2.5.x Installation

THIS DOCUMENT IS INPROGRESS, NOT COMPLETE

Table of Contents

Introduction

The purpose of this document is to guide production level installations of the Crafter WEM Platform
This document is intended to guide administrators in the installation of production level installations of the Crafter CMS Platform.

If you are doing an evaluation, a demo, or if you are a developer, you should consider using our installers which are designed to get you up and running within minutes. You can find the latest installers here: Crafter Downloads

Relevant Crafter WEM and Alfresco Version
This document applies to version Crafter CMS 2.5.x.
Please refer to the Supported Platforms Matrix document for detailed information on the combinations of operating systems, databases and application servers that have been tested and certified by Alfresco.

Crafter CMS Deployment Architecture

High level Server Topology and Anatomy

CMS "Authoring" Node

Application	File name	Purpose	Interaction
Studio Application	studio.war	User interface for CMS, includes plug-ins for Crafter Studio	Authors interact with cms primarily through share
Crafter Engine	ROOT.war	Provides in-context preview and authoring functionality	Authors are directed to this application via Alfresco Share for in-context preview, editing and so on.
Crafter Search	crafter-search.war	provides a light weight wrapper and service layer around a search index (typically Apache Solr	Users never interact with this application directly. Crafter Engine interacts with this application via ReSTful service.
Crafter Search Index	solr-crafter.war	Provide the website search and dynamic query capability (site not cms)	Users never interact with this application directly. Crafter Search interacts with this application via ReSTful services

Delivery Node

Application	File name	Purpose	Interaction
Crafter Engine	ROOT.war	Provides site delivery functionality	website visitors interact with this application directly

Crafter Search	crafter-search.war	provides a light weight wrapper and service layer around a search index (typically Apache Solr	Users never interact with this application directly. Crafter Engine interacts with this application via ReSTful service.
Crafter Search Index	solr-crafter.war	Provide the website search and dynamic query capability (site not cms)	Users never interact with this application directly. Crafter Search interacts with this application via ReSTful services
Crafter Deployer	cstudio-publishing-receiver.zip	Stand-alone application to receive content deployed from Alfresco repositories	Users never interact with this application directly. Alfresco pushes content to receiver via ReSTful services

Basic Resources

Binaries

- http://wiki.craftersoftware.com/display/CRAFTER/C2.5+Setup+Guide

The Bundles

There is a separate bundle for the Authoring/CMS and the Delivery node. Each bundle contains an Apache Tomcat instance that comes prepared with the basic folders and configuration files, but without the WAR files.
http://wiki.craftersoftware.com/display/CRAFTER/C2.5+Setup+Guide

Instructions for building from source can be found here:Source Code

-
    - Source code & Build Instructions

> ⚠ You do NOT need to build from source. These links are provided for completeness and as a convenience to those interested in source level assets.

Authoring / CMS Installation

OS Application level User / Run As

1. Create a user that the application will run under such as 'crafter'
2. Make sure 'crafter' is the owner of all files in this setup

Disk Space / Locations

Content Storage

Our recommendation is to create a mount specifically for managing content and another for managing indexes. Ideally the content store, the indexes and the database are all on different disks (and in some cases different machines.) Remember, the CMS is all about i/o. Anything you can do to make disk access faster the better your performance will be. A Clustered versus a non-clustered setup also plays a role in what storage you use. Please read the clustering guide from Alfresco before making your determinations.
You will need room for:

- Content Store
    - Consider the volume of content you will be hosting, the growth and the number of versions which are likely to occur.
- Search Indexes
    - Indexes are usually the size of your XML based assets plus room to grow.
- Database
    - Databases size includes extracted metadata. Consider the number of assets, versions for each asset and the number of stock and custom properties on each asset.

> ⚠ For the purpose of this document we'll put everything on a single environment. This is just for simplicity. We could create a more sophisticated topology but it would represent only one of many possibilities. The key here is that we will demonstrate where you can set up the various paths in the configuration. The work of mounting etc will depend on your infrastructure and will not be represented here.

