Setup ACS6.x and SOLR6 using distribution package step by step

ACS6.x is mainly based on containers which makes use of popular container platform Docker

Many of us are not familiar with containerization setup and may not be ready to use it yet. In previous versions we had option to use installers and it used to setup everything seamlessly. With ACS6.x we don't have installer option available and it is expected to leverage the use of containerization technologies.

But there are ways to setup ACS6.x in the same ways as we used to do for previous versions i.e. using distribution package provided by Alfresco.

There are other ways to setup as well, have a look at this solution. It was demonstrated during DevCon 2020 by Jeff Potts.

https://www.slideshare.net/jpotts/no-docker-no-problem-automating-installation-and-config-with-ansible

https://github.com/jpotts/ansible-alfresco

The purpose of this post is to document all the steps at one place when setting up ACS6.x manually using distribution package.

ACS 6.2.0-ga is the latest stable version available for community to use it and we will use the same version here.

What we need before we start doing setup?

1- ACS-6.2.0-ga package (alfresco-content-services-community-distribution-6.2.0-ga)
2- ASS-1.4.0 package (alfresco-search-services-1.4.0)
3- Java: Oracle jdk-11.0.1 or later/Open JDK 11.0.1 or later
4- Tomcat: Tomcat 8.5.43
5- ActiveMQ: ActiveMQ v5.15.8 (Optional). Mandatory only when using transformation services
6- DB: PostgreSQL 11.4
7- ImageMagick: ImageMagick v7.0.10
8- Libreoffice: LibreOffice v6.3.5

Checkout this documentation for additional details on supported platforms

Platform:

1- Windows 10 x64
2- Oracle JDK 11.0.4 x64

Type of deployment:

1- ACS, Share and SOLR6 on same machine
2- Setup Without SSL

Let's start gathering all the required pre-requisites listed above.

Download ACS 6.x:

https://download.alfresco.com/cloudfront/release/community/201911-GA-build-368/alfresco-content-services-community-distribution-6.2.0-ga.zip

Download ASS 1.4:

https://download.alfresco.com/cloudfront/release/community/SearchServices/1.4.0/alfresco-search-services-1.4.0.zip

Download and Install Oracle JDK 11.0.4:

https://www.oracle.com/java/technologies/javase/jdk11-archive-downloads.html

https://www.oracle.com/webapps/redirect/signon?nexturl=https://download.oracle.com/otn/java/jdk/11.0.4+10/cf1bbcbf431a474eb9fc550051f4ee78/jdk-11.0.4_windows-x64_bin.zip

Note: Make sure you set the JAVA_HOME environment variable (on windows). It is the installation path of jdk. E.g. JAVA_HOME=C:\Program Files\Java\jdk-11.0.4

Download Tomcat 8.5.43 binary package:

https://archive.apache.org/dist/tomcat/tomcat-8/v8.5.43/bin/apache-tomcat-8.5.43-windows-x64.zip

Note: Make sure ports 8005, 8080, 8443, AJP port 8009 are open and not in use already. These are default ports used for tomcat. If you have these ports already in use, make sure you change the ports accordingly in <TOMCAT_INSTALLATION>/conf/server.xml.

Download ActiveMQ v5.15.8 binary package, it will be used for transformation services down the line:

https://archive.apache.org/dist/activemq/5.15.8/apache-activemq-5.15.8-bin.zip

Download and Install PostgreSQL 11.4:

https://get.enterprisedb.com/postgresql/postgresql-11.4-1-windows-x64.exe

Note: Make sure port 5432 is open and not already in use. Port 5432 is default for postgres to get db connection. If you have this port already in use, make sure you select a different port and use the same while configuring alfresco-global.properties.

Download ImageMagick v7.0.10:

https://imagemagick.org/download/binaries/ImageMagick-7.0.10-45-Q16-HDRI-x64-dll.exe

Download LibreOffice v6.3.5:

https://downloadarchive.documentfoundation.org/libreoffice/old/6.3.5.2/win/x86_64/LibreOffice_6.3.5.2_Win_x64.msi (installer)

Optional Alfresco module packages (amps)-Useful for admins/developers:
https://repo1.maven.org/maven2/org/orderofthebee/support-tools/support-tools-repo/1.1.0.0/support-tools-repo-1.1.0.0-amp.amp

https://repo1.maven.org/maven2/org/orderofthebee/support-tools/support-tools-share/1.1.0.0/support-tools-share-1.1.0.0-amp.amp
https://repo1.maven.org/maven2/de/fmaul/javascript-console-repo/0.6/javascript-console-repo-0.6.amp
https://repo1.maven.org/maven2/de/fmaul/javascript-console-share/0.6/javascript-console-share-0.6.amp

Now, we have completed all the pre-requisites. Let’s start the setup.

Follow the steps given below:

Prepare database:

Assuming you have already installed Postgres, launch “pgAdmin4” from your postgres install location *<PostgresInstallLocation>\PostgreSQL\11\pgAdmin 4\bin\pgAdmin4.exe*

