



White Paper

Alfresco Backup and Disaster Recovery

Copyright 2014 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.

This copyright applies to the current version of the licensed program.

Document History

VERSION	DATE	AUTHOR	DESCRIPTION OF CHANGE
0.1	11-Apr-13	Toni de la Fuente	Initial version
0.2	16-Oct-13	Toni de la Fuente	Added backup procedures skeleton by David Almazan
0.3	30-Oct-13	Toni de la Fuente	Final draft document
0.4	3-Nov-13	Toni de la Fuente	Added Solr backup URLs
0.5	7-Jan-14	Toni de la Fuente	Made some changes and corrections after Mike Farman review.

Table of contents

INTRODUCTION	1
COSTS AND BUSINESS IMPACT	1
BACKUPS METHODS.....	2
DISASTER RECOVERY	2
BACKUP OVERVIEW AND STRATEGY	5
SCHEDULING AN ALFRESCO BACKUP.....	5
<i>Solr scheduled backup job</i>	5
<i>Lucene scheduled backup job</i>	7
<i>Other scheduled jobs to consider on a backup strategy</i>	9
BACKUP PROCEDURE	10
WHAT MUST BE BACKED UP	10
STATIC DATA	10
DYNAMIC DATA	10
WHY IS ORDER IMPORTANT?	11
COLD BACKUP	11
WARM BACKUP	12
HOT BACKUP	12
<i>Backing up the Database</i>	13
RESTORE.....	16
COLD BACKUP RESTORE	16
HOT BACKUP RESTORE	16
<i>Notes for Lucene restore</i>	17
SINGLE FILE RESTORATION.....	17
RESTORING CONTENT FILES	18
THIRD PARTY RESOURCES FOR ALFRESCO BACKUP AND DISASTER RECOVERY	20

Introduction

To talk about backup is to talk about business continuity. Multiple types of hazards can occur while a system is operating, hardware or software failures, data corruption, natural disasters, human errors, performance issues, etc. Also planned or unplanned interruptions like maintenance tasks or upgrades must be taken into account for business continuity.

Since the protection, integrity, privacy of contents and the business continuity of service matter to Alfresco, we are aware of the importance of backup and recovery. This document is about helping you become familiar with the main components of Alfresco in order to develop a backup and recovery strategy for Alfresco Enterprise 4 and higher. It is widely and incorrectly thought that security is solely a task or a product, but when we talk about security we are speaking about a continuous process. Therefore we must also take into consideration the Security Plan of the organization, the organization's Contingency Plan and their Disaster Recovery Plan.

Depending on the particular needs of each project the number of backed up components may vary in terms of number of servers or locations. This document will cover all components, procedures and methodologies for all needed components in Alfresco in order to be able to backup and restore any production environment.

Costs and business impact

To plan a backup and recovery process the first task to address is to figure out how much data loss is acceptable and how that loses might impact our business.

The time between backups is called Recovery Point Objective (RPO) that means the maximum amount of data we can lose. On the other hand the time taken to restore the application and make it available is called Recovery Time Objective (RTO). RPO has to be defined by your business. RTO depends on procedures and ability to recover data and get the application back online. Both objectives must be based on the financial impact to the business when the system is unavailable. That financial factor includes loss of business and reputation due to the downtime and lack of system availability.

Once RPO and RTO are known, you can calculate how often you need to perform backups and how long the intervals should be between each backup process. If a system has to be available 24x7, there is no backup window. In this case a hot backup has to be performed.

The total amount of storage to backup and number of copies to maintain has to be planned before hand. The number of full and incremental backups to keep must be included in the backup policy as well.

Backups methods

There are different backup levels:

- **Full backup:** when we are doing a complete copy of all of the files. This backup tends to be slow and is typically performed as first backup or at a regular interval of time.
- **Incremental:** when only the changes from the last backup are backed up. Faster backup than cumulative; could be slower to restore than cumulative because there could be more files to restore.
- **Cumulative or Differential:** only copy changes after the most recent full backup. This method may be slower than incremental but is usually faster to restore.

Types of backup techniques depending on the system availability:

- **Cold:** a complete backup of all components of Alfresco with the entire system shut down.
- **Warm:** backup performed while some services of Alfresco are unavailable, i.e.: set the repository to read only mode.
- **Hot:** backup performed while the system is running and potentially being used.

Other concepts to take into account:

- **Backup window:** time to do it. With Alfresco it depends on the type of backup chosen.
- **Backup rotation:** time period while doing incremental backups between periodic and full backups: daily, weekly or monthly are most common.
- **Backup destination:** Network device (NAS, Amazon S3, SCP, FTP, etc.), SAN, disk to tape, disk to disk. Each backup method can be oriented for different solutions and depending on the amount of data to backup. For disaster recovery consider using a remote backup method.

Disaster recovery

Disaster recovery is the process, policies and procedures related to preparing for recovery or continuation of technology infrastructure, critical to the organization, after a natural or human-induced disaster.