Application Server

OS Application level User / Run As

1. Create a user that the application will run under, such as '`crafter`'

Install the Bundle

Linux

1. Create an install location: `/opt/alfresco`
2. Grant the `crafter` account ownership
3. Unzip the cms node bundle inside this install location (see The Bundle)

Java

Ensure that the machine has the appropriate version of Java installed.

1. Alfresco is only certified for 1.7.x at this time
2. Note that there is a 2 GB memory limit for JVMs running on 32 bit operating systems. 64 bit JVMs will allow you to utilize much more memory.

```
java -version
java version "1.7.0_51"
Java(TM) SE Runtime Environment (build 1.7.0_51-b13)
Java HotSpot(TM) 64-Bit Server VM (build 24.51-b03, mixed mode)
```

Configure JAVA HOME

1. Determine where java is installed:

    which java

    ```
    which java
    /usr/bin/java
    ```

2. Modify (if required) the location of `JAVA_HOME` for Apache Tomcat

/opt/alfresco/apache-tomcat/bin/setenv.sh

```
export JAVA_HOME=/usr
```

3. Modify (if required) the PATH to Java for Apache Tomcat

/opt/alfresco/scripts/setenv.sh

```
export PATH="$PATH:/usr/bin"
```

Configuring JVM Memory

In this section we can tell you where but not exactly how – as this is a matter of tuning. That said, you must have enough perm gen allocated for Alfresco to start up properly without running out of memory.

"/opt/alfresco/apache-tomcat/bin/setenv.sh"

```
JAVA_OPTS="$JAVA_OPTS -server -Xss1024K -Xms1G -Xmx8G -XX:MaxPermSize=512M
-XX:NewSize=512m"
```

Configure Apache Tomcat Ports

The default ports in the CMS bundle are as follows:

Purpose	Port
Server	8005
AJP/1.3 Connector	8009
HTTP/1.1 Connector	8080
SSL HTTP/1.1 Connector	8443

⚠ **Which ports to use**
These should be kept unless a change is needed
For the sake of users the server is usually made available on port 80 and 443 which means no port is shown in the browser. To accomplish this take one of the two following approaches:

- Use Apache HTTPD in front of Alfresco. Apache HTTPD (or other web server) will run on port 80 and forward to Tomcat on 8080 via modProxy or the like. This approach is preferred as it is more commonly understood and offers the other performance enhancements that apache can bring to the environment.
- Use port forwarding (IPTABLES)

Edit the Apache Tomcat server configuration to change the ports if needed. We recommend using a pattern, for example by increasing all port numbers by a factor 1000:

Default Value	New Value
8005	9005

8080	9080
8443	9443
8009	9009

Search and replace these values in VI:

vi /opt/alfresco/apache-tomcat/conf/server.xml

```
:%s/8005/9005/g
:%s/8080/9080/g
:%s/8443/9443/g
:%s/8009/9009/g
```

Configuring the location of your repository data and keystore for SSL

Originally located in the bundle as `/opt/alfresco/alf_data`, move this data directory to an appropriate mount point as discussed above. After that, configure the `keyStoreFile` and `truststoreFile` paths in `server.xml` accordingly.

/opt/alfresco/apache-tomcat/conf/server.xml

```
<Connector
port="8443"
protocol="org.apache.coyote.http11.Http11Protocol"
SSLEnabled="true"
maxThreads="150"
scheme="https"
keystoreFile="/opt/alfresco/alf_data/keystore/ssl.keystore"
keystorePass="kT9X6oe68t"
keystoreType="JCEKS"
secure="true"
connectionTimeout="240000"
truststoreFile="/opt/alfresco/alf_data/keystore/ssl.truststore"
truststorePass="kT9X6oe68t"
truststoreType="JCEKS"
clientAuth="false"
sslProtocol="TLS"
allowUnsafeLegacyRenegotiation="true" />
```