In my case: C:\PostgreSQL\11\pgAdmin 4\bin\pgAdmin4.exe

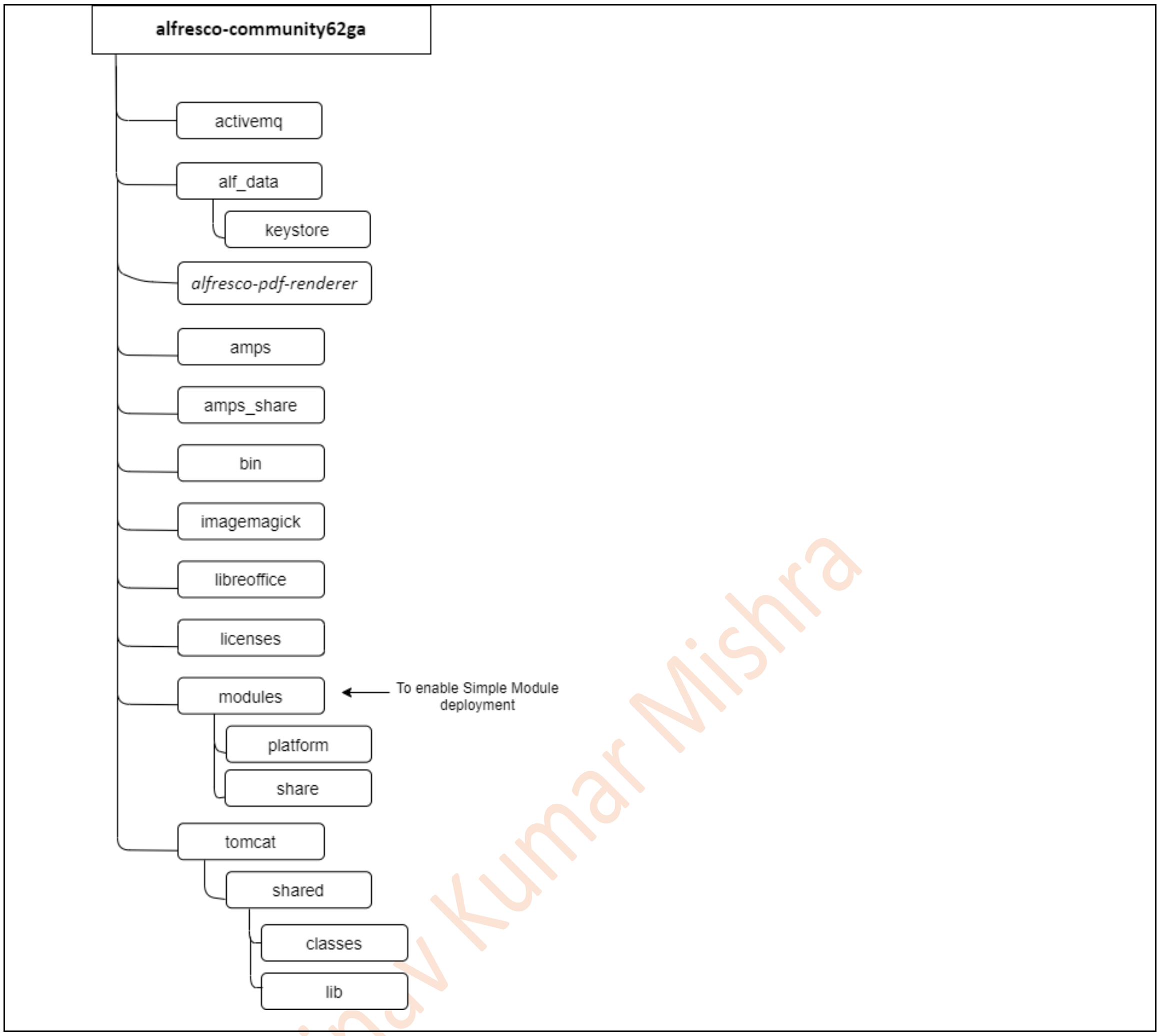
- Create a new user named “alfresco” and set a password which will be used by alfresco content service to connect to database. Here we will be using username as “alfresco” and password “alfresco”.
- Create a new database named : “alfresco”, make sure you select default encoding which is “utf8”.
- Grant all permissions for user “alfresco” on database “alfresco”.
- Alternatively, you can open the query tool and execute following SQL queries:

```
create role alfresco LOGIN password 'alfresco';
create database alfresco encoding 'utf8';
grant all on database alfresco to alfresco;
```

Prepare acs6.2.0-ga setup directory structure:

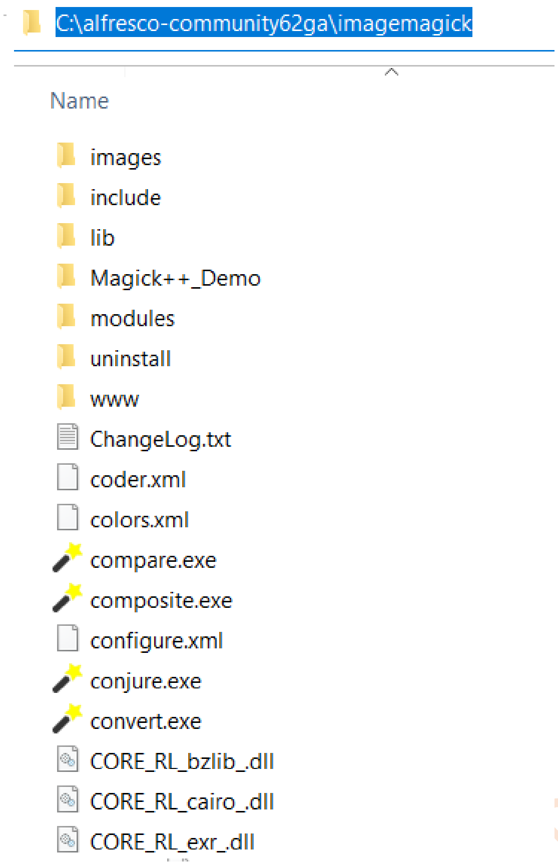
Create directory structure as shown below on your windows drive (e.g. C:\):

You can create the same structure anywhere else, make sure all the config paths are mapped accordingly. For this post all the setup and configurations will be done in C:\ drive.



Install dependencies:

1- Install imagemagick in the directory setup as per the structure given above. E.g. install image magic in ***C:\alfresco-community62ga\imagemagick*** directory (use custom installation option to select the install directory). Alternatively you can download the binary package and extract the image magic binaries in the folder given above as per your choice.

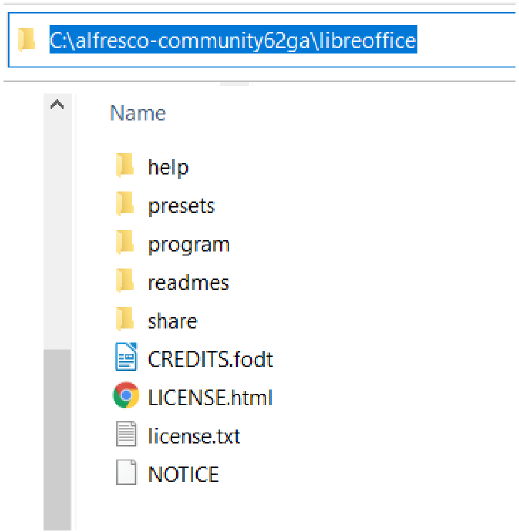


C:\alfresco-community62ga\imagemagick

Name

images
include
lib
Magick++_Demo
modules
uninstall
www
ChangeLog.txt
coder.xml
colors.xml
compare.exe
composite.exe
configure.xml
conjure.exe
convert.exe
CORE_RL_bzlib_.dll
CORE_RL_cairo_.dll
CORE_RL_exr_.dll

2- Install libreoffice in the directory setup as per the structure given above. E.g. install libreoffice in ***C:\alfresco-community62ga\libreoffice*** directory (use custom installation option to select the install directory).

