Integrator to work with Alfresco on architecture and planning.
11. Take advantage of the support for multiple security protocols, which makes it suitable for large enterprises.
12. Use the [Alfresco Support Portal](#), an Enterprise subscription site that provides downloads, further documentation, and a Knowledge Base.
13. Take advantage of Alfresco training. Get the knowledge and information you need to make your implementation successful.

Common mistakes made by Alfresco administrators

1. Not keeping extended configurations and customizations separate in the shared directory. Do not put them in the configuration root. If you do, you will lose them during upgrades.
2. Not ensuring that the database driver is copied to the application server `lib` directory when installing.
3. Not testing the backup strategy.
4. Making changes to the system without testing them thoroughly on a test and pre-production machine first.
5. Failing to set the `dir.root` property to an absolute path location.
6. Not fully shutting down a running instance of Alfresco, so the next time you try and start it, Alfresco says: Address already in use: `JVM_Bind:8080` (especially on Linux).

Tips for Alfresco administrators

1. Make sure you use a transactional database.
2. Keep your Search indexes on your fastest local disk.
3. Version only what and when you need to.
4. If you find yourself constantly creating the same space hierarchy as well as rules and aspects to them, consider creating a Space template instead.
5. Increase the database connection pool size for large numbers of concurrent users or sessions.

6. Use the System Information to view system properties, such as schema and server versions.
7. Use the Node Browser (searchable by node reference, xpath, or lucene) to view all properties, parent and child nodes, aspects applied, permissions, and associations.

Glossary

The glossary explains the meaning of terms and acronyms that are used in the Alfresco documentation.

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)

A

Glossary terms starting with the letter A.

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)

AAC

See Advanced Audio Coding.

Access Control Entry

An Access Control Entry (ACE) defines a user's access rights to objects or items on a computer or network device.

Access Control List

An Access Control List (ACL) is a list of Access Control Entries (ACE).

ACE

See Access Control Entry.

Acegi security

The Acegi security project was the predecessor to Spring Security, which is a Spring sub-project providing security features to enterprise systems.

ACL

See Access Control List.

ACP

See Alfresco Content Package.

action

A unit of work that is performed on a node. Examples include adding an aspect, copying a node, or emailing notification.

Active Directory

Active Directory (AD) is a Microsoft directory service designed for Windows that helps to manage varied and complex network resources.

ActiveMQ

See Apache ActiveMQ.

Activiti workflow engine

Activiti is a lightweight workflow and Business Process Management (BPM) platform. It provides a fast and reliable BPMN 2 process engine for Java. It is open-source and distributed under an Apache license.

Activiti is based on open standards and is designed to integrate well with Spring applications, like Alfresco. You can learn more about Activiti on the [Activiti](#) website.

activity

An activity in Alfresco refers to the updates to content within a site, including the uploaded files, blogs, discussions, calendars, and the team wiki.

AD

See Active Directory.

Admin Console

The Admin Console is an Enterprise-only application that gives you control over the management and settings of the Alfresco environment. It is a standalone console for managing the administration of the Alfresco repository and comprises separate pages that identify a particular administrative activity or feature.

Admin Tools

Also known as Share Admin Tools. 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.

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.

Advanced Audio Coding

Advanced Audio Coding (AAC) is an audio coding system., designed to replace the MP3 format.

AFTS

See Alfresco Full Text Search.

alf_data

Directory containing binary content and indexes.

alfresco-global.properties

The `alfresco-global.properties` file contains the default configuration settings for Alfresco. The standard global properties file that is supplied with the installers contains settings for the location of the content and index data, the database connection properties, the location of third-party software, and database driver properties.

Alfresco Advanced Versioning Manager

Alfresco Advanced Versioning Manager (AVM) is an advanced store implementation designed to support the version control requirements of large websites and web applications.

Alfresco Community Edition

Alfresco Community Edition is the open source alternative for Enterprise Content Management. Distributed under the LGPLv3 license, it is free to download and use forever. Alfresco Community Edition has limits in terms of scalability and availability and is not supported, which makes it best suited for developers and technical enthusiasts in non-business critical environments.

Previously known as Alfresco Community Labs.

Alfresco Content Package

Alfresco Content Package (ACP) files hold exported information produced when using the Export feature.

Alfresco Day Zero Configuration Guide

The default Alfresco configuration is for single user evaluation, which does not take into account scale and volume. Day Zero configuration is a combination of environment validation and reconfiguration of Alfresco in preparation for high traffic, concurrent transaction, large content sets and product grade hardware. The Alfresco Day Zero Configuration Guide provides instructions on how to do this.

Alfresco Enterprise

See Alfresco One.

Alfresco FTS

See Alfresco Full Text Search.

Alfresco Full Text Search

Alfresco Full Text Search (AFTS) is the search syntax used by Alfresco. It can be used standalone or it can be embedded in CMIS-SQL using the *contains()* predicate function. 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.

Alfresco in the Cloud

Alfresco in the Cloud is a fully managed Software as a Service (SaaS) enterprise content management solution that allows users to securely access their corporate documents and files on any device, from any location.

Alfresco Media Management

Alfresco Media Management provides the capability to transform and add metadata and relationships to your digital media.

Alfresco Mobile

Alfresco Mobile is a free mobile document management app for iOS and Android tablets and smartphones, which enables you to view all your content stored in your Alfresco on-premise or Alfresco in the Cloud accounts.