Automatic Start-up of Authoring Application Server

Linux

Warning

This startup script doesn't work at present. Ensure that the executable flag is set for the following scripts:

```
chmod +x /opt/alfresco/ctlscript.sh
chmod +x /opt/alfresco/scripts/setenv.sh
chmod +x /opt/alfresco/apache-tomcat/bin/setenv.sh
```

/etc/init.d/alfresco

```
#!/bin/sh
#
# Alfresco startup script
#
export ALFRESCO_HOME_PATH='/opt/alfresco'
export JAVA_OPTS='-Xms256m -Xmx1024m -XX:MaxPermSize=1024m -server
-Dfile.encoding=UTF8'
export JAVA_OPTS="${JAVA_OPTS} -Dalfresco.home=${ALFRESCO_HOME_PATH}
-Dcom.sun.management.jmxremote"
# Following only needed for Sun JVMs before to 1.5 update 8

export JAVA_OPTS="${JAVA_OPTS}
-XX:CompileCommand=exclude,org/apache/lucene/index/IndexReader\$1,doBody
-XX:CompileCommand=exclude,org/alfresco/repo/search/impl/lucene/index/Inde
xnoformat\$Merger,mergeIndexes
-XX:CompileCommand=exclude,org/alfresco/repo/search/impl/lucene/index/Inde
xnoformat\$Merger,mergeDeletions"

#Uncomment to connect with a debugger
#export JAVA_OPTS="$JAVA_OPTS -Xdebug -Xnoagent -Djava.compiler=NONE
-Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=5605"

tomcatPID=`ps -ef|grep "${ALFRESCO_HOME_PATH}/apache-tomcat"|grep java| awk
'{print $2}'`;
cur_dir=`pwd`;

case "$1" in
start)
cd ${ALFRESCO_HOME_PATH};
sh ctlscript.sh start
cd $cur_dir;
;;
stop)
cd ${ALFRESCO_HOME_PATH};
sh ./ctlscript.sh stop;
if ! test -z $tomcatPID;
then
echo "Killing Tomcat process $tomcatPID";
kill -9 $tomcatPID;
fi
cd $cur_dir;
;;
restart)
$0 stop;
sleep 1;
$0 start
;;
*)
echo "Usage: $0 {start|stop|restart}";
exit 1;
```

```
;;
esac
```

```
exit 0
```

Alfresco / Authoring Database Setup

Crafter and Alfresco share the same database schema. Crafter tables do not join or interfere with Alfresco tables in anyway and pose no threat / risk to an Alfresco upgrade. The schema is shared only to allow sharing the same connection pool.

Schema

You may use whatever schema name you wish.

- The most common and recommended schema name is `alfresco`.
    - In this example we are using `crafter` but will soon be updated to `alfresco`.

Specific Database Setups

MySQL

1. Create the Alfresco database. The schema name here is `crafter`, if you change this you will need to modify `db.username` in `alfresco-global.properties` as well.

    ```
    CREATE DATABASE crafter DEFAULT CHARACTER SET utf8;
    ```

2. create a user. username/password here are crafter, if you change this you will need to modify alfresco-global.properties as well.

    ```
    CREATE USER 'crafter' IDENTIFIED BY 'crafter';
    CREATE USER 'crafter'@'localhost' IDENTIFIED BY 'crafter';
    ```

3. grant rights to the user. Start with grant all as the system will need to create your database for you. You may later choose to remove certain privileges.