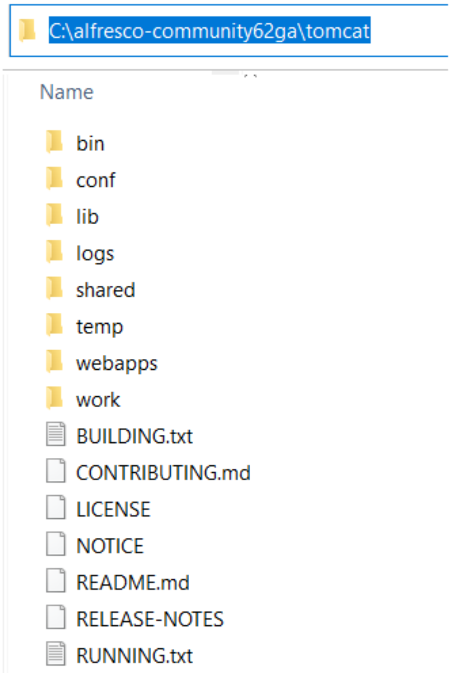


C:\alfresco-community62ga\libreoffice

Name

help
presets
program
readmes
share
CREDITS.fodt
LICENSE.html
license.txt
NOTICE

3- Install Tomcat, get the tomcat binary package (apache-tomcat-8.5.43-windows-x64.zip) downloaded as per the steps given above and extract its contents directly under **tomcat** directory**.** For example: ***C:\alfresco-community62ga\tomcat*** directory



C:\alfresco-community62ga\tomcat

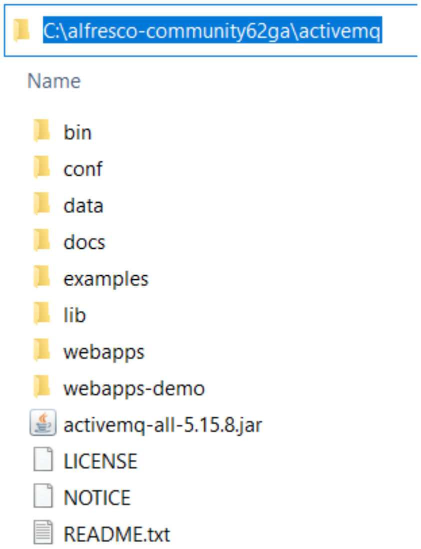
Name

bin
conf
lib
logs
shared
temp
webapps
work
BUILDING.txt
CONTRIBUTING.md
LICENSE
NOTICE
README.md
RELEASE-NOTES
RUNNING.txt

Note: Make sure ports 8005, 8080, 8443, AJP port 8009 are open and not in use already. These are default ports used for tomcat. If you have these ports already in use, make sure you change the ports accordingly in <TOMCAT_INSTALLATION>/conf/server.xml.

4- Install ActiveMQ, get the activemq binary package (apache-activemq-5.15.8-bin.zip) downloaded as per the steps given above and extract its contents directly under **activemq** directory**.** For example: ***C:\alfresco-community62ga\activemq*** directory. Note that activemq may not be used at this time as we will be setting up with legacy transformation service but keeping the message broker setup ready for future use in case you would like to setup transformation service.

Note: Make sure ports 61616, 5672, 61613, 1883, 61614 and 8161 are open and not in use already. These are default ports used for activemq. If you have these ports already in use, make sure you change the ports accordingly in <AMQ_INSTALLATION>/conf/activemq.xml and jetty.xml as needed.



C:\alfresco-community62ga\activemq

Name

bin
conf
data
docs
examples
lib
webapps
webapps-demo
activemq-all-5.15.8.jar
LICENSE
NOTICE
README.txt

Unzip alfresco-content-services-community-distribution-6.2.0-ga.zip:

- Set **ALF_HOME** environment variable, this will be helpful for alfresco scripts such as apply_amps.bat, clean_tomcat.bat etc. to work properly. You always option to manually change the paths in these files as needed. (Optional step)
  - ***ALF_HOME=C:\alfresco-community62ga***

- Copy contents from "C:\Downloads\alfresco-content-services-community-distribution-6.2.0-ga\alf_data\keystore" into "C:\alfresco-community62ga\alf_data\keystore"

  Following files will be copied:
  - **keystore ->** Secret key keystore containing the secret key used to encrypt and decrypt node properties.
  - **keystore-passwords.properties ->** Contains password protecting the keystore entries.

- Copy contents from "C:\Downloads\alfresco-content-services-community-distribution-6.2.0-ga\keystore" into "C:\alfresco-community62ga\alf_data\keystore"

  Following files will be copied:
  - **CreateSSLKeystores.txt ->** 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).
  - **readme.txt ->** Text file containing information about other files in a directory.
  - **generate_keystores.bat ->** Windows batch file for generating secure keys for SOLR communication.

Note: These files will be used to setup Data Encryption (not supported on community version) and SSL for SOLR and Alfresco communication (we are not setting up SSL here so we are just copying the files but we will not be using it for now)

- Copy contents from “C:\Downloads\alfresco-content-services-community-distribution-6.2.0-ga\alfresco-pdf-renderer\alfresco-pdf-renderer-1.1-win64.tgz” into “C:\alfresco-community62ga\alfresco-pdf-renderer” and extract the **alfresco-pdf-renderer-1.1-win64.tgz**

  You would have following file in “C:\alfresco-community62ga\alfresco-pdf-renderer” directory:
  - alfresco-pdf-renderer.exe [Full path: C:\alfresco-community62ga\alfresco-pdf-renderer\alfresco-pdf-renderer.exe]

- Copy contents from “C:\Downloads\**alfresco-content-services-community-distribution-6.2.0-ga\amps**” into “C:\alfresco-community62ga\amps”. We will be installing this alfresco module packages (amps) at later stage.