Alfresco Module Package

An Alfresco Module Package (AMP) is a collection of code, XML, images, CSS, that collectively extend the functionality or data provided by the standard Alfresco repository. An AMP file can contain as little as a set of custom templates or a new category. It can contain a custom model and associated user interface customizations. It could contain a complete new set of functionality.

Alfresco Office Services

Alfresco Office Services (AOS) provides a fully-compatible SharePoint repository that allows the Microsoft Office Suite applications (for example, Word, PowerPoint, Visio and Excel) to interact with Alfresco as if it was SharePoint. AOS enables online editing for Office documents within Alfresco Share and allows users to modify Office files without checking them in and out. Alfresco locks the file while it is being modified and releases the lock when the file is saved and closed. AOS is installed as a core part of Alfresco Enterprise.

Alfresco One

Alfresco One is a combination of Alfresco Enterprise and Alfresco in the Cloud. It is a Hybrid ECM (Enterprise Content Management) solution that combines the efficiency, collaboration, and control of an ECM platform with the agility and flexibility of the cloud.

With Alfresco One, document management, collaboration, and process automation can be synced to the cloud, avoiding the risks associated with uncontrolled consumer file sharing and enabling secure collaboration outside of the firewall.

It is recommended for corporations, governments, and other organizations looking for a production-ready open source ECM solution, with the primary benefit of being a stable, reliable, certified, supported application with warranty and indemnity, with the support of Alfresco and its certified partners.

Alfresco REST API

The Alfresco REST API lets you access content in an on-premise Alfresco repository, and in Alfresco cloud, from your own applications. The API is RESTful and consists of two parts; the standard CMIS API, which lets you manage and access content, and the Alfresco One REST API which lets you manage Alfresco's additional features such as ratings and comments, which are not covered by the CMIS standard.

Alfresco Share

Alfresco Share is a rich web-based collaboration environment for managing documents, wiki content, blogs and more. Share leverages the Alfresco repository to provide content services and uses the Alfresco Surf platform to provide the underlying presentation framework.

Alfresco WAR

See Alfresco Web Application Archive.

Alfresco Web Application Archive

The Alfresco Web Application Archive (WAR) file is used to deploy Alfresco in existing application servers.

Alfresco Web Editor

The Alfresco Web Editor is a Alfresco Surf-based web application that provides in-context editing capabilities for Alfresco repository content. The editor provides a mechanism for non-technical users to make edits to Alfresco content directly within a web page.

AJAX

Asynchronous JavaScript and XML (AJAX) is a method of building fast interactive web applications, which allows the web page to update very quickly by refreshing only the part of the page required by the user.

Amazon RDS

Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.

AMD

See Asynchronous Module Definition.

AMP

See Alfresco Module Package.

AOS

See Alfresco Office Services.

Apache Tika

Apache Tika is metadata detection and extraction software. See [Apache Tika](#) for more information.

Apache Tomcat

Apache Tomcat, also known as Tomcat, is an open source application server developed by the Apache Software Foundation that provides an environment for Java code to run in cooperation with a web server. It allows software implementation of the Java Servlet and JavaServer Pages technologies from Oracle Microsystems.

Tomcat is the default application server used by Alfresco.

Apache ActiveMQ

ActiveMQ is the open source message broker from Apache, which is used by and shipped with Alfresco Analytics.

API

See Application Programming Interface.

Application Programming Interface

Alfresco supports multiple Application Programming Interfaces (APIs) that allow developers to extend and customize the product.

aspect

Aspects allow property and association definitions to be shared across many types of nodes. This means that a cross-cutting feature of an ECM domain model can be encapsulated and applied throughout the rigid part of the model represented by types.

Asynchronous Module Definition

Asynchronous Module Definition (AMD) is a standard for defining modules in JavaScript, such that the module and its dependencies can be asynchronously loaded. This approach is particularly useful in a browser environment.

AtomPub

Atom Publishing Protocol (AtomPub) is an application-level protocol for publishing and editing web resources.

AWE

See Alfresco Web Editor.

B

Glossary terms starting with the letter B.

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)

bcrypt

bcrypt is a key derivation function (or KDF) that derives one or more secret keys from a secret value, such as a master key, a password, or a pass phrase using a pseudo-random function.

BPMN 2.0 standard

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.

breadcrumb

A navigation link that allows you to jump to any sub-level of the path.

C

Glossary terms starting with the letter C.

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)

CachingContentStore

The *CachingContentStore* (CCS) class adds 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.

CAS

See Central Authentication Service.

catalog

Digital Asset Management term. Cataloguing is the process of adding metadata to digital assets (for example, images and video). A catalog comprises the digital assets (i.e. the assets that include descriptive metadata) stored in a repository.

category

A category in Alfresco allows you to organize your content to help you retrieve the content you are looking for. You classify your content items in Alfresco with categories. The categories are organized into related groups to form a hierarchy. You can link a content item to more than one category.

CCC (Community Chart Components)

CCC is the charting library from CTools. CCC provides open-source and customizable chart visualization. You can use CCC2 to create a custom user interface in Alfresco Analytics.

Central Authentication Service

The Central Authentication Service (CAS) is an authentication system that enables enterprise single sign-on to web sites.

CIFS

Microsoft Common Internet File System (CIFS) is a network file system for sharing files across the Internet.