    ```
    GRANT ALL PRIVILEGES ON crafter.* TO 'crafter'@'%';
    GRANT ALL PRIVILEGES ON crafter.* TO 'crafter'@'localhost';
    ```

4. set user password

    ```
    UPDATE mysql.user SET Password=PASSWORD('crafter')
    WHERE User='crafter';
    FLUSH PRIVILEGES;
    ```

Oracle

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

2. Create a user (schema). Username/password here are `crafter`, if you change this you will need to modify `db.username` in `alfresco-global.properties` as well.

```
-- USER SQL
CREATE USER crafter IDENTIFIED BY crafter ;

-- ROLES
GRANT "OLAP_XS_ADMIN" TO crafter ;
GRANT "WFS_USR_ROLE" TO crafter ;
GRANT "DELETE_CATALOG_ROLE" TO crafter ;
GRANT "TT_CACHE_ADMIN_ROLE" TO crafter ;
GRANT "HS_ADMIN_SELECT_ROLE" TO crafter ;
GRANT "CWM_USER" TO crafter ;
GRANT "SPATIAL_WFS_ADMIN" TO crafter ;
GRANT "OLAP_DBA" TO crafter ;
GRANT "OWB$CLIENT" TO crafter ;
GRANT "XFILES_ADMINISTRATOR" TO crafter ;
GRANT "RESOURCE" TO crafter ;
GRANT "APEX_ADMINISTRATOR_ROLE" TO crafter ;
GRANT "OWB_DESIGNCENTER_VIEW" TO crafter ;
GRANT "CTXAPP" TO crafter ;
GRANT "SPATIAL_CSW_ADMIN" TO crafter ;
GRANT "GATHER_SYSTEM_STATISTICS" TO crafter ;
GRANT "AUTHENTICATEDUSER" TO crafter ;
GRANT "CONNECT" TO crafter ;
GRANT "HS_ADMIN_EXECUTE_ROLE" TO crafter ;
GRANT "LOGSTDBY_ADMINISTRATOR" TO crafter ;
GRANT "JAVADEBUGPRIV" TO crafter ;
GRANT "XDB_WEBSERVICES_WITH_PUBLIC" TO crafter ;
GRANT "XDBADMIN" TO crafter ;
GRANT "XDB_WEBSERVICES_OVER_HTTP" TO crafter ;
GRANT "EXP_FULL_DATABASE" TO crafter ;
GRANT "CSW_USR_ROLE" TO crafter ;
GRANT "XFILES_USER" TO crafter ;
GRANT "OLAPI_TRACE_USER" TO crafter ;
GRANT "JAVAIDPRIV" TO crafter ;
GRANT "DBFS_ROLE" TO crafter ;
GRANT "ADM_PARALLEL_EXECUTE_TASK" TO crafter ;
GRANT "AQ_ADMINISTRATOR_ROLE" TO crafter ;
GRANT "JAVA_DEPLOY" TO crafter ;
GRANT "OEM_MONITOR" TO crafter ;
GRANT "XDB_WEBSERVICES" TO crafter ;
GRANT "JAVAUSERPRIV" TO crafter ;
GRANT "MGMT_USER" TO crafter ;
GRANT "OWB_USER" TO crafter ;
GRANT "JAVA_ADMIN" TO crafter ;
GRANT "JMXSERVER" TO crafter ;
GRANT "EXECUTE_CATALOG_ROLE" TO crafter ;
GRANT "SCHEDULER_ADMIN" TO crafter ;
GRANT "DATAPUMP_IMP_FULL_DATABASE" TO crafter ;
GRANT "WM_ADMIN_ROLE" TO crafter ;
GRANT "ORDADMIN" TO crafter ;
GRANT "AQ_USER_ROLE" TO crafter ;
GRANT "DATAPUMP_EXP_FULL_DATABASE" TO crafter ;
GRANT "SELECT_CATALOG_ROLE" TO crafter ;
GRANT "RECOVERY_CATALOG_OWNER" TO crafter ;
GRANT "OLAP_USER" TO crafter ;
GRANT "DBA" TO crafter ;
GRANT "JAVASYSPRIV" TO crafter ;
GRANT "XDB_SET_INVOKER" TO crafter ;
GRANT "IMP_FULL_DATABASE" TO crafter ;
GRANT "HS_ADMIN_ROLE" TO crafter ;
GRANT "EJBCLIENT" TO crafter ;
GRANT "OEM_ADVISOR" TO crafter ;

