

Alfresco Repository

Directory Structure



Top Level Directories

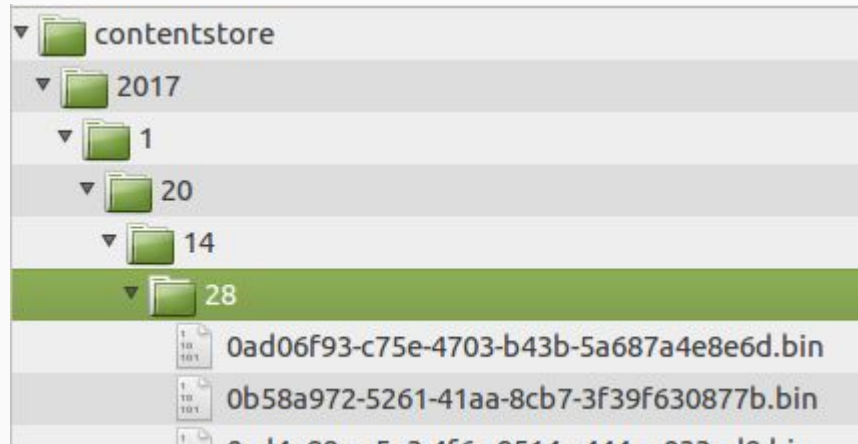
- After installing Alfresco the following directory structure can be seen:
 - **alf_data** – content files, index files, and sometimes db data files
 - **amps** – AMPs that should be installed into alfresco.war
 - **amps_share** – AMPs that should be installed into share.war
 - **bin** – contains MMT tool and script to run it
 - **common** – libs like imageMagic (on Windows you have a imagemagick dir instead)
 - **java** – JVM if installed full package (version 8)
 - **libreoffice** – for transformation (version 5.2)
 - **modules** - dropzone for share and platform JAR modules
 - **postgresql** – the database if installed full package
 - **scripts** – start script
 - **solr4** - Apache Solr Configuration
 - **tomcat** – Apache Tomcat app server (version 7)

/alf_data

- **alf_data** location is set in alfresco-global.properties in tomcat/shared/classes
 - **contentstore** – the main location for all live content
 - **contentstore.deleted** – deleted content (ends up here after 14 days)
 - **audit.contentstore** – audit data if auditing is turned on
 - **cachedcontent** – location for cached content
 - **oouser** - open office direct user
 - **postgresql** – data files for the PostgreSQL database
 - **keystore** – certificate stores
 - **solr4** – contains the Apache Solr index files (cores)
 - **solr4Backup** – backup of solr indexes, 3am every night

alf_data/contentstore

- Files stored according to date and time
- File ID is UUID, change .bin to file format and open



alf_data/contentstore continued

- To find a specific document in the content store you have to first go into the **Node Browser**
- There look up the item and its `cm:content` property
- This property looks something like this:
 - `contentUrl=store:/ /2017/2/2/18/1/62781911-635d-4366-80dc-13d4a3e5e4fe.bin|mimetype=application/pdf|size=375530|encoding=utf-8|locale=en_GB_`
- And in this case the PDF file would be found under:
 - `alfresco/alf_data/contentstore/2017/2/2/18/1`

alf_data/contentstore.deleted

- Whenever you delete a file from the Alfresco repository via any of the user interfaces it is not physically deleted from disk, instead it's moved to the Archive Store via metadata changes
- Then when the file is deleted via **My Profile | Trashcan** it is moved from the `contentstore` to the `contentstore.deleted` directory (after 14 days)
- Files live in `contentstore.deleted` forever

alf_data/solr4

- All content in the repository is indexed via the Lucene index engine that is embedded in the Apache Solr web application
 - Unless it is not possible to convert content to text, or
 - Unless indexing has been turned off for the content type
- The index files are kept in the `solr4/index/archive` and `solr4/index/workspace` directories
- Note that the configuration for the indexes (cores) are kept in the `alfresco/solr4/` directory, and not under `alf_data`

alf_data/postgresql

- If a full installation of Alfresco has been done then the PostgreSQL database data files ends up in this directory
- By default the database is called `alfresco`
- Have a look in the `tomcat\shared\classes\alfresco-global.properties` file for information about what db name, username, password etc that was setup during installation

alf_data/keystore

- This directory contains the certificate store and the related trust store, it contains default Alfresco certificates
- In a production environment new certificates would have to be issued and installed
- The files have the following meaning:
 - **keystore** is the secret key keystore, containing the secret key used to encrypt and decrypt node properties
 - **ssl.keystore** is the repository keystore, containing the repository private/public key pair and certificate
 - **ssl.truststore** is the repository truststore, containing certificates that the repository trusts
 - **browser.p12** is a pkcs12 keystore generated from ssl.keystore that contains the repository private key and certificate for use in browsers such as Firefox

/amps and /amps_share

- The Alfresco Module Packages (AMPs) that are used to extend Alfresco functionality can be found in the following two directories:
 - **/amps** – put amps here that should customize alfresco.war
 - **/amps-share** - put amps here that should customize share.war
- The directory will contain any modules that was selected during the installation
- You can drop your own AMPs into this directory and run the **bin\apply_amps.sh** script to install them
- You can also use the module management tool (MMT) directly
- Contained in the **bin\alfresco-mmt.jar**

/modules

- Alfresco extension modules also come in the form of JARs, that can be dropped into the following two directories:
 - **/platform** – put JARs here that should extend the alfresco.war
 - **/share** - put JARs here that should extend the share.war
- Does not alter the WAR files like AMPs
- Requires a restart of application server

/tomcat

- This is the main application server directory, contains Apache Tomcat
 - **bin** – contains scripts to start Tomcat
 - *Windows*: contains the tomcat7w.exe to manage service settings such as JVM stuff:
`tomcat7w.exe //ES//alfrescoTomcat`
 - Contains script to set JVM memory: `setenv.sh` and `setenv.bat`
 - **conf** – application server configuration (i.e. `server.xml`), the `conf/Catalina/localhost` has information about the web application contexts (`alfresco.xml`, `share.xml`, `solr4.xml`)
 - **lib** – contains JDBC driver, if you are not installing a full package, or use for example MySQL, then you would have to copy the JDBC driver here plus create database
 - **logs** – log files

/tomcat continued

- **shared** – configuration goes here, lives over re-deployments
 - `shared/classes/alfresco-global.properties` – main config file for Alfresco (i.e. `alfresco.war`)
 - `shared/classes/alfresco/web-extension/share-config-custom.xml` – main configuration file for `share.war`
- **webapps** – all the Alfresco web applications
- **work** – JSPs compiled into Servlets
- **scripts** – contains the `ctl.sh` for starting and stopping Tomcat (on *Windows* you will have `servicerun.bat`)