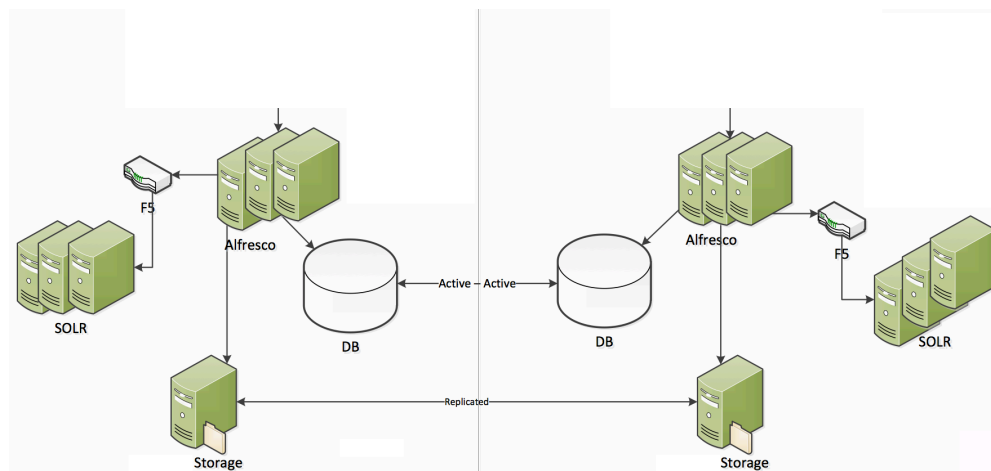
You can use several technologies to protect application data, including resilient storage, mirroring and replication. A server cluster and disaster recovery environment needs one or more of these technologies.

The disaster recovery configurations must consider the original deployments or origin systems and how to back them up to a target deployment. Depending on the service we want to provide from the disaster recovery environment, we can consider the next levels:

- **Disaster recovery deployment with full capacity:** backups and configuration are replicated to a target deployment with existing hardware and software that has the same capacity as the original one.
- **Disaster recovery deployment with reduced capacity:** backups and configuration are replicated to a target deployment with existing hardware and software but with less capacity than the original one.
- **Data disaster recovery only:** Backups and configuration are replicated without hardware or software deployed.

The Alfresco subscription includes a stand-by disaster recovery environment. The procedures and concepts covered by this guide can be used to implement your **disaster recovery** plan as an **asynchronous** procedure.

A disaster recovery plan and environment can be performed in different ways. As in the diagram below the copy or replication of the content storage and database is made by vendor specific solutions.



Vendor specific solutions for database or storage replication are out of the scope of this guide. However major database vendors provide replication solutions for their products:

- **Postgresql:** http://wiki.postgresql.org/wiki/Streaming_Replication
- **MySQL:** <http://dev.mysql.com/doc/refman/5.5/en/replication.html>
- **Oracle:** <http://www.oracle.com/technetwork/database/features/data-integration/index.html>

On the storage side most of the major storage vendors have replication solutions also for CAS solutions¹.

¹ <http://www.xenit.eu/alf2cas>

² <http://localhost:8080/share/page/console/admin-console/application>

Backup overview and strategy

It is important to keep a strict policy on backup and recovery procedure tasks. With Alfresco, the logical order to carry out a hot backup is the indexes (backing up from the automated backup copy), then a backup of the data base and finally a backup of the content store and other required folders such as: logs, configuration files, audit files, etc.

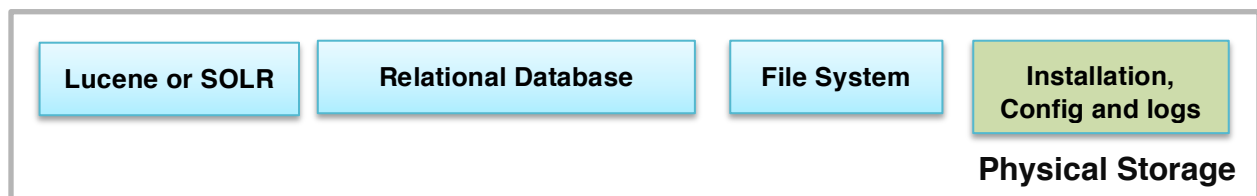
For a cold backup the order is not important.

Keep in mind that depending on the kind of index server the automatic backup will be done by default at 2AM for Lucene or 2AM and 4AM for Solr cores.

Consider using file system snapshots for the Content Store and snapshot-like backup for DB to reduce the lock time and the application availability.

Testing recovery often as a preventive measure. Add a checked Alfresco recovery procedure to your contingency plan.