-- SYSTEM PRIVILEGES
GRANT ALTER TABLESPACE TO crafter ;
GRANT DROP ANY TRIGGER TO crafter ;
GRANT CREATE USER TO crafter ;
GRANT CREATE ANY OUTLINE TO crafter ;
GRANT FLASHBACK ANY TABLE TO crafter ;
GRANT ALTER ANY SEQUENCE TO crafter ;
GRANT ALTER ANY LIBRARY TO crafter ;
GRANT ADMINISTER SQL MANAGEMENT OBJECT TO crafter ;
GRANT CREATE MINING MODEL TO crafter ;
GRANT UPDATE ANY TABLE TO crafter ;
GRANT UPDATE ANY CUBE TO crafter ;
GRANT CREATE TRIGGER TO crafter ;
GRANT DROP ANY EVALUATION CONTEXT TO crafter ;
GRANT DROP PROFILE TO crafter ;
GRANT CREATE TABLESPACE TO crafter ;
GRANT DEBUG CONNECT SESSION TO crafter ;
GRANT DROP ANY DIRECTORY TO crafter ;
GRANT CREATE ASSEMBLY TO crafter ;
GRANT SELECT ANY CUBE TO crafter ;
GRANT CREATE SEQUENCE TO crafter ;
GRANT ON COMMIT REFRESH TO crafter ;
GRANT SELECT ANY SEQUENCE TO crafter ;
GRANT CREATE ANY SQL PROFILE TO crafter ;
GRANT DROP ANY SQL PROFILE TO crafter ;
GRANT ADMINISTER ANY SQL TUNING SET TO crafter ;
GRANT ADVISOR TO crafter ;
GRANT ALTER ANY MINING MODEL TO crafter ;
GRANT EXECUTE ANY OPERATOR TO crafter ;
GRANT ALTER PROFILE TO crafter ;
GRANT EXECUTE ANY TYPE TO crafter ;
GRANT CREATE ANY DIRECTORY TO crafter ;
GRANT CREATE TABLE TO crafter ;
GRANT CREATE ANY INDEX TO crafter ;
GRANT ADMINISTER RESOURCE MANAGER TO crafter ;
GRANT BECOME USER TO crafter ;
GRANT MANAGE TABLESPACE TO crafter ;
GRANT DROP ANY MINING MODEL TO crafter ;
GRANT EXECUTE ASSEMBLY TO crafter ;
GRANT SELECT ANY TABLE TO crafter ;
GRANT DROP ROLLBACK SEGMENT TO crafter ;
GRANT CREATE OPERATOR TO crafter ;
GRANT ALTER ANY CUBE TO crafter ;
GRANT ALTER PUBLIC DATABASE LINK TO crafter ;
GRANT CREATE ANY PROCEDURE TO crafter ;
GRANT CREATE ANY CUBE TO crafter ;
GRANT DROP ANY INDEXTYPE TO crafter ;
GRANT SELECT ANY MINING MODEL TO crafter ;
GRANT EXECUTE ANY CLASS TO crafter ;
GRANT CREATE ANY MATERIALIZED VIEW TO crafter ;
GRANT SELECT ANY TRANSACTION TO crafter ;
GRANT ANALYZE ANY DICTIONARY TO crafter ;
GRANT CREATE EXTERNAL JOB TO crafter ;
GRANT INSERT ANY TABLE TO crafter ;
GRANT CREATE LIBRARY TO crafter ;
GRANT GRANT ANY OBJECT PRIVILEGE TO crafter ;
GRANT CREATE JOB TO crafter ;
GRANT CREATE ANY OPERATOR TO crafter ;
GRANT ALTER ANY RULE TO crafter ;
GRANT CREATE ANY LIBRARY TO crafter ;
GRANT CREATE ANY SEQUENCE TO crafter ;
GRANT DROP PUBLIC SYNONYM TO crafter ;
GRANT CREATE CLUSTER TO crafter ;
GRANT FORCE ANY TRANSACTION TO crafter ;
GRANT UPDATE ANY CUBE DIMENSION TO crafter ;
GRANT CREATE EVALUATION CONTEXT TO crafter ;
GRANT CREATE ANY CUBE BUILD PROCESS TO crafter ;
GRANT DROP ANY OPERATOR TO crafter ;
GRANT DROP USER TO crafter ;