  Following files will be copied:
  - **alfresco-share-services.amp ->** Alfresco share service module containing extensions for search, datalist, sample sites patch etc.
    Note: This amp installation is mandatory otherwise several functionalities including share search may not work. And you will also see following warning when you login to share: ***Alfresco is running without Share Services. See your System Administrator for more details.***

- Copy the contents from “C:\Downloads\alfresco-content-services-community-distribution-6.2.0-ga\bin” into “C:\alfresco-community62ga\bin” folder.

  Following files will be copied:
  - **alfresco-mmt.jar ->** Alfresco Module Management Tool (MMT), A java library which supports alfresco module package installation/uninstallation/listing and preview operations etc.
  - **apply_amps.bat ->** Utility to install amps kept in “amps” and “amps_share” directory. It will install amps related to alfresco on alfresco.war (keeps the backup of original war file) and amps related to share on share.war (keeps the backup of original war file)

    For more details visit: https://docs.alfresco.com/community/tasks/amp-install.html

  - **clean_tomcat.bat ->** Windows batch file for cleaning out temporary application server files from previous installations.
  - **Win32NetBIOS.dll ->** DLL handles the connection between the native CIFS server and Alfresco Content Services.
  - **Win32NetBIOSx64.dll ->** DLL handles the connection between the native CIFS server and Alfresco Content Services.
  - **Win32Utils.dll ->** DLL handles the connection between the native CIFS server and Alfresco Content Services.
  - **Win32Utilsx64.dll ->** DLL handles the connection between the native CIFS server and Alfresco Content Services.

- Copy the contents from “C:\Downloads\**alfresco-content-services-community-distribution-6.2.0-ga\licenses**” into “C:\alfresco-community62ga\licenses” folder. It contains files that have information about license terms used by alfresco including all third party licenses.
- Copy the contents from “C:\Downloads\**alfresco-content-services-community-distribution-6.2.0-ga\web-server\conf**” into “C:\alfresco-community62ga\tomcat\conf” folder. **conf** directory contains Catalina repository and Share xml files.
- Copy the contents from “C:\Downloads\**alfresco-content-services-community-distribution-6.2.0-ga\web-server\lib**” into “C:\alfresco-community62ga\tomcat\lib” or “C:\alfresco-community62ga\tomcat\shared\classes\lib” folder. **lib** directory contains the PostgreSQL JDBC jar file (postgresql-42.2.6.jar).
- Copy the contents from “C:\Downloads\**alfresco-content-services-community-distribution-6.2.0-ga\web-server\shared\classes**” into “C:\alfresco-community62ga\tomcat\shared\classes” folder. **shared** directory includes the Alfresco Content Services configuration files:

  - **/classes/alfresco-global.properties.sample ->** Sample alfresco global properties file, which is used for configuration properties.
  - **/classes/alfresco ->** Contains the directory structure for the configuration override files, including the extension, and web-extension directories.

- Delete all files/folders from “C:\alfresco-community62ga\tomcat\webapps”
- Copy the contents from “C:\Downloads\**alfresco-content-services-community-distribution-6.2.0-ga\web-server\webapps**” into “C:\alfresco-community62ga\tomcat\webapps” folder.

  Following files will be copied:

  - **alfresco.war ->** Alfresco repository web application.
  - **share.war ->** Share interface web application.
  - **ROOT.war ->** Application for the server root, also contains additional code/setup for handling alfresco office services module (alfresco-office-services). Alfresco Office Services (AOS) allow you to access Alfresco Content Services directly from all your Microsoft Office applications. We will not be installing this module since SSL is a mandatory enablement for this module.
  - **_vti_bin.war ->** App to help and support AOS module.

<u>Configure tomcat and alfresco content services:</u>

- Open **server.xml** file which can be found under: “C:\alfresco-community62ga\tomcat\conf” directory [Full path: C:\alfresco-community62ga\tomcat\conf\server.xml] and update following:

  - Find Connector with port "8080”.
  - Add the **URIEncoding** and **maxHttpHeaderSize** attributes.

```
<Connector port="8080" protocol="HTTP/1.1"
           URIEncoding="UTF-8"
```

```
            connectionTimeout="20000"
            maxHttpHeaderSize="32768"
           redirectPort="8443" />
```

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 the Kerberos authentication protocol. We need to increase this buffer size.

Update AJP Connector config you are planning to use it. By default it is commented in tomcat server.xml, make sure you add URIEncoding attribute to the config if you uncomment it. This port is used when configuring external authentication. See here for more details: https://docs.alfresco.com/community/concepts/auth-basics.html

e.g.

```
<Connector port="8009" URIEncoding="UTF-8" protocol="AJP/1.3"
redirectPort="8443" />
```

- Open the C:\alfresco-community62ga\tomcat\conf\catalina.properties file and update following:
  - Change the value of the **shared.loader=** property to the following:

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

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

- Rename "C:\alfresco-community62ga\tomcat\shared\classes\alfresco-global.properties.sample" to "**alfresco-global.properties**"
- Open "C:\alfresco-community62ga\tomcat\shared\classes\alfresco-global.properties" and add the following configuration properties:
  - Add the "dir.root" property as path to alf_data folder.

    ```
    dir.root=C:/alfresco-community62ga/alf_data
    ```

  - Add alfresco and share host, port, context and protocol specific properties.