All Alfresco components to backup:



Scheduling an Alfresco backup

First take into account RTO and RPO. Alfresco has a default configuration for backup of its indexes. These tasks are done by scheduled jobs and depend on the type of indexes chosen:

Solr scheduled backup job

Live SOLR indexes are located in the following paths specified by the property `${data.dir.store}` `${dir.root}/solr/workspace-SpacesStore/conf/solrcore.properties` and `${dir.root}/solr/archive-SpacesStore/conf/solrcore.properties`

```
${dir.root}/solr/workspace/SpacesStore/index  
${dir.root}/solr/archive/SpacesStore/index
```

You can set the Solr backup using different methods:

Unless we are doing a cold backup we must copy the Solr from its own backup. Its backup configuration is already planned; these are the default values that can be modified by adding them into `alfresco-global.properties`:

```
solr.backup.alfresco.cronExpression=0 0 2 * * ?  
solr.backup.archive.cronExpression=0 0 4 * * ?
```

```

solr.backup.alfresco.remoteBackupLocation=${dir.root}/solrBackup/alfresco
solr.backup.archive.remoteBackupLocation=${dir.root}/solrBackup/archive
solr.backup.alfresco.numberToKeep=3
solr.backup.archive.numberToKeep=3

```

Configuration above means that the Alfresco core (workspace://SpacesStore) is done every day at 2:00AM creating its files in `${dir.root}/solrBackup/alfresco`. The Alfresco archive core (archive://SpacesStore) is done every day at 4:00AM. In both cases 3 copies of the last 3 backups are kept. Information stored in those folders are what we have to copy to our backup target. If we have a backup strategy we may only need to keep one backup and also make the backup in a different periodicity.

Using the Alfresco Admin Console

In order to change the backup properties using the “Administration Console” (prior to 4.2 thru Share² and the Admin Console³ in 4.2 and later) you will need to enter the “Administration Console”. Click on the “Search Service”:

You can then specify the backup location and how often the backups occur.

Backup Settings

Main Store

Backup Location:

The location where the index backup is stored on the Solr server.

Backup Cron Expression:

A unix-like expression, using the same syntax as the cron command, that defines when backups occur. The default value is `0 0 2 * * ?` meaning the backup is performed daily at 02.00.

Backups To Keep:

The number of backups to keep (including the latest backup).

Archive Store Properties

Backup Location:

The location where the index backup is stored on the Solr server.

Backup Cron Expression:

A unix-like expression, using the same syntax as the cron command, that defines when backups occur. The default value is `0 0 4 * * ?` meaning the backup is performed daily at 04.00.

Backups To Keep:

The number of backups to keep (including the latest backup).

Using the JMX Client to manually force or create the Solr backup

With a JMX client like jconsole:

For Alfresco Solr core:

² <http://localhost:8080/share/page/console/admin-console/application>

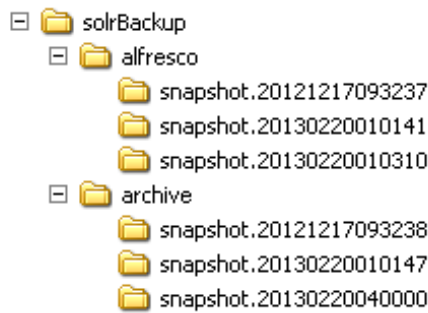
³ <http://localhost:8080/alfresco/service/enterprise/admin/admin-searchservice>

- JMX MBeans > Alfresco > Schedule > DEFAULT > MonitoredCronTrigger > search.alfrescoCoreBackupTrigger > Operations > executeNow()

For Archive Solr core:

- JMX MBeans > Alfresco > Schedule > DEFAULT > MonitoredCronTrigger > search.archiveCoreBackupTrigger > Operations > executeNow()

You can check the Solr backups on the file system. These will contain the date appended to the folder name for easy reference:



Using the Solr admin panel to manually force or create the Solr backup

SOLR can also be backed up direct using next URLs:

For the alfresco core and only keep 1 backup:

https://localhost:8443/solr/alfresco/replication?command=backup&location=/opt/alfresco/alf_data/solrBackup/alfresco&numberToKeep=1

For the archive core and only keep 1 backup:

https://localhost:8443/solr/archive/replication?command=backup&location=/opt/alfresco/alf_data/solrBackup/archive&numberToKeep=1

In order to do the backup from the command line, you may use “curl” and run it like this (see comment about pem cert below):

- curl -k --cert /opt/alfresco/alf_data/keystore/browser.pem:alfresco
“https://localhost:8443/solr/alfresco/replication?command=backup&location=/opt/alfresco/alf_data/solrBackup/alfresco&numberToKeep=1”
- curl -k --cert /opt/alfresco/alf_data/keystore/browser.pem:alfresco
“https://localhost:8443/solr/archive/replication?command=backup&location=/opt/alfresco/alf_data/solrBackup/archive&numberToKeep=1”

Please, note that “curl” does not support p12 certificates therefore you need to convert the default browser.p12 to browser.pem by running (password is alfresco):

- openssl pkcs12 -in /opt/alfresco/alf_data/keystore/browser.p12 -out /opt/alfresco/alf_data/keystore/browser.pem -nodes

Lucene scheduled backup job

Lucene indexes are located in the following paths:

```

${dir.root}/lucene-indexes

```

You can set the Lucene backup using different methods:

Unless we are doing a cold backup we must copy the Lucene backup, whose backup configuration already is scheduled. These are the default values that can be modified by adding them into alfresco-global.properties:

```

index.backup.cronExpression=0 0 3 * * ?