GRANT EXECUTE ANY INDEXTYPE TO crafter ;
GRANT ALTER ANY EDITION TO crafter ;
GRANT LOCK ANY TABLE TO crafter ;
GRANT DROP ANY TYPE TO crafter ;
GRANT CHANGE NOTIFICATION TO crafter ;
GRANT CREATE ANY DIMENSION TO crafter ;
GRANT DROP ANY DIMENSION TO crafter ;
GRANT READ ANY FILE GROUP TO crafter ;
GRANT CREATE ANY RULE TO crafter ;
GRANT ALTER ANY ASSEMBLY TO crafter ;
GRANT EXEMPT IDENTITY POLICY TO crafter ;
GRANT ALTER ROLLBACK SEGMENT TO crafter ;
GRANT CREATE RULE TO crafter ;
GRANT CREATE ANY VIEW TO crafter ;
GRANT SYSOPER TO crafter ;
GRANT CREATE PROCEDURE TO crafter ;
GRANT INSERT ANY MEASURE FOLDER TO crafter ;
GRANT SYSDBA TO crafter ;
GRANT ANALYZE ANY TO crafter ;
GRANT ALTER ANY TYPE TO crafter ;
GRANT DROP ANY EDITION TO crafter ;
GRANT CREATE ANY TRIGGER TO crafter ;
GRANT MANAGE ANY FILE GROUP TO crafter ;
GRANT DROP ANY RULE TO crafter ;
GRANT CREATE DIMENSION TO crafter ;
GRANT CREATE ROLLBACK SEGMENT TO crafter ;
GRANT FLASHBACK ARCHIVE ADMINISTER TO crafter ;
GRANT ALTER ANY RULE SET TO crafter ;
GRANT DROP ANY SEQUENCE TO crafter ;
GRANT DROP ANY TABLE TO crafter ;
GRANT CREATE CUBE DIMENSION TO crafter ;
GRANT EXECUTE ANY RULE TO crafter ;
GRANT DROP ANY LIBRARY TO crafter ;
GRANT EXECUTE ANY PROCEDURE TO crafter ;
GRANT DROP ANY VIEW TO crafter ;
GRANT DROP ANY CONTEXT TO crafter ;
GRANT FORCE TRANSACTION TO crafter ;
GRANT CREATE ANY JOB TO crafter ;
GRANT DROP ANY ROLE TO crafter ;
GRANT DELETE ANY CUBE DIMENSION TO crafter ;
GRANT DROP ANY CLUSTER TO crafter ;
GRANT UPDATE ANY CUBE BUILD PROCESS TO crafter ;
GRANT CREATE ANY INDEXTYPE TO crafter ;
GRANT ADMINISTER SQL TUNING SET TO crafter ;
GRANT EXECUTE ANY PROGRAM TO crafter ;
GRANT DROP ANY ASSEMBLY TO crafter ;
GRANT ALTER DATABASE LINK TO crafter ;
GRANT GRANT ANY PRIVILEGE TO crafter ;
GRANT ALTER ANY PROCEDURE TO crafter ;
GRANT MERGE ANY VIEW TO crafter ;
GRANT CREATE ANY EVALUATION CONTEXT TO crafter ;
GRANT ALTER ANY OPERATOR TO crafter ;
GRANT ALTER ANY CUBE DIMENSION TO crafter ;
GRANT COMMENT ANY MINING MODEL TO crafter ;
GRANT ALTER ANY ROLE TO crafter ;
GRANT EXECUTE ANY ASSEMBLY TO crafter ;
GRANT CREATE CUBE BUILD PROCESS TO crafter ;
GRANT EXECUTE ANY RULE SET TO crafter ;
GRANT ALTER ANY TRIGGER TO crafter ;
GRANT UNDER ANY TABLE TO crafter ;
GRANT BACKUP ANY TABLE TO crafter ;
GRANT CREATE SYNONYM TO crafter ;
GRANT DROP ANY CUBE BUILD PROCESS TO crafter ;
GRANT DROP ANY CUBE TO crafter ;
GRANT ALTER DATABASE TO crafter ;
GRANT ALTER ANY TABLE TO crafter ;
GRANT CREATE VIEW TO crafter ;
GRANT EXECUTE ANY LIBRARY TO crafter ;
GRANT CREATE RULE SET TO crafter ;
GRANT EXEMPT ACCESS POLICY TO crafter ;
GRANT CREATE ANY CLUSTER TO crafter ;