    ```
            alfresco.context=alfresco
            alfresco.host=${localname}
            alfresco.port=8080
            alfresco.protocol=http
    ```

```
share.context=share
share.host=${localname}
share.port=8080
share.protocol=http
```

- Add the database connection properties which alfresco will use to create a JDBC connection with db. Use the db name and credentials when we prepared the database.

```
db.driver=org.postgresql.Driver
db.username=alfresco
db.password=alfresco
db.name=alfresco
db.url=jdbc:postgresql://localhost:5432/${db.name}
db.pool.max=275
db.pool.validate.query=SELECT 1
```

- Add the server mode property, leave it default to ‘UNKNOWN’.

```
system.serverMode=UNKNOWN
```

- Add the alfresco rmi services port and host properties.

```
alfresco.rmi.services.port=50500
alfresco.rmi.services.host=0.0.0.0
```

- Add alfresco-pdf-renderer (used for pdf previews), imagemagick (used for image preview and thumbnail generation), libreoffice and jodconverter properties.

```
alfresco-pdf-renderer.root=C:/alfresco-community62ga/alfresco-pdf-renderer
alfresco-pdf-renderer.exe=${alfresco-pdf-renderer.root}/alfresco-pdf-renderer.exe

img.root=C:/alfresco-community62ga/imagemagick
img.dyn=${img.root}/lib
img.exe=${img.root}/convert.exe
img.coders=${img.root}/modules/coders
img.config=${img.root}
img.gslib=${img.root}/lib

ooo.exe=C:/alfresco-community62ga/libreoffice/program/soffice.exe
ooo.enabled=false
ooo.port=8100
jodconverter.enabled=true
jodconverter.officeHome=C:/alfresco-community62ga/libreoffice
jodconverter.portNumbers=8100
```

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

OOoJODconverter

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

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

OOoDirect

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

Note:

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

- Add property for email notification on invite, it is by default disabled.

```
notification.email.siteinvite=false
```

- Add the license location property.

```
dir.license.external=C:/alfresco-community62ga
```

- Add the activemq url property and disable messaging subsystem until you want to configure transformation services. It will be used when you setup transformation service.

```
messaging.broker.url=\"failover:(nio://localhost:61616)?timeout=3000&jms.useCompression=true\"
messaging.subsystem.autoStart=false
```

We are not setting up transformation service currently but keeping activemq setup ready for future in case you would like to start using transformation service. For more details visit:

https://hub.alfresco.com/t5/alfresco-content-services-blog/introducing-alfresco-transform-service-new-with-acs-6-1/ba-p/288174
https://hub.alfresco.com/t5/alfresco-content-services-hub/alfresco-community-edition-201910-ea-release-notes/ba-p/292243
https://github.com/Alfresco/alfresco-transform-core
https://docs.alfresco.com/transform/tasks/transform-install.html
https://docs.alfresco.com/transform/concepts/transform-local-legacy.html
https://hub.alfresco.com/t5/alfresco-content-services-blog/quick-reference-for-transformers-in-acs-community-6-2/ba-p/294824

- Disable transformation service and enable legacy transformation service:

```
transform.service.enabled=false
local.transform.service.enabled=false
legacy.transform.service.enabled=true
```

- Add following properties related to security, smart folder and JMX. All values are default ootb

```
security.anyDenyDenies=false
smart.folders.enabled=false
alfresco.jmx.connector.enabled=false
```

- Add following properties related to FTP connection (Optional)

```
 ftp.port=2121
ftp.enabled=true
ftp.server.enables=true
```

- Enable logging so that you can configure any custom loggers or change/update log levels (Optional)
    - Go to **C:\alfresco-community62ga\tomcat\shared\classes\alfresco\extension** folder and **rename** "**custom-log4j.properties.sample**" to "**custom-log4j.properties**"

- Apply amps to alfresco.war and share.war.
    - **Mandatory Amps**:
        - "***alfresco-share-services.amp***" amp is mandatory for alfresco.war

          Note: This amp installation is mandatory otherwise several functionalities including share search may not work. And you will also see following warning when you login to share: ***Alfresco is running without Share Services. See your System Administrator for more details***

    - **Optional Amps**: As part of this installation, we will be installing some optional amps which will be useful for admins/developers that can help in debugging and administration.

        - **support-tools-repo-1.1.0.0-amp.amp -\>** A module that provides utilities for developers and administrators via alfresco admin console. Download the amp from here, and copy in “C:\alfresco-community62ga\\amps” folder.
        - **support-tools-share-1.1.0.0-amp.amp -\>** A module that provides utilities for developers and administrators via share admin console. Download the amp from here, and copy in “C:\alfresco-community62ga\\amps_share” folder.

          For more details visit:
          https://github.com/OrderOfTheBee/ootbee-support-tools

        - **javascript-console-repo-0.6.amp -\>** A module that provides ability for developers and administrators to execute JavaScript code. It is repository layer dependency for share amp. Download the amp from here, and copy in “C:\alfresco-community62ga\\amps” folder.
        - **javascript-console-share-0.6.amp -\>** A module that provides ability for developers and administrators to execute JavaScript code. Download the amp from here, and copy in “C:\alfresco-community62ga\\amps_share” folder.

          For more details visit:
          https://github.com/share-extras/js-console

    - Open a command prompt and navigate to “C:\alfresco-community62ga\\bin”
    - Execute “**apply_amps.bat**” script, it will apply all the amps available in “C:\alfresco-community62ga\\amps” folder to alfresco.war and all the amps available in “C:\alfresco-community62ga\\amps_share” to share.war.

- Make sure you update the "***repository-url***" to point to the alfresco host and port, in case you are setting up alfresco and share on different servers. Default values are "***localhost***" and "***8080***". We are setting up both acs and share on same host so we will leave it default.

```
<!--
 If set, will present a WebDAV link for the current item on the Document and Folder
details pages.
  Also used to generate the "View in Alfresco Explorer" action for folders.
-->
<repository-url>http://localhost:8080/alfresco</repository-url>
```

- Lastly, make sure you update the share "***Remote***" configuration to point to the alfresco host and port, in case you are setting up alfresco and share on different servers. Default values are "***localhost***" and "***8080***". We are setting up both acs and share on same host so we will leave it default.

```
<config evaluator="string-compare" condition="Remote">
    <remote>
```

```
              <endpoint>
                <id>alfresco-noauth</id>
                <name>Alfresco - unauthenticated access</name>
                <description>Access to Alfresco Repository WebScripts that do not require
        authentication</description>
                <connector-id>alfresco</connector-id>
                <endpoint-url>http://localhost:8080/alfresco/s</endpoint-url>
                <identity>none</identity>
              </endpoint>

              <endpoint>
                <id>alfresco</id>
                <name>Alfresco - user access</name>
                <description>Access to Alfresco Repository WebScripts that require user
        authentication</description>
                <connector-id>alfresco</connector-id>
                <endpoint-url>http://localhost:8080/alfresco/s</endpoint-url>
                <identity>user</identity>
              </endpoint>

              <endpoint>
                <id>alfresco-feed</id>
                <name>Alfresco Feed</name>
                <description>Alfresco Feed - supports basic HTTP authentication via the
        EndPointProxyServlet</description>
                <connector-id>http</connector-id>
                <endpoint-url>http://localhost:8080/alfresco/s</endpoint-url>
                <basic-auth>true</basic-auth>
                <identity>user</identity>
              </endpoint>

              <endpoint>
                <id>alfresco-api</id>
                <parent-id>alfresco</parent-id>
                <name>Alfresco Public API - user access</name>
                <description>Access to Alfresco Repository Public API that require user
        authentication. This makes use of the authentication that is provided by parent
        'alfresco' endpoint.</description>
                <connector-id>alfresco</connector-id>
                <endpoint-url>http://localhost:8080/alfresco/api</endpoint-url>
                <identity>user</identity>
              </endpoint>
            </remote>
</config>
```