```

```

dir.indexes.backup=${dir.root}/backup-lucene-indexes

```

Configuration above means that the whole Lucene index backup is done every day at 3:00AM creating its files in `${dir.root}/backup-lucene-indexes`. Unlike the Solr backup, only one backup is kept. Information stored in this folder is what we have to copy to our backup target.

Using the Alfresco Admin Console

In order to change the backup properties using “Administration Console” (prior to 4.2 thru Share⁴ and the Admin Console⁵ in 4.2 and later), you will need to enter the “Administration Console”. Click on the “Search Service”:

You can then specify the backup location and how often the backups occur.

Backup Settings

Index Backup Directory:

The location where the index backup is stored on the Lucene server.

Backup Cron Expression:

A unix-like expression, using the same syntax as the cron command, that defines when backups occur. The default value is `0 0 3 * * ?` meaning the backup is performed daily at 03.00.

Using the JMX Client to manually force or create the Lucene backup

With a JMX client like jconsole:

For Lucene:

- JMX MBeans > Alfresco > Schedule > DEFAULT > MonitoredCronTrigger > indexBackupTrigger > Operations > executeNow()

A reliable way to know if the Lucene index backup has finished for the day is to check the subdirectories of `${dir.root}/backup-lucene-indexes` for their timestamp. This folder should contain the following sub-directories:

⁴ <http://localhost:8080/share/page/console/admin-console/application>

⁵ <http://localhost:8080/alfresco/service/enterprise/admin/admin-searchservice>

- Archive
- Locks
- System
- User
- Workspace

During the index backup, these sub-directories will be deleted. These directories are not created until the index backup is completely finished. Make sure the timestamp on the backup sub-directories show time values later than the last backup completed.

Other scheduled jobs to consider on a backup strategy

An orphan node is a node that has been cleaned from the trashcan. After 14 days as an orphan (without any metadata on the DB) the Content Store Cleaner job sends all orphan nodes to `$(dir.root)/contentstore.deleted`. This process runs every day at 4:00AM and can be modified by:

- `system.content.orphanCleanup.cronExpression=0 0 4 * * ?`

To prevent waiting 14 days until an orphan node is sent to `$(dir.root)/contentstore.deleted` you can modify:

- `system.content.orphanProtectDays=XX`

The deleted content store folder must be deleted manually. Alfresco keeps all files there and it never deletes or cleans that folder.

You can decide if content should be removed from the system immediately after being orphaned (cleaned from the trashcan) by setting this option to true:

- `system.content.eagerOrphanCleanup=false`

Backup procedure

What must be backed up

This section describe the overall concept of “Backup and Restore” of an Alfresco Repository.

There are two types of data that need to be considered, static and dynamic.

- **Static Data:** includes software components that do not change through the usage of the alfresco repository.
- **Dynamic Data:** is data that changes as a result of using Alfresco.

Static Data

- Operating System (not covered by this procedure).
- **Application Server Install and configuration files.**
- Database installation files (if it is in same server, not recommendable).
- **Alfresco extensions (customizations)*.**
- 3rd Party applications used by Alfresco (Open Office, ImageMagick, SWFTools).

***ATTENTION:** Pay special attention to your customizations (such as AMPs, jar files or any other custom file that may be in the extension directory). Add them to your backup procedure because that will save you time when performing a recovery.

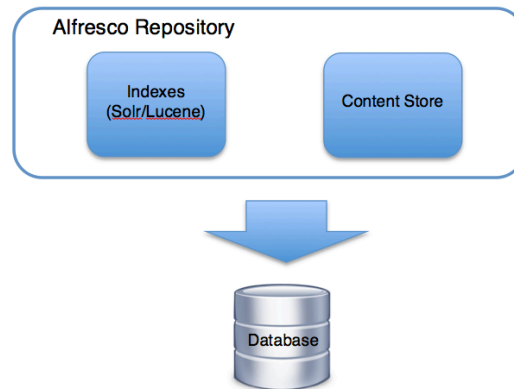
Dynamic Data

- Alfresco Indexes (Solr or Lucene)
- Database (RDBMS data files, table spaces, archive logs and control files).
- Alfresco Content Stores – the default and any other additional store used by Content Store Selector. Content Store Deleted is not required.

For purposes of this document, we will focus on the backup and restore procedure of the dynamic data of Alfresco. Please consider making a backup of the static data for quick and easy recovery.⁶

These are shown in the following diagram:

⁶ All data backup including static data is covered by Alfresco Backup and Recovery Tool in <http://blyx.com/alfresco-bart> *not supported by Alfresco Support Services



Why is order important?

During the backup process of an Alfresco System the following order should be taken into account:

- Indexes should be backed up first. If new rows are added in the database after the Lucene/SOLR backup is done, it's still possible to regenerate the missing Lucene/SOLR indexes from the SQL transaction data.
- Database backup should be performed next. If you have a SQL node pointing to a missing file, that node will be an orphan. If you have a file without a SQL node data, that file will not be included in the backup.