GRANT DROP ANY INDEX TO crafter ;
GRANT CREATE TYPE TO crafter ;
GRANT EXECUTE ANY EVALUATION CONTEXT TO crafter ;
GRANT ALTER RESOURCE COST TO crafter ;
GRANT ALTER ANY CLUSTER TO crafter ;
GRANT CREATE PUBLIC SYNONYM TO crafter ;
GRANT ALTER ANY INDEX TO crafter ;
GRANT CREATE ANY MINING MODEL TO crafter ;
GRANT GLOBAL QUERY REWRITE TO crafter ;
GRANT CREATE ANY RULE SET TO crafter ;
GRANT CREATE MEASURE FOLDER TO crafter ;
GRANT DROP ANY CUBE DIMENSION TO crafter ;
GRANT CREATE ROLE TO crafter ;
GRANT RESTRICTED SESSION TO crafter ;
GRANT DROP ANY PROCEDURE TO crafter ;
GRANT ALTER USER TO crafter ;
GRANT CREATE ANY CONTEXT TO crafter ;
GRANT CREATE ANY SYNONYM TO crafter ;
GRANT CREATE ANY CUBE DIMENSION TO crafter ;
GRANT ALTER ANY OUTLINE TO crafter ;
GRANT ENQUEUE ANY QUEUE TO crafter ;
GRANT CREATE ANY TABLE TO crafter ;
GRANT SELECT ANY CUBE DIMENSION TO crafter ;
GRANT ALTER ANY EVALUATION CONTEXT TO crafter ;
GRANT CREATE SESSION TO crafter ;
GRANT DEQUEUE ANY QUEUE TO crafter ;
GRANT QUERY REWRITE TO crafter ;
GRANT EXPORT FULL DATABASE TO crafter ;
GRANT CREATE PUBLIC DATABASE LINK TO crafter ;
GRANT RESUMABLE TO crafter ;
GRANT UNLIMITED TABLESPACE TO crafter ;
GRANT UNDER ANY VIEW TO crafter ;
GRANT DROP ANY OUTLINE TO crafter ;
GRANT CREATE ANY EDITION TO crafter ;
GRANT CREATE ANY ASSEMBLY TO crafter ;
GRANT ALTER ANY INDEXTYPE TO crafter ;
GRANT DROP ANY MATERIALIZED VIEW TO crafter ;
GRANT CREATE INDEXTYPE TO crafter ;
GRANT ALTER ANY SQL PROFILE TO crafter ;
GRANT ALTER SYSTEM TO crafter ;
GRANT DROP ANY SYNONYM TO crafter ;
GRANT GRANT ANY ROLE TO crafter ;
GRANT CREATE MATERIALIZED VIEW TO crafter ;
GRANT DROP ANY RULE SET TO crafter ;
GRANT MANAGE SCHEDULER TO crafter ;
GRANT DROP TABLESPACE TO crafter ;
GRANT SELECT ANY DICTIONARY TO crafter ;
GRANT IMPORT FULL DATABASE TO crafter ;
GRANT DELETE ANY MEASURE FOLDER TO crafter ;
GRANT DELETE ANY TABLE TO crafter ;
GRANT AUDIT SYSTEM TO crafter ;
GRANT ALTER ANY MATERIALIZED VIEW TO crafter ;
GRANT DEBUG ANY PROCEDURE TO crafter ;
GRANT CREATE PROFILE TO crafter ;
GRANT CREATE ANY MEASURE FOLDER TO crafter ;
GRANT UNDER ANY TYPE TO crafter ;
GRANT COMMENT ANY TABLE TO crafter ;
GRANT ALTER ANY DIMENSION TO crafter ;
GRANT CREATE ANY TYPE TO crafter ;
GRANT DROP ANY MEASURE FOLDER TO crafter ;
GRANT DROP PUBLIC DATABASE LINK TO crafter ;
GRANT CREATE CUBE TO crafter ;
GRANT CREATE DATABASE LINK TO crafter ;
GRANT INSERT ANY CUBE DIMENSION TO crafter ;
GRANT ALTER SESSION TO crafter ;
GRANT MANAGE ANY QUEUE TO crafter ;
GRANT ADMINISTER DATABASE TRIGGER TO crafter ;
GRANT AUDIT ANY TO crafter ;
GRANT MANAGE FILE GROUP TO crafter ;
```