<classpathRoot>

The `<classpathRoot>` is the default directory whose contents are automatically added to the start of your application server's classpath. The location of this directory varies depending on your application server.

CMIS

See Content Management Interoperability Services.

Community Chart Components

See CCC.

codec

Digital Asset Management term. A codec is a program that can encode and decode digital data (audio or video). H.264/MPEG-4 AVC is a codec. FFmpeg supports many common codecs.

composite action

A composite action is a collection of actions that are combined in a designated order to make one unit of work. If any action within the composite fails, then no other actions within this composite are completed. The unit of work can be any size that you determine.

condition

A condition is a test applied to an action used to determine whether the action can be performed.

content

Files or documents made of two main elements: the content itself and information about the content (metadata). For example, documents, video, audio, images, XML, and HTML.

content aspect

See aspect.

Content Delivery Network

A Content Delivery (or Distribution) Network (CDN) delivers contents to users using servers and data centers provided by the CDN operator. Examples are Amazon CloudFront and Brightcove Zencoder.

Content Management Interoperability Services

The Content Management Interoperability Services (CMIS) standard defines a domain model and Web Services and Restful AtomPub bindings that can be used by applications to work with one or more Content Management repositories/systems.

The CMIS interface is designed to be layered on top of existing Content Management systems and their existing programmatic interfaces. It is not intended to prescribe how specific features should be implemented within those CM systems, not to exhaustively expose all of the CM system's capabilities through the CMIS interfaces. Rather, it is intended to define a generic/universal set of capabilities provided by a CM system and a set of services for working with those capabilities.

Source: <http://docs.oasis-open.org/cmis/CMIS/v1.0/cmis-spec-v1.0.html>

Alfresco fully implements both the CMIS 1.0 and 1.1 standards to allow your application to manage content and metadata in an Alfresco repository or in Alfresco in the cloud.

content model

A content model is a collection of related content types and aspects.

content type

Content can be categorized as a type and can be one type at any one time. The type describes the fundamental structure of the content.

content store

By default, Alfresco is configured with two file content stores: the File Content Store and Deleted Content Store.

CTools Community Data Access (CDA)

CTools is a collaboration environment, and CDA (Community Data Access) is a Pentaho plugin that allows you to flexibly access data. CTools CDA can be used in Alfresco Analytics.

cURL

cURL is a command-line tool for getting or sending files using URL syntax. Libcurl is the equivalent library. cURL uses libcurl, which supports most common internet protocols.

D

Glossary terms starting with the letter D.

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)

DAM

See Digital Asset Management.

DAO

See Data Access Object.

dashboard

The Alfresco dashboard is an interactive user interface that presents and organizes information to the user.

dashlet

A dashlet is an application that appears in the Alfresco dashboard that presents information to the user. Users can organize dashlets into different layouts and set keyboard short cuts for each dashlet.

Data Access Object

A Data Access Object (DAO) can be used where data storage is required to provide an abstract interface between the data access and the application logic.

Day Zero

See Alfresco Day Zero Configuration Guide.

Deleted Content Store

The deleted content store saves orphaned files that are removed (nightly, by default) by the content store cleaner.

description document

A description document provides the main configuration elements for a web script. The description document records the URL that initiates the web script, together with a given short name and description, and with authentication and transactional needs.

Digital Asset Management

Digital Asset Management (DAM) is the process of storing and manipulating digital assets, for example, video, audio, and image files.

dir.indexes

The `dir.indexes` folder contains all Lucene indexes and deltas against those indexes.

dir.root

The `dir.root` property is specified in the `alfresco-global.properties` file. It points to the default directory `alf_data`, which contains the content and the Lucene indexes.

disposition schedule

Disposition schedules are a key function of the records management system. The disposition schedule defines the procedures required for maintaining records in the records management system until their eventual destruction or transfer to another location.

A disposition schedule contains one or more steps that define a particular action to be carried out at a date or after an event has occurred.

Document Management

Document Management is the engine that Alfresco uses for storing and retrieving documents, including content contribution and categorization, advanced search, content transformation, versioning and auditing, collaboration, and security.

E

Glossary terms starting with the letter E.

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)

ECM

See Enterprise Content Management.

Embedded API

Embedded APIs are used by custom extensions executed directly against the content application server. Alfresco provides three main embedded APIs - Alfresco Java Foundation API, JavaScript API and Template API.

Enterprise Content Management

Enterprise Content Management (ECM) is a set of technologies used to capture, store, preserve and deliver content and documents and content related to organizational processes. ECM tools and strategies allow the management of an organization's unstructured information, wherever that information exists.

Source: http://en.wikipedia.org/wiki/Enterprise_content_management

Enterprise to Cloud Sync

The Alfresco Enterprise to Cloud Sync feature lets you work with your content without you needing access to the on-premise system. After you synchronize content to the cloud, the on-premise and cloud instances of your content are automatically synchronized with each other whenever either version is updated. This relationship continues until you unsync the content.

Exchangeable Image File Format

Exchangeable Image File Format (EXIF) is a metadata standard used to store technical information about digital images. Software such as [ExifTool](#) allows you to manipulate this metadata.

execution script

It is possible to create web scripts relying only on the scripting capabilities of FreeMarker templating language. However, an optional execution script gives the web script greater flexibility, especially when creating and/or updating Alfresco objects. An execution script may be written in JavaScript or Java.