Cold Backup

This is the simplest and safest form of backup, since we do not have to deal with updates while the backup is taking place.

Here are the necessary steps to perform a cold backup of the system. The order from 2 to 4 is not important:

1. Stop the whole Alfresco system (application server or all cluster servers if applicable).
2. Backup Lucene or SOLR.
3. Backup the Alfresco DB.
4. Backup the Content Store, and any other Content Store if applicable.
5. Consider backup of all your static, installation and customization files.
6. Start Alfresco.

Content Store use to be located in $\${dir.root}/contentstore$ or any other path given by the property $\${dir.contentstore}$ in `alfresco-global.properties`.

By default, the $\${dir.root}$ contains both the content and indexes. Being server stopped a copy of the files are enough as backup (not the index backup itself).

It is possible to backup just the content and do a full reindex when a backup of the Content Store and DB is restored (for both Lucene and Solr). As said before, note that you can also separate content and indexes into different directories and not necessary inside `#{dir.root}` (that use to be `alf_data`).

In a cold backup you must next copy the default directories:

Contentstore:

- `#{dir.root}/contentstore`

Index if Lucene:

- `#{dir.root}/lucene-indexes`

Indexes if Solr

- `#{dir.root}/solr/workspace/SpacesStore/index`
- `#{dir.root}/solr/archive/SpacesStore/index`

Weakness of this procedure is that users cannot use the application while the backup is taking place. This procedure must be done when users or applications don't need access to Alfresco. Night hours being the most common time frame for doing this procedure.

Advantage of this procedure is the reliability and consistency of the backup data. No locks and differences between any data group (indexes, DB and Content Store) should be found.

Warm Backup

For a manual backup, the Alfresco system administrator has an optional workaround to perform a cold backup without stopping Alfresco server but setting it up as read only and forcing the index backup before copying.

With a JMX client like `jconsole`:

- JMX: Alfresco > Configuration > `sysAdmin` > default > Attributes > `server.allowWrite=false`

Once the system is in Read Only mode you can proceed doing the backup as in the Hot Backup procedure. Once the backup is done set the property back before: `server.allowWrite=true`

The advantage of this procedure is that applications and users can still consume content from Alfresco but without writing privileges.

Hot Backup

In an Alfresco system, the ability to support hot backup is dependent on the hot backup capabilities of the database product Alfresco is configured to use.

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 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.

In order to successfully perform a hot backup you should follow these steps in order:

1. Consider backing up all your static, installation and customization files.
2. Make sure you have the index backup stored:
 - a. Being `/${dir.root}/backup-lucene-indexes` if Lucene is used
 - b. Or `/${dir.root}/solrBackup/alfresco` and `/${dir.root}/solrBackup/archive` for Solr
3. Make sure that any job that generates index backup is not running while performing the database backup. See schedule backup section.
4. Backup the database Alfresco is configured to use, using the database backup tools (see below Backing up the Database section).
5. Backup specific subdirectories in the Alfresco `dir.root`.
 - a. `/${dir.root}/contentstore`
 - b. `/${dir.root}/cachedcontent` if applicable
 - c. Any other content store

Backing up the Database

The ability to support a hot backup is dependent on the hot backup capabilities of the database product. These are some of the requirement for a consistent hot backup:

- It needs a tool that can "snapshot" a consistent version of the database (it must capture a transactionally consistent copy of all tables).
- The "snapshot" operation should either operate without locking the database, or complete extremely quickly.
- You should ensure that any backup procedure of the database is validated before putting it into a production environment.

Considerations for MySQL

As a recommendation, in order to backup MySQL, you may use the following sample command being `server=localhost`, `user=alfresco`, `password=alfresco`, db name=`alfresco`:

- `mysqldump --single-transaction -h localhost -u alfresco -palfresco alfresco > alfresco.sql`

The option "`--single-transaction`" creates a consistent snapshot by dumping all tables in a single transaction. While a `--single-transaction` dump is in process, to ensure a valid dump file (correct table contents and binary log position), no other connection should use the following

statements: ALTER TABLE, DROP TABLE, RENAME TABLE, TRUNCATE TABLE, as consistent snapshot is not isolated from them. Option automatically turns off --lock-tables. In this case it will guarantee a consistent Database backup.

The command above will generate the file "alfresco.sql", which contains a dump of the tables. These files are readable using a text editor, so spotting corruption becomes easier. Copy that file to a safe place.

There are third party applications for MySQL backup⁷.

Considerations for PostgreSQL

Default PostgreSQL tools like pg_dump are logical and do not produce file-system-level backups. To create incremental Point-in-Time Recovery (PITR) and file-system-level backups of Postgresql please refer to WAL⁸.

For a standard command to get a dump of the existing alfresco database:

- `pg_dump --host=localhost --username=alfresco --file=alfresco.dump alfresco -w`

There are third party applications for PostgreSQL backup⁹.