This script grants all roles to `crafter` user. You can choose which roles you want to grant or to revoke.
There are also Oracle tools that can help you create user (Oracle enterprise management console and SQL developer)

MS SQL Server

1. Create the database. The database name here is `crafter`, if you change this you will need to modify `db.username` in `alfresco-global.properties` as well.

```
USE [master]
GO
CREATE DATABASE [crafter]
GO
ALTER DATABASE [crafter] SET ALLOW_SNAPSHOT_ISOLATION ON
GO
```

2. Create a user. Username/password here are crafter, if you change this you will need to modify alfresco-global.properties as well.

```
USE [master]
GO
CREATE LOGIN [crafter2] WITH PASSWORD=N'crafter',
DEFAULT_DATABASE=[crafter], CHECK_EXPIRATION=OFF, CHECK_POLICY=OFF
GO
```

Recommendation
Use Microsoft SQL Server Management Studio to create the database and user.
Setting `ALLOW_SNAPSHOT_ISOLATION` must however be executed as a SQL query, as it is not available in database properties.

PostgreSQL

1. Create a user. The username and password here are both `crafter`, if you change this you will need to modify `db.username` and `db.password` in `alfresco-global.properties` as well.

```
CREATE USER crafter WITH PASSWORD 'crafter';
```

2. Create the database. Database name here is crafter, if you change this you will need to modify alfresco-global.properties as well.

```
CREATE DATABASE crafter WITH ENCODING='UTF8' OWNER=crafter;
```

Install Alfresco Repository Application

System level Configuration

Make sure all of your nodes have an appropriate ULIMIT set for file handles. Alfresco, Solr, Lucene are all fairly greedy with respect to file handles.

Alfreco Web Application Archive

Place the war file for Alfresco+Crafter CMS in to the `webapps` directory

```
cd /opt/alfresco/apache-tomcat/webapps
wget
http://downloads.craftercms.org/craftercms/community/2-4-0/webapps/alfresco.war
```

Configure Alfresco

Content Store Path

/opt/alfresco/apache-tomcat/shared/classes/alfresco-global.properties

```
dir.root=/opt/alfresco/alf_data
```

Configure Database

MySQL

/opt/alfresco/apache-tomcat/shared/classes/alfresco-global.properties

```
db.driver=org.gjt.mm.mysql.Driver
db.username=crafter
db.password=crafter
db.name=crafter
db.url=jdbc:mysql://localhost:3306/crafter?useUnicode=yes&characterEncoding=UTF-8

hibernate.dialect=org.hibernate.dialect.MySQLInnoDBDialect
```

⚠ The system does not actually use Hibernate, but it does use this class

Oracle

/opt/alfresco/apache-tomcat