EXIF

See Exchangeable Image File Format.

<extension>

The `<extension>` directory is where you store files that extend and override the Alfresco default files. When Alfresco is installed, there are sample files in this directory. Many of these files have a `.sample` suffix, which must be removed to activate the file.

For example: for Tomcat, `<extension>` is:`<TOMCAT_HOME>/shared/classes/alfresco/extension/`

Extract, Transform and Load (ETL)

ETL is a process used by Alfresco Analytics to extract data, transform it for analysis, and load it into a data warehouse. In Alfresco Analytics, the tool used for ETL processing is provided by Pentaho, called Kettle.

F

Glossary terms starting with the letter F.

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)

FFmpeg

FFmpeg is an audio and video conversion program. See [FFmpeg](#) for more information.

File System Transfer Receiver

The File System Transfer Receiver (abbreviated to FTR or FSTR) is a mechanism for publishing from Alfresco's Document Management repository to a file system using the replication service. It is delivered as a compressed zip file and includes an embedded Tomcat and an embedded Derby database.

File 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 your document. The files are stored in a directory structure to assist incremental backup. The metadata of your file is stored in the database.

Filing rule

A filing rule is specified in a Smart Folder Template and defines where a new file is stored in the repository, when it is uploaded to a Smart Folder. The filing rule also specifies the type and aspects that are applied to the new file, along with its property values.

Flash Player

Adobe Flash Player is free software used to stream and view video, audio and multimedia and Rich Internet Applications (RIA) on a computer or supported mobile device.

FreeMarker

FreeMarker is a template engine, which is a generic tool to generate text output based on templates. The output can be anything from HTML to auto-generated source code. FreeMarker is a class library that Java programmers can embed into their products.

One or more FreeMarker templates are required to render the web script output in the format for your specific needs. A web script is invoked calling a URL. The URL response is rendered via one of the supplied templates and is chosen at run time based on the required response content-type or status outcome. The template has access to all URL arguments, common repository data entry points and any data items built by the optional execution script.

FSTR

See File System Transfer Receiver.

FTR

See File System Transfer Receiver.

G

Glossary terms starting with the letter G.

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)

GhostScript

Ghostscript is a high quality, high performance Postscript and PDF interpreter and rendering engine. Source: <http://ghostscript.com/FAQ.html>

H

Glossary terms starting with the letter H.

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)

H.264 codec

H.264/MPEG-4 AVC is a commonly used video compression format.

Hazelcast

Hazelcast is an in-memory open source data grid based on Java. By having multiple nodes form a cluster, data is evenly distributed among the nodes. This allows for horizontal scalability both in terms of available storage space and processing power. Backups are also distributed in a similar fashion to other nodes, based on configuration, thereby protecting against single node failure. Source: [Wiki](#)

Hibernate

Hibernate is an object-relational mapping (ORM) library for the Java language, used to abstract the connection to the database that Alfresco is using.

Hybrid ECM

A Hybrid Enterprise Content Management (ECM) solution, like Alfresco One, combines the efficiency, collaboration, and control of an ECM platform with the agility and flexibility of the cloud.

Hyperic

VMWare Hyperic is an application monitoring and performance management tool for virtual, physical, and cloud infrastructures. For more information, see <http://www.hyperic.com>.

I

Glossary terms starting with the letter I.

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)

i18n

Internationalization is abbreviated to the numeronym i18n or I18N, where 18 stands for the number of letters between the first *i* and last *n*. I18n is the means of changing software to meet international requirements in terms of language, regional differences and requirements for other markets.

IDE

See Integrated Development Environment.

IFrame

An IFrame, or Inline Frame, is an HTML document embedded in another HTML document. IFrames are used in Alfresco Analytics to present content in Alfresco Share from Pentaho Analyzer.

ImageMagick

ImageMagick is a software suite to create, edit, and compose bitmap images. It can read, convert and write images in a large variety of formats. Images can be cropped, colors can be changed, various effects can be applied, images can be rotated and combined, and text, lines, polygons, ellipses and Bézier curves can be added to images and stretched and rotated.

Source: <http://www.imagemagick.org/script/index.php>

IMAP

See Internet Message Access Protocol.

ingest

A Digital Asset Management term. Ingestion is the process of supplying media to a Digital Asset Management system (like Alfresco Media Management) and adding metadata (descriptive information) to it to allow for better retrieval when searching.

Integrated Development Environment

An Integrated Development Environment or Interactive Development Environment (IDE) provides a framework for software development, delivered as an application. It provides a single interface with modules such as a code editor, compiler, debugger, version control and a UI builder.

International Press Telecommunications Council**Internet Message Access Protocol**

Internet Message Access Protocol (IMAP) is a method of accessing and storing email on a mail server.

IPTC

The IPTC (International Press Telecommunications Council) is the global standards body for news media.

IPTC standard

The IPTC developed core and extended unformational metadata standards that define a large number of fields to describe images. Alfresco Media Management fully supports this standard and exposes the metadata in Alfresco Share.

J

Glossary terms starting with the letter J.

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)

JAAS

See Java Authentication and Authorization Service.

Java Authentication and Authorization Service

Java Authentication and Authorization Service (JAAS) is a set of Java packages that can be used for user authentication and authorization.