Considerations for Oracle

For Oracle there exists a variety of tools to perform backups. Most commonly used are exp/imp, Datapump and RMAN.

Oracle's export (exp)¹⁰ and import (imp) utilities are used to perform logical database backup and recovery. When exporting, database objects are dumped to a binary file. The export/import utilities are commonly used to perform backup and recovery but only for small databases, say < +50GB. For larger databases use RMAN instead. Exp/imp also eliminates database fragmentation (export, drop and re-import tables) and detect database corruption.

Datapump¹¹ is a server based bulk data movement infrastructure that supersedes the old import and export utilities. The old export/ import tools are still available, but do not support all Oracle 10g and 11g features. The new utilities are named expdp and impdp.

⁷ <http://www.percona.com/software/percona-xtrabackup>

⁸ <http://www.postgresql.org/docs/9.1/static/continuous-archiving.html>

⁹ <http://www.pgbarman.org/>

¹⁰ http://www.orafaq.com/wiki/Import_Export_FAQ

¹¹ <http://www.orafaq.com/wiki/Datapump>

RMAN¹² Recovery Manager (RMAN) is an Oracle provided utility for backing-up, restoring and recovering Oracle Databases. RMAN ships with the Oracle database and doesn't require a separate installation.

¹² http://www.orafaq.com/wiki/Recovery_Manager

Restore

The restore procedure depends on the requirements and tools used to make the backup. Also all paths required to recover a backup are related to your configuration as said before. This is why doing backup of your configuration and customizations are also mandatory.

Remember that time taken to restore the application and make it available again is called Recovery Time Objective (RTO).

Cold backup restore

Once we have all components (static and dynamic) you can proceed to restore all sets.

NOTE: To ensure the restore works properly and to avoid scripts and configuration try to respect same path as you had before.

1. Be sure all Alfresco related services are stopped (except database because you will need to get access to it in order to recover the backup.)
2. Restore installation and customizations. If you don't have a copy of your installation do a clean installation and then restore all dynamic components. You should have a backup of your customizations and apply them before continuing. If your are using Solr and SSL do a backup of your `/${dir.root}/keystore`.
3. Copy the existing copy of your indexes to the new index folder (or same path as before.) Remember that doing a restore from a cold backup you can copy the existing indexes instead of the backed up one.
4. Based on your database vendor restore the database backup following their instructions.
5. Copy the existing Content Store backup into the proper directory based on your `alfresco-global.properties` configuration.
6. Start Alfresco and test.

Hot backup restore

The procedure to restore a hot backup files is pretty similar to cold backup restoration but with the difference on the indexes. Once we have all components (static and dynamic) you can proceed to restore all sets.

1. Be sure all Alfresco related services are stopped (except database because you will need to get access to it in order to recover the backup.)
2. Restore installation and customizations. If you don't have a copy of your installation do a clean installation and then restore all dynamic components. You should have a backup of your customizations and apply them before continuing. If your are using Solr and SSL do a backup of your `/${dir.root}/keystore`.
3. Copy the backup indexes (Lucene or Solr) to the original index folder, remember to copy the existing backup as is and in the same folder as live indexes (not backup folders). Nevertheless if no indexes are found Alfresco will try to do a full index (this depends on

your index configuration.) If you have old files on the existing indexes folder remove them or copy them to a temporary place.

4. Based on your database vendor restore the database backup following their instructions. Most of cases you should have an empty database before restoring a backup.
5. Copy the existing Content Store backup into the proper directory based on your `alfresco-global.properties` configuration. If you have old files on the existing content store folder remove them or copy them to a temporary place.

NOTE: To ensure the restore works properly and to avoid scripts and configuration try to respect same path as you had before.

Notes for Lucene restore

Alfresco will (by default) detect that the indexes are stale, and incrementally reindex just that content that has changed since the last backup was performed. As the size of your content set grows, the time savings from performing incremental reindexing rather than full reindexing will become greater and greater (incremental reindexing is typically measured in minutes, whereas full reindexing can take hours or days for large content sets.) In such a case make sure that `"index.recovery.mode"` is set to AUTO in `"alfresco-global.properties"`.

Note that in addition to full restorations, the backup sets created via either the cold or hot backup procedures described above can also be used to restore just the Lucene indexes. This is useful in cases where the repository itself does not need to be restored but for some reason the Lucene indexes are stale and rebuilding them from scratch is undesirable.