Configure search service:

- Unzip the “alfresco-search-services-1.4.0.zip” package which we downloaded initially.
- Copy the extracted folder “alfresco-search-services” to e.g. C:\ drive. Full path will be “C:\alfresco-search-services”
- Open “C:\alfresco-community62ga\tomcat\shared\classes\alfresco-global.properties” and add the following configuration properties:

```
            solr.host=localhost
            solr.port=8983
            solr.secureComms=none #Possible values are: none, https
            solr.base.url=/solr
            index.subsystem.name=solr6
```

  Note: We are setting up solr6 without SSL, hence using non SSL port and setting secureComms property as none.

- We are enabling the multi language search support, Its optional if you wish to enable it. By default it is disabled. Open “C:\alfresco-search-services\solrhome\conf\shared.properties” and update following:

```
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
```

- We are enabling search suggestions for some properties, it is disabled by default. This feature is optional if you wish to enable it. You can also add additional properties down the line. Open “C:\alfresco-search-services\solrhome\conf\shared.properties” and update following:

```
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
```

  More details on shared.properties can be found here: https://docs.alfresco.com/search-community/concepts/solr-shared-properties.html

- If you would setup search services on a different or remote machine, you would need to set the SOLR_SOLR_HOST, SOLR_SOLR_PORT, SOLR_SOLR_BASEURL, SOLR_ALFRESCO_HOST, SOLR_ALFRESCO_PORT and SOLR_ALFRESCO_BASEURL environment variables. This is optional but we will keep the mapping for future reference.
- We are setting up without SSL, so set this variable SOLR_ALFRESCO_SECURECOMMS to **none**. It is mandatory as default value is **https**
  Open “C:\alfresco-search-services\solr.in.cmd” file and add following at the end of the file:

```
set SOLR_SOLR_HOST=localhost
set SOLR_SOLR_PORT=8983
set SOLR_SOLR_BASEURL=/solr
set SOLR_ALFRESCO_HOST=localhost
set SOLR_ALFRESCO_PORT=8080
set SOLR_ALFRESCO_BASEURL=/alfresco

:: Since we are setting up with no SSL, this property need to be set to none. Default is https
set SOLR_ALFRESCO_SECURECOMMS=none
```

Alternatively you can set the properties in “C:\**alfresco-search-services\solrhome\templates\rerank\conf\solrcore.properties**” file as well before cores are created. If you have both alfresco and archive cores already created use the environment variable based approach.

For all the externalized search service configurations (environment variables), visit: https://docs.alfresco.com/search-community/concepts/external-properties-solr.html

- Starting and stopping solr6 (aka alfresco search service) instructions.

  Starting Solr6:

  ```
  .\solr\bin\solr.cmd start -a "-Dcreate.alfresco.defaults=alfresco,archive"
  ```

  The command line param, **-a** passes additional JVM parameters, e.g., system properties using -D. Note that, ***-Dcreate.alfresco.defaults=alfresco,archive*** command automatically creates the alfresco and archive cores. So ***you need to pass this param only on first/initial startup*** of solr in order to allow cores being created and configured.

  Once Search Services is up and running, you should see a message similar to the following:

  ```
  Waiting up to 180 seconds to see Solr running on port 8983 [\]
  Started Solr server on port 8983 (pid=443218). Happy searching!
  ```

  Stopping Solr6:

  ```
  .\solr\bin\solr stop -p 8983

  Or

  .\solr\bin\solr stop -all
  ```

- Configure solr log, The logs are stored in the C:\alfresco-search-services\logs\solr.log file, by default. This can be configured in solr.in.cmd using ***SOLR_LOGS_DIR***. This is optional setup. We are keeping default settings.

  ```
  set SOLR_LOGS_DIR=..\..\logs
  set LOG4J_CONFIG=file:!SOLR_LOGS_DIR!\log4j.properties
  ```

- Verify the SOLR Admin UI

Use the URL below to launch SOLR Admin UI and validate the status and cores:

```
http://localhost:8983/solr
http://localhost:8983/solr/#/alfresco --> To check alfresco core status
```

Now we are done with all the setup and config changes. Here is full "alfresco-global.properties", "custom-log4j.properties", "shared.properties" and "solr.in.cmd" for reference:

alfresco-global.properties:

```
################################
## Common Alfresco Properties #
################################


dir.root=C:/alfresco-community62ga/alf_data

#
# URL Generation Parameters (The ${localname} token is replaced by the local server name)
#-------------
alfresco.context=alfresco
alfresco.host=${localname}
alfresco.port=8080
alfresco.protocol=http
share.context=share
share.host=${localname}
share.port=8080
share.protocol=http

### database connection properties ###
db.driver=org.postgresql.Driver
db.username=alfresco
db.password=alfresco
db.name=alfresco
db.url=jdbc:postgresql://localhost:5432/${db.name}
# Note: your database must also be able to accept at least this many connections.  Please see your
database documentation for instructions on how to configure this.
db.pool.max=275
db.pool.validate.query=SELECT 1

# The server mode. Set value here
# UNKNOWN | TEST | BACKUP | PRODUCTION
system.serverMode=UNKNOWN

### RMI registry port for JMX ###
alfresco.rmi.services.port=50500

# Default value of alfresco.rmi.services.host is 0.0.0.0 which means 'listen on all adapters'.
# This allows connections to JMX both remotely and locally.
alfresco.rmi.services.host=0.0.0.0

#
#
# Assign individual ports for each service for best performance
# or run several services on the same port. You can even run everything on 50500 if needed.
# Select 0 to use a random unused port.
#monitor.rmi.service.port=50508


### External executable locations ###

#PDF Renderer config, used for previewing pdf docs.
alfresco-pdf-renderer.root=C:/alfresco-community62ga/alfresco-pdf-renderer
alfresco-pdf-renderer.exe=${alfresco-pdf-renderer.root}/alfresco-pdf-renderer.exe

#Imagemagick config
img.root=C:/alfresco-community62ga/imagemagick
img.dyn=${img.root}/lib
img.exe=${img.root}/convert.exe
img.coders=${img.root}/modules/coders
img.config=${img.root}
img.gslib=${img.root}/lib

####LibreOffice and jodconverter configurations
#You can transform a document from one format to another using the OOoJodconverter subsystem.
This feature requires you to install LibreOffice.
#The Jodconverter integration is a library that improves the stability and performance of LibreOffice
in Alfresco Content Services.
#The Jodconverter runs on the same machine as the Alfresco Content Services 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
#The subsystem for direct OpenOffice integration, in which the Alfresco server manages OpenOffice
directly. To enable or disable this subsystem, use the following property:
#ooo.enabled=false
#Note:
#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

ooo.enabled=false
jodconverter.enabled=true
ooo.exe=C:/alfresco-community62ga/libreoffice/program/soffice.exe
ooo.enabled=false
ooo.port=8100
jodconverter.enabled=true
jodconverter.officeHome=C:/alfresco-community62ga/libreoffice
jodconverter.portNumbers=8100


### E-mail site invitation setting ###
notification.email.siteinvite=false

### License location ###
dir.license.external=C:/alfresco-community62ga

#Confiure ActiveMQ URL.
messaging.broker.url=\"failover:(nio://activemq:61616)?timeout=3000&jms.useCompression=true\"
messaging.subsystem.autoStart=false

#Transform service configurations
transform.service.enabled=false
local.transform.service.enabled=false
legacy.transform.service.enabled=true

### Allow extended ResultSet processing
security.anyDenyDenies=false

### Smart Folders Config Properties ###
smart.folders.enabled=false

### Remote JMX (Default: disabled) ###
alfresco.jmx.connector.enabled=false

################ Solr Search service configurations ###############

#
# Index Recovery Mode
#-------------
#index.recovery.mode=AUTO

# Set this property unless you have explicitly chosen to expose some repository APIs without
authentication
solr.host=localhost
solr.port=8983
#none, https
solr.secureComms=none
solr.base.url=/solr
index.subsystem.name=solr6

### FTP Server Configuration ###
ftp.port=2121
ftp.enabled=true
ftp.server.enables=true
```

custom-log4j.properties:

```
#Tranformer specific logs
log4j.logger.org.alfresco.repo.content.transform.TransformerDebug=info
log4j.logger.org.alfresco.util.exec.RuntimeExecBootstrapBean=info
log4j.logger.org.alfresco.util.exec.RuntimeExec=info

#ScriptLogger
log4j.logger.org.alfresco.repo.jscript.ScriptLogger=debug

#Log for email executer
log4j.logger.org.alfresco.repo.action.executer.MailActionExecuter=info

#Transaction specific logs
log4j.logger.org.alfresco.repo.transaction.RetryingTransactionHelper=info

#Solr specific logs
log4j.logger.org.alfresco.solr.query.AbstractQParser=debug
log4j.logger.org.alfresco.repo.search.impl.solr.SolrQueryHTTPClient=debug

#Metadata extractor
log4j.logger.org.alfresco.repo.content.metadata.AbstractMappingMetadataExtracter=info
log4j.logger.org.alfresco.repo.content.metadata.MetadataExtracterRegistry=info

#Thumbnail logs
log4j.logger.org.alfresco.repo.thumbnail=info

# FTP server debugging
log4j.logger.org.alfresco.ftp.protocol=info
log4j.logger.org.alfresco.ftp.server=info
```

shared.properties:

```
# Shared Properties file

#Host details an external client would use to connect to Solr
solr.host=localhost
#If not set then solr.port will be the jetty.port
#solr.port=8983
solr.baseurl=/solr

# 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
alfresco.identifier.property.4={http://www.alfresco.org/model/content/1.0}lockOwner

# 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

# 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
alfresco.cross.locale.property.1={http://www.alfresco.org/model/content/1.0}lockOwner

# 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

alfresco.model.tracker.cron=0/10 * * * * ? *
```

solr.in.cmd:

```
@echo off

REM Increase Java Min/Max Heap as needed to support your indexing / query needs
set SOLR_JAVA_MEM=-Xms1g -Xmx1g

REM Alfresco configuration. This file is automatically included by solr. You can define your custom
settings here
set SOLR_OPTS=%SOLR_OPTS% -Dsolr.jetty.request.header.size=1000000 -
Dsolr.jetty.threads.stop.timeout=300000 -Ddisable.configEdit=true

REM Location where Solr should write logs to. Absolute or relative to solr start dir
```

```
set SOLR_LOGS_DIR=..\..\logs
set LOG4J_CONFIG=file:!SOLR_LOGS_DIR!\log4j.properties


set SOLR_SOLR_HOST=localhost
set SOLR_SOLR_PORT=8983
set SOLR_SOLR_BASEURL=/solr
set SOLR_ALFRESCO_HOST=localhost
set SOLR_ALFRESCO_PORT=8080
set SOLR_ALFRESCO_BASEURL=/alfresco

:: Since we are setting up with no SSL, this property need to be set to none. Default is https
set SOLR_ALFRESCO_SECURECOMMS=none
```

<u>Start and test:</u>

There are three services that we need to start one by one in order:

1- Start DB
- a. To start DB, you can do either of the following:
    - i. Use windows services, and *start “postgresql-x64-11” service*.
    - *ii.* Or Open command prompt and execute this command: *C:\PostgreSQL\11\bin\pg_ctl.exe start -D “C:\PostgreSQL\11\data"*

2- Start Alfresco
- a. To start alfresco, you can do either of the following:
    - i. Open command prompt and navigate to “C:\alfresco-community62ga\tomcat\bin” and execute following command: *catalina.bat start*
    - ii. Or navigate to “C:\alfresco-community62ga\tomcat\bin” folder and execute *“startup.bat”*

3- Start Solr6
- a. To start SOLR6, use the following command by navigating to “C:\alfresco-search-services\solr\bin” folder via command prompt:
    - *i.* Initial start only: *C:\alfresco-search-services\solr\bin\solr.cmd start -a "-Dcreate.alfresco.defaults=alfresco,archive"*
    - *ii.* Consecutive start command: *C:\alfresco-search-services\solr\bin\solr.cmd start*

Note: See the search service configuration section for more details.