Java Foundation API

The Alfresco Java Foundation API is a set of services providing full access to the capabilities of the Alfresco repository. It is an in-process API meaning that the client must sit within the same process as the repository. For example, Alfresco Share uses this API and is packaged together with the repository in a single .war file for deployment to an application server.

Java Management Extension (JMX) interface

The JMX interface allows you to access Alfresco through a standard console that supports JMX remoting (JSR-160). Example consoles include, JConsole, MC4J, and JManage.

Java Virtual Machine

A Java Virtual Machine (JVM) interprets compiled Java binary code so that the computer processor can run instructions from a Java program.

JavaScript

JavaScript is a scripting language most often used for client-side web development.
JavaScript is a trademark of Oracle.

JavaScript API

The Alfresco JavaScript API allows script writers to develop JavaScript (ECMA Script) 1.6 compatible scripts that access, modify and create Alfresco repository objects. The JavaScript API provides a simple, clean, and object-oriented access to well known Alfresco concepts such as nodes, properties, associations, and aspects.

JavaScript Object Notation

JavaScript Object Notation (JSON) is an open standard format that uses human-readable text to transmit data objects consisting of attribute–value pairs. It is used primarily to transmit data between a server and web application, as an alternative to XML. Source: <http://en.wikipedia.org/wiki/JSON>

JavaServer Faces

JavaServer Faces (JSF) is a Java specification for building component-based user interfaces for web applications. Source: http://en.wikipedia.org/wiki/JavaServer_Faces

JBoss

JBoss is a division of RedHat, who produce open-source middleware software. They produce a range of enterprise and community products.

JConsole

The JConsole graphical user interface is a monitoring tool that complies to the Java Management Extensions (JMX) specification. JConsole uses the extensive instrumentation of the Java Virtual Machine (Java VM) to provide information about the performance and resource consumption of applications running on the Java platform. Source: <http://docs.oracle.com/javase/7/docs/technotes/guides/management/jconsole.html>

Jetty web server

The Jetty web server provides an HTTP server and Servlet container capable of serving static and dynamic content either from a standalone or embedded instantiations. From Jetty7 onwards, the Jetty web server and other core components are hosted by the Eclipse Foundation. Source: <http://eclipse.org/jetty/about.php>

JLAN

Alfresco JLAN, formerly known as the Alfresco Intelligent File System, is open-source software under the GNU Public License (GPL). Alfresco JLAN is an embedded virtual file system that offers a Java client and server implementation of Microsoft Windows's CIFS (Common Internet File System) protocol, allowing content to appear as a shared drive.

JMagick

JMagick is an open source Java interface of ImageMagick. It is implemented in the form of Java Native Interface (JNI) into the ImageMagick API. Source: <http://www.yeo.id.au/jmagick/>

JODConverter

JODConverter automates conversions between office document formats using OpenOffice.org.

JSF

See JavaServer Faces.

JSON

See JavaScript Object Notation.

JVM

See Java Virtual Machine.

K

Glossary terms starting with the letter K.

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)

Kettle

See Pentaho Kettle.

Kofax

Kofax is a leading provider of smart process applications that simplify the business critical First Mile™ of information-intensive customer interactions. Source: <http://www.kofax.com>

L

Glossary terms starting with the letter L.

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)

LibreOffice

LibreOffice is a free and open source office suite, developed by The Document Foundation. It was forked from OpenOffice.org in 2010, which was an open-sourced version of the earlier StarOffice. The LibreOffice suite includes programs for word processing, spreadsheets, slideshows, diagrams and drawings. Source: <http://en.wikipedia.org/wiki/Libreoffice>

log4j

Apache log4j is an open-source logging library for Java, developed as an Apache Software Foundation project. For more information, see <http://logging.apache.org/log4j/1.2/>.

Lucene

Apache Lucene is a high-performance, full-featured text search engine library written entirely in Java. It is a technology suitable for nearly any application that requires full-text search, especially cross-platform. Source: <http://lucene.apache.org/java/docs/index.html>

M

Glossary terms starting with the letter M.

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)

Maven SDK

The Maven Alfresco SDK is a community project that provides an easy to use approach to developing applications and extensions for Alfresco. You can use the SDK to develop, package, test, run, document and release your Alfresco extension project.

MBean

An MBean (managed bean) is a Java object that represents a manageable resource, which is running in the JVM (Java Virtual Machine). Alfresco uses MBeans for JMX-based monitoring and management through read-only, configuration and management beans.

MD4

The MD4 Message-Digest Algorithm is a cryptographic hash function which is considered practically impossible to invert and have many information security applications.

MDX

MDX, also known as Multidimensional Expressions, is a query language for OLAP databases.

metadata

Metadata describes data itself, and provides information about the content of the data.

MIME type

Multipurpose Internet Mail Extensions (MIME) is a standard that extends an email to support processing of non-standard formats, for example, non-text or non-ASCII character sets.

MIME types form a standard way of classifying file types on the Internet. Internet programs such as Web servers and browsers all have a list of MIME types, so that they can transfer files of the same type in the same way, no matter what operating system they are working in.

A MIME type has two parts: a type and a subtype. They are separated by a forward slash (/). For example, the MIME type for Microsoft Word files is *application* and the subtype is *msword*. Together, the complete MIME type is *application/msword*. Source: [What is a MIME type?](#)

MMT

See Module Management Tool.