Single file restoration

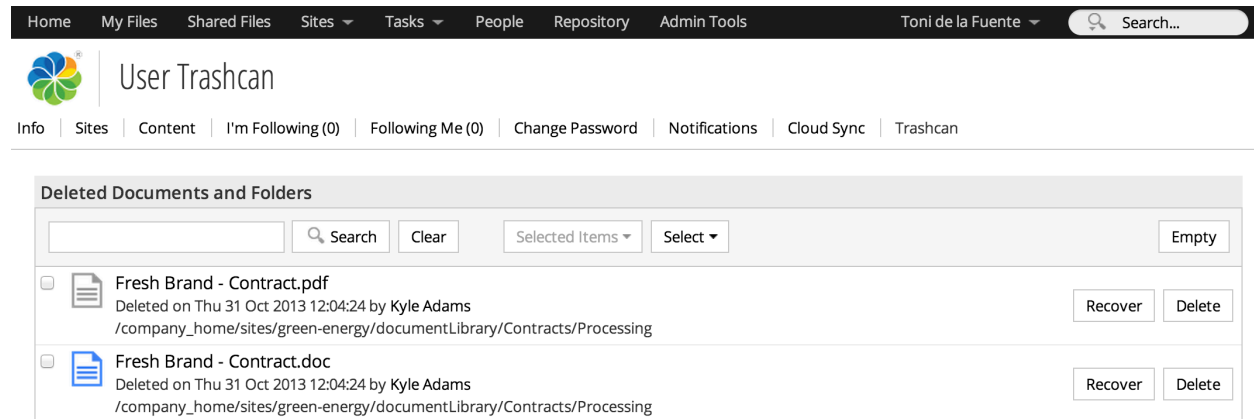
One of the most demanded features of backup and Alfresco is the selective restoration of content. This can be possible by default if we know how Alfresco works when content is deleted.

Alfresco has a trashcan that can be manage by the administrator and since version 4.2 users also can have access into it (see screenshot below.) Trashcan can be found in the "My Profile" section then "Trashcan."

When a node is deleted in Alfresco by a user, this content will remain in the trashcan forever unless the user or the administrator clean the trashcan. If the trashcan is cleaned by the user or administrator, content are marked as orphans and after 14 days the content is moved to `contentstore.deleted`. (See section "Other scheduled jobs to consider on a backup strategy" for more details).

Knowing this allows the end user or the administrator to perform single file restorations. This may be possible depending on the company policy and maintenance of deleted files.

If single file restoration is required and the file is no longer in the trashcan you have to restore an Alfresco installation and then restore the single file from that restore.



The screenshot shows the Alfresco User Trashcan interface. At the top, there is a navigation bar with links for Home, My Files, Shared Files, Sites, Tasks, People, Repository, and Admin Tools. The user's name, Toni de la Fuente, and a search bar are also visible. Below the navigation bar, the main content area is titled "Deleted Documents and Folders". It contains a search bar, a "Clear" button, and a "Selected Items" dropdown menu. There are two entries in the trashcan:

- Fresh Brand - Contract.pdf**: Deleted on Thu 31 Oct 2013 12:04:24 by Kyle Adams. Path: /company_home/sites/green-energy/documentLibrary/Contracts/Processing. Buttons: Recover, Delete.
- Fresh Brand - Contract.doc**: Deleted on Thu 31 Oct 2013 12:04:24 by Kyle Adams. Path: /company_home/sites/green-energy/documentLibrary/Contracts/Processing. Buttons: Recover, Delete.

Restoring Content Files

If some inconsistencies exist between your database and your content store, in some cases it may be necessary to restore a content file from an object that exists in the database but no longer in the content store file system. This is a typical error received when the content is not available.



The Web Script `/alfresco/s/api/node/content/workspace/SpacesStore/56e3e126-86ea-46d9-8ffd-c580c4e0d0ca/File A.pdf` has responded with a status of 404 - Not Found.

404
Description: Requested resource is not available.

Message: 01210002 Unable to locate content for node ref workspace://SpacesStore/56e3e126-86ea-46d9-8ffd-c580c4e0d0ca (property: {http://www.alfresco.org/model/content/1.0}content)

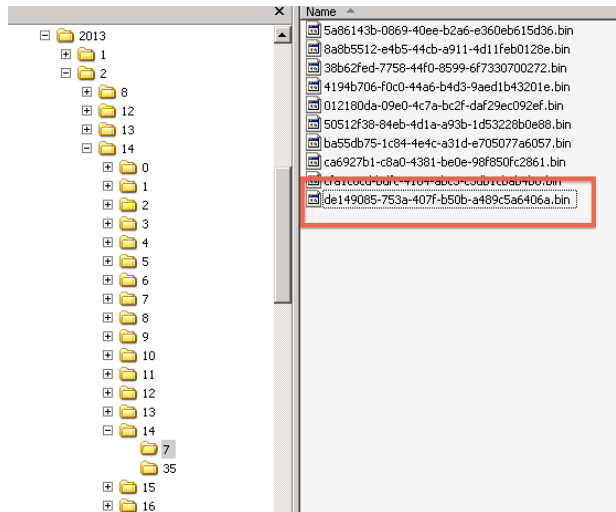
Exception: org.springframework.extensions.webscripts.WebScriptException - 01210002 Unable to locate content for node ref workspace://SpacesStore/56e3e126-86ea-46d9-8ffd-c580c4e0d0ca (property: {http://www.alfresco.org/model/content/1.0}content)

Getting the content from a backup and using the "Node Browser" can fix this. You can view the "Node Browser" to check the internal properties for an object, including the path in the content store.