I have created this bat file “**start.bat**”, you can use this to start all services instead of doing one by one.

start.bat:

```
@echo off

ECHO ################ Starting ACS, DB and Solr Services ##############
ECHO.


SET ALF_INSTALL_PATH=%1
SET SOLR_INSTALL_PATH=%2
SET POSTGRES_INSTALL_PATH=%3

:init
        IF "%~1" == "" (
          SET ALF_INSTALL_PATH=C:\alfresco-community62ga
        )

        IF "%~2" == "" (
                SET SOLR_INSTALL_PATH=C:\alfresco-search-services
        )

        IF "%~3" == "" (
                SET POSTGRES_INSTALL_PATH=C:\PostgreSQL\11
        )

        goto startDB

:startDB
        echo Starting DB...
        :: Using the windows service to start the db.
        :: net start postgresql-x64-11
        REM You can also use this command, if there is any issue with permission elevation on
windows
        %POSTGRES_INSTALL_PATH%\bin\pg_ctl.exe restart -D "%POSTGRES_INSTALL_PATH%\data"
        if errorlevel 1 (goto end) else (goto startACS)

:startACS
        echo.
        echo Starting ACS...
        SET CATALINA_HOME=%ALF_INSTALL_PATH%\tomcat
        start /MIN /WAIT cmd /c %ALF_INSTALL_PATH%\tomcat\bin\catalina.bat start
        if errorlevel 1 (goto end) else (goto startSolr)

:startSolr
        echo.
        set "initial=false"

        :: check if cores exists
        echo.
        set "initial=false"
        CD %SOLR_INSTALL_PATH%\solrhome
        :: check if cores exists

        set Exts=alfresco archive
        for %%A in (%Exts%) do (
          echo Checking core: %%A
          if not exist %%A\NUL (
                echo %%A doesn't exist
            set "initial=true"
          ) else (
                echo %%A already exist
                set "initial=false"
          )
        )

        CD %ALF_INSTALL_PATH%
        if "%initial%" == "true" (
                GOTO startSolrInitial
        ) else (
                GOTO startSolrConsecutive
        )

:startSolrInitial
        echo.
        echo Starting SOLR for the first time, alfresco and archive cores will be created...
        start /MIN /WAIT cmd /c %SOLR_INSTALL_PATH%\solr\bin\solr.cmd start -a "-
Dcreate.alfresco.defaults=alfresco,archive"
        if errorlevel 1 (goto end)

:startSolrConsecutive
        echo.
        echo Starting SOLR...
        start /MIN /WAIT cmd /c %SOLR_INSTALL_PATH%\solr\bin\solr.cmd start
        if errorlevel 1 (goto end)


:end
        echo.
  echo Exiting..
        timeout 10
```

Stopping services:

There are three services that we need to stop one by one in order:

1- Stop Alfresco
   a. To stop alfresco, you can do either of the following:
      i. Open command prompt and navigate to “C:\alfresco-community62ga\tomcat\bin” and execute following command: *catalina.bat stop*
      ii. Or navigate to “C:\alfresco-community62ga\tomcat\bin” folder and execute *“shutdown.bat”*

2- Stop Solr6
   a. To stop SOLR6, use the following command by navigating to “C:\alfresco-search-services\solr\bin” folder via command prompt:
      C:\alfresco-search-services\solr\bin\solr.cmd stop -all
      Or
      C:\alfresco-search-services\solr\bin\solr.cmd stop -p 8983

   Note: See the search service configuration section for more details.

3- Stop DB
   a. To stop DB, you can do either of the following:
      i. Use windows services, and *stop “postgresql-x64-11” service*.
      ii. Or Open command prompt and execute this command: *C:\PostgreSQL\11\bin\pg_ctl.exe stop -D "C:\PostgreSQL\11\data"*

I have created this bat file “**stop.bat**”, you can use this to stop all services instead of doing one by one.

stop.bat:

```
@echo off

ECHO ################ Stopping ACS, DB and Solr Services ##############
ECHO.

SET ALF_INSTALL_PATH=%1
SET SOLR_INSTALL_PATH=%2
SET POSTGRES_INSTALL_PATH=%3

:init
        IF "%~1" == "" (
          SET ALF_INSTALL_PATH=C:\alfresco-community62ga
        )

        IF "%~2" == "" (
                SET SOLR_INSTALL_PATH=C:\alfresco-search-services
        )
        IF "%~3" == "" (
                SET POSTGRES_INSTALL_PATH=C:\PostgreSQL\11
        )

        goto stopACS

:stopACS
        echo.
        echo Stopping ACS from %ALF_INSTALL_PATH% ...
        SET CATALINA_HOME=%ALF_INSTALL_PATH%\tomcat
        start /MIN /WAIT cmd /c %ALF_INSTALL_PATH%\tomcat\bin\catalina.bat stop
        if errorlevel 1 (goto end) else (goto stopDB)

:stopDB
        echo.
        echo Stopping DB from %POSTGRES_INSTALL_PATH% ...
        :: Using the windows service to stop the db.
        :: net stop postgresql-x64-11
        REM You can also use this command, if there is any issue with permission elevation on windows
        %POSTGRES_INSTALL_PATH%\bin\pg_ctl.exe stop -D "%POSTGRES_INSTALL_PATH%\data"
        if errorlevel 1 (goto end) else (goto stopSolr)

:stopSolr
        echo.
        echo Stopping SOLR from %SOLR_INSTALL_PATH% ...
        %SOLR_INSTALL_PATH%\solr\bin\solr.cmd stop -all
        if errorlevel 1 (goto end)

:end
        echo.
  echo Exiting..
        timeout 10
```

References:

https://docs.alfresco.com