Model-View-Controller

Model-View-Controller (MVC) is a design pattern for implementing user interfaces. The model is central and consists of application data and business logic. The view is the output, and the controller accepts input and converts it to commands for the model or view.

Module Management Tool (MMT)

The Alfresco Module Management Tool (MMT) helps to install and manage modules packaged as AMP (Alfresco Module Package) files. These AMP files are applied to a target WAR file; for example, share.war.

Mondrian

See Pentaho Mondrian.

Multi-tenancy

Alfresco 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.

Multicast

Multicast is communication from a single sender to multiple recipients.

MVC

See Model View Controller.

MyBatis

MyBatis is open-source software distributed under the Apache License 2.0. It is a Java persistence framework that couples objects with stored procedures or SQL statements using an XML descriptor or annotations. Source: <http://en.wikipedia.org/wiki/MyBatis>

N

Glossary terms starting with the letter N.

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)

namespace

A namespace is an abstract container providing context for the items it holds and allowing items to have the same name (residing in different namespaces). Alfresco namespaces start with <http://www.alfresco.org>.

The top-level namespace sub-divisions are: model, identify an Alfresco data model; view, identify a view of content held in the Alfresco repository; ws, identify an Alfresco web service definition; test, identify a test definition.

node

A node is an overall term for an item of content or a folder. API methods are available to work with a node's comments, tags, and ratings. A node in the content repository must be of a single type, but can be attached to one or more aspects. The aspects are either inherited from its type (as defined in the content model), or can be attached or detached at runtime, allowing a node to dynamically inherit features and capabilities.

noderef

A noderef is the unique identifier for a node. It consists of a store identifier and a unique ID.

NT LAN Manager

NT LAN Manager (NTLM) is a set of Microsoft security protocols that provide network authentication.

NTLM

See NT LAN Manager.

O

Glossary terms starting with the letter O.

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)

OASIS

The Organization for the Advancement of Structured Information Standards (OASIS) is an organization that drives the development, convergence, and adoption of information and web service standards. For more information, see <https://www.oasis-open.org/>.

OLAP (OnLine Analytical Processing) cube

An OLAP cube is a database that is optimised for data warehouse and analytical processing applications. OLAP cubes are used in Alfresco Analytics.

Open JDK

OpenJDK (Open Java Development Kit) is an open-source implementation of the Java SE (Java Platform, Standard Edition) specifications.

OpenLDAP

OpenLDAP is an open-source software implementation of LDAP (Lightweight Directory Access Protocol). For more information, see <http://www.openldap.org>.

OpenSearch

OpenSearch is a collection of simple formats for the sharing of search results. For more information, see <http://www.opensearch.org/Home>.

P

Glossary terms starting with the letter P.

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)

PBCore

A metadata standard for multimedia, developed by the public broadcasting community. See [PBCore](#) for more information.

Pentaho Business Analytics (BA) Server

Pentaho provide business analytics capabilities in the Pentaho BA server. This is used to process information and create reports in Alfresco Analytics. For more information, see <http://docs.alfresco.com/analytics/concepts/analytics-architecture.html>.

Pentaho Kettle

Kettle is the tool used to perform Extract, Transform and Load processing in Alfresco Analytics.

Pentaho Mondrian

Pentaho Mondrian is an Online Analytical Processing server. It processes analytical data in Alfresco Analytics and converts it from SQL to MDX.

Phishing attack

A phishing attack attempts to acquire information, such as user names, passwords, and other financial details by simulating a trustworthy entity.

POM

A Project Object Model (POM) is the fundamental unit of work in Maven. It is an XML file that contains information about the project and configuration details used by Maven to build the project. It contains default values for most projects.

The POM was renamed from *project.xml* in Maven 1 to *pom.xml* in Maven 2. For more information, see <http://maven.apache.org/guides/introduction/introduction-to-the-pom.html>.

Portlet

Many Alfresco installations use Alfresco in conjunction with portals. Portals provide a page view composed of individual windows called portlets, which aggregate and assemble information according to a user's preference.

A portlet consists of specialized query views or browser views. The portlets are designed to work with JSR-168 or JSR-286-compatible portals. The presentation technology is either web scripts or Surf.

properties

A node contains multiple properties, a set of manageable attribute values, defined according type and aspects. For example standard content properties in Alfresco are title, description, creator, and modification date.

proxy

Digital Asset Management term. A proxy file is a copy of a file, that is usually of lower resolution or a variant of the original file. Another word for a proxy file is a rendition. You might view a proxy video file in Alfresco Share that is of lower resolution than the original video file.

Public API

See Alfresco REST API.

Q

Glossary terms starting with the letter Q.

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)

R

Glossary terms starting with the letter R.

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)

RBAC

See Role Based Access Control.

Records Management

Records Management (RM) is an Alfresco solution for managing the life cycle of electronic and other types of records.

regex

A regex (regular expression) is a sequence of characters that form a search pattern. Also written in capitals (REGEX).

Remote API

A remote API allows a client connecting from a separate tier to communicate with the Alfresco content application server. Remote APIs are based on web services and RESTful and CMIS protocols, and are language agnostic, allowing you to develop against these APIs using a range of languages including Java, PHP, Ruby, and .NET. Alfresco provides several remote APIs.

Remote Method Invocation

Remote Method Invocation (RMI) allows you to write distributed objects using Java.

rendition

A rendition is an alternative representation of a content node. Renditions are derived from their source nodes and are usually updated automatically when their source node is updated. In the context of Media Management, renditions are different editions or versions of an asset; for example, an alternative file format. See [proxy](#).

repository

The Alfresco repository is the combination of the content, the indexes, and the database. It contains a collection of Stores. The Alfresco repository architecture is based on the Spring framework. Every part of the Alfresco repository is either a component or a service.

REST architecture

Representational state transfer (REST) is a style of software architecture for distributed hypermedia systems such as the World Wide Web. The Alfresco RESTful API provides access to its services via HTTP. The REST API is built upon the Alfresco web scripts platform allowing customization and extension of the API.

RM

See Records Management.

RMI

See Remote Method Invocation.

Role Based Access Control

Role Based Access Control (RBAC) is a means of restricting user access to a system or systems.

rsync

Rsync is a Unix-based algorithm that quickly synchronizes local and remote files and directories. Rsync sends only file differences across the network instead of complete files.

rule

A rule is an action or a composite action with a rule type. Every rule has a rule type and it characterizes the types of events in the repository that will cause the rule to be triggered.

When a rule is triggered the composite action it represents is executed. A rule type is made up of a number of rule triggers. A rule trigger relates to a repository policy and transforms the information acquired when that policy is fired into the information needed to trigger a rule.

S

Glossary terms starting with the letter S.

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)

Sarbanes-Oxley Act

The Sarbanes-Oxley Act (SOX) is a U.S. law that was passed in 2002 to protect shareholders and the public from accounting errors and fraud in U.S. companies. For more information, see <http://www.soxlaw.com/>.

SDK

An SDK (Software Development Kit) is a package of software or development tools that can be reused by developers to prevent duplication of effort and assist them in creating their application code.

SharePoint

Microsoft SharePoint is a web application platform and framework for collaborating and managing documents. Alfresco Office Services (AOS) provides a fully-compatible SharePoint repository that allows the Microsoft Office Suite applications (for example, Word, PowerPoint, Visio and Excel) to interact with Alfresco as if it was SharePoint.

Simple and Protected GSSAPI Negotiation Mechanism

Simple and Protected GSSAPI Negotiation Mechanism (SPNEGO) is used by a client application to determine which authentication protocol a remote server supports. SPNEGO is a specification defined by [IETF RFC 2478](#).

Single Sign-On (SSO)

Single Sign-On is an authentication process that allows a user to enter a single logon and password to access multiple software applications.

site

A site is a collaborative area in Alfresco Share for a unit of work or a project.

Smart Folder

A Smart Folder displays the results of a query in a folder format. It is “smart”, because there is no physical folder to represent it in the repository and the results are created dynamically. A Smart Folder can also contain a hierarchy of Smart Folders.

Smart Folder Template

A Smart Folder Template is a JSON file that is stored in Alfresco in Repository/Data Dictionary/Smart Folder Templates. When the template is run in a physical folder, a Smart Folder structure is created.

smart space

A smart space has rules defined for how the content is managed in that space. Typical rules include simple workflow, check-in/check-out, and versioning.

Solr

Apache Solr is an open source enterprise search platform from the Apache Lucene project. See <http://lucene.apache.org/solr/> for more information.

SPNEGO

See Simple and Protected GSSAPI Negotiation Mechanism.

Spring

Spring is an open-source application framework for Java/JEE. The Alfresco repository uses the Spring Framework as the core foundation of its architecture. Includes Spring AOP, Spring MVC, Spring Webflow and Spring Security.

SSO

See Single Sign-On.

store

A store is a logical partition within the repository, grouped for a particular automated use. Each store contains a hierarchy of nodes with one root node.

Surf Platform

The Alfresco Surf Platform is a framework for web application and site assembly that bundles a full site construction object model and toolkit for building web applications.

The Surf Platform is built to be very lightweight and can be built as a standalone WAR file. It is designed from the ground up to stand alone within the Web Tier. It includes the Alfresco Web Script Runtime so that application developers can build components, pages, and templates using the rich scripting and FreeMarker templating processing capabilities of Alfresco within in the Web Tier.

The Surf Platform Freemarker Template and JavaScript API provides an extension to the existing Alfresco Freemarker Template and JavaScript API capabilities, including additional, root-scoped objects.

SurfBug

SurfBug is an Alfresco debugging tool built into Alfresco Surf that displays a variety of information about the various components on a Surf page. As well as providing a visual indication of component location on the page, information about each component is provided, including relevant file names, properties, IDs, and sub-component details.

T

Glossary terms starting with the letter T.

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)

tag

A tag is a non-hierarchical keyword or term assigned to a piece of information.

template

A template is a document that can be applied to an object or objects (for example, one or more documents) to produce another document (see FreeMarker or XSLT template files).

tenant

A tenant is a customer, company or organization. Alfresco supports a single-instance, single-tenant (ST) environment where each tenant runs a single instance that is installed on one server or across a cluster of servers. Multi-tenancy (MT) features are also supported.

thumbnail

A thumbnail is a small image that provides a preview of the file.

Tika

See Apache Tika.