1. You will need to locate the store path from the document doing a "NodeRef" search in the "Node Browser". In the example above "workspace://SpacesStore/56e3e126-86ea-46d9-8ffdc580c4e0d0ca"

Properties			
Name	Type	Value	Residual
cm:name	d:text	File A.pdf	false
sys:node-dbid	d:long	1900	false
sys:store-identifier	d:text	SpacesStore	false
sys:locale	d:locale	en_US	false
cm:content	d:content	contentUrl=store://2013/2/14/14/7/de149085-753a-407f-b50b-a489c5a6406a.bin mime-type=application/pdf size=68961 encoding=UTF-8 locale=en_US id=352	false
cm:title	d:mltext	G:\WPA_APWEPMD- D~1\...\WT2220\T2220- ~1.FRP Printing	false
cm:modified	d:datetime	14 Feb 2013 14:24:19 GMT-0500 (EST)	false
sys:node-uuid	d:text	56e3e126-86ea-46d9-8ffd-c580c4e0d0ca	false
cm:created	d:datetime	14 Feb 2013 14:07:56 GMT-0500 (EST)	false
sys:store-protocol	d:text	workspace	false
cm:creator	d:text	admin	false
cm:modifier	d:text	admin	false

2. You will then need to locate the file in the backup of the content store.



3. You will then need to copy the located file into the right folder in the content store file system.

Third party resources for Alfresco backup and disaster recovery

As an example, scripting for both Linux and Windows can be done for backups.

Sample script based on Robocopy¹³ for Alfresco installed on Windows with MySQL (it doesn't include the index backup copy):

```
REM ***** Backup *****
REM Backup a mySQL Alfresco instance
REM This script expects a standard, standalone installation
REM You should change the paths to reflect the specifics of your installation
REM
REM You should shutdown your Alfresco Server but leave mySQL running when executing
this script
REM

@echo off
@echo Copy shared extensions to .\backup\shared\classes\alfresco
robocopy .\tomcat\shared\classes\alfresco .\backup\shared\classes\alfresco /E /MIR
/log:.\backup\shared-alfresco-extensions-backup.log

@echo Copy shared lib to .\backup\shared\lib
robocopy .\tomcat\shared\lib .\backup\shared\lib /E /MIR /log:.\backup\shared-lib-
backup.log

@echo Dumping alfresco MySQL database to .\backup\alf_data\alfresco.dmp
mysql_x64\bin\mysqldump -u alfresco --password=alfresco alfresco >
.\alf_data\alfresco.sql
@echo Backing up .\alf_data
title Backing up .\alf_data
robocopy .\alf_data .\backup\alf_data /E /MIR /log:.\backup\alf_data-backup.log

REM Now copy .\backup somewhere else!

REM ***** TODO Restore *****
REM Copy alf_data to where it should be
REM Create the empty Alfresco DB
REM import the DB using the following
REM mysql_x64\bin\mysql -u alfresco --password=alfresco < alfresco.sql
```

¹³ <http://en.wikipedia.org/wiki/Robocopy>

Sample script for cold backup on Linux with MySQL:

```
#!/bin/bash

Days=`date +%Y%m%d-%H%M%S`;
Alfresco_root="/opt/alfresco";
Alfresco_repository="$Alfresco_root/alfresco_repository/";
Alfresco_indexes="$Alfresco_root/alfresco_indexes/";
Alfresco_backup_dir="/opt/Alfresco_Backup_$Days";
user_mysql="alfresco";
user_password="alfresco";
alfresco_db="alfresco";

echo "=== Stopping Alfresco ==="
echo $Alfresco_root/alfresco.sh stop
echo sleep 10;
echo "=== Making the directory backup $Alfresco_backup_dir ==="
echo mkdir $Alfresco_backup_dir;
echo cd $Alfresco_backup_dir;
echo "=== Make Mysql Backup ==="
echo mysqldump --user=$user_mysql --password=$user_password $alfresco_db -q --single-transaction -R -t --disable-keys --complete-insert| bzip2 > $alfresco_db_$Days.sql.backup

echo "=== Creating Aflresco Repository Backup ==="
echo tar cjfv alfresco_repository_$Days.tar.bz2 $Alfresco_repository
echo "=== Creating Alfresco Indexes Backup ==="
echo tar cjfv alfresco_indexes_$Days.tar.bz2 $Alfresco_indexes
echo "=== Restating Alfresco ==="
echo $Alfresco_root/alfresco.sh start
```

Apart from the examples described above, there is full featured backup tool based on Duplicity¹⁴ and Linux oriented a Backup and Recovery tool made for Alfresco called “**Alfresco BART**”¹⁵ (Alfresco Backup and Recovery Tool). It supports a variety of essential features like

¹⁴ <http://duplicity.nongnu.org/>

¹⁵ <https://github.com/toniblyx/alfresco-backup-and-recovery-tool>

full and incremental backup, encryption, compression, FTP, Amazon S3, SCP or local backups, restore wizard, full and single repository file restore and more.