TinyMCE

TinyMCE is a platform independent web based Javascript HTML WYSIWYG editor control released as Open Source under LGPL (GNU Lesser General Public License). Source: <http://www.tinyMCE.com>

transcoding

Transcoding is the process of converting from one format to another, for example, for video or audio files. When you view a video file in Alfresco Share using Media Management, the file is transcoded to reduce the size and resolution of the video.

type

A node has one content type to define its class in the repository; such as content, space or custom object (wiki, forum, and so on). These types are defined in one of the content models of the repository.

U

Glossary terms starting with the letter U.

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)

V

Glossary terms starting with the letter V.

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)

W

Glossary terms starting with the letter W.

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)

WAR file

The Alfresco Web application ARchive (WAR) file is for deployment in existing application servers.

Web Content Management

Web Content Management (WCM) is an Alfresco product for the rapid deployment of web content, allowing users to create, develop, and maintain content for websites.

Web Editor Framework

Web Editor Framework (WEF) is the client-side JavaScript framework on which the Alfresco Web Editor is built. It is built using YUI and can be extended easily. New tabs and buttons can be packaged and dropped into the framework.

web script

Web scripts allow you to extend or customize Alfresco and make it accessible to other tools and applications. Alfresco supplies several example web scripts. Web scripts are bound to an HTTP method and custom URL, and are activated by URL calls. There are two types of web script: data and user interface. Data web scripts return data, and user interface web scripts build user interfaces within the Alfresco web client, or within third-party applications.

Web Script Runtime (WSR)

A Web Script Runtime is the environment (that is, the host) within which a web script is executed. There are three Web Script Runtimes available: Servlet Runtime (HTTP Access), JSR-168 Runtime (Portlet Access), and JSF Runtime (JSF Component Access). New runtimes may be plugged-in allowing web scripts to be re-used and re-hosted in a number of different applications.

Web Services API

A WSDL based API providing standard remote access to the Alfresco repository from any client environment, for example, PHP, .NET, Java.

<web-extension>

The `<web-extension>` directory is where you store files that extend and override the Alfresco default files for Alfresco Share. When Alfresco is installed, there are sample files in this directory. Many of the files have a `.sample` suffix, which must be removed to activate the file.

For example: for Tomcat, `<web-extension>` is:`<TOMCAT_HOME>/shared/classes/alfresco/web-extension/`

WebDAV

Web-based Distributed Authoring and Versioning. A protocol that allows users to edit and manage files on remote web servers.

WEF

See Web Editor Framework.

Whitelist

Whitelist is a generic name for a list of email addresses or IP addresses that are considered to be spam free.

workflow

A workflow is a work procedure and workflow steps that represent the activities users must follow in order to achieve the required outcome. Alfresco provides two different types of workflow: simple and advanced. Simple workflow defines content rules for a space. Advanced workflow provides two out-of-the-box workflows (Review and Approve; Adhoc Task).

WSDL

WSDL (Web Service Definition Language) is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. Source: <http://www.w3.org/TR/wsdl>

WSR

See Web Script Runtime.

X

Glossary terms starting with the letter X.

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)

Xalan

The Apache Xalan project develops libraries and programs that transform XML documents using XSLT (Extensible Stylesheet Language Transformation) stylesheets. They have two sub-projects running for Xalan Java and Xalan C++ implementations.

XAM

XAM (eXtensible Access Method) is a standard for fixed storage. It simplifies the management of fixed content by removing dependencies. XAM was created by the Storage Networking Industry Association (SNIA).

XCOPY

XCOPY (extended copy) is a command for copying files or directories across a network.

XMP

XMP is the Extensible Metadata Platform provided by Adobe. For more information, see www.adobe.com/products/xmp.html.

Y

Glossary terms starting with the letter Y.

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)

YUI library

The YUI (Yahoo! User Interface) library is an open-source JavaScript library for building interactive web applications.

Z

Glossary terms starting with the letter Z.

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [X](#) | [Y](#) | [Z](#)

Zero Day

See Alfresco Day Zero Configuration Guide.

Copyright

Copyright 2016 by Alfresco and others.

Information in this document is subject to change without notice. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Alfresco. The trademarks, service marks, logos, or other intellectual property rights of Alfresco and others used in this documentation ("Trademarks") are the property of Alfresco and their respective owners. The furnishing of this document does not give you license to these patents, trademarks, copyrights, or other intellectual property except as expressly provided in any written agreement from Alfresco.

The United States export control laws and regulations, including the Export Administration Regulations of the U.S. Department of Commerce, and other applicable laws and regulations apply to this documentation which prohibit the export or re-export of content, products, services, and technology to certain countries and persons. You agree to comply with all export laws, regulations, and restrictions of the United States and any foreign agency or authority and assume sole responsibility for any such unauthorized exportation.

You may not use this documentation if you are a competitor of Alfresco, except with Alfresco's prior written consent. In addition, you may not use the documentation for purposes of evaluating its functionality or for any other competitive purposes.

If you need technical support for this product, contact Customer Support by email at support@alfresco.com. If you have comments or suggestions about this documentation, contact us at documentation@alfresco.com.

This copyright applies to the current version of the licensed program.

Disclaimer

While Alfresco has used commercially reasonable efforts to ensure the accuracy of this documentation, Alfresco assumes no responsibility for the accuracy, completeness, or usefulness of any information or for damages resulting from the procedures provided. Furthermore, this documentation is supplied "as is" without guarantee or warranty, expressed or implied, including without limitation, any warranty of fitness for a specific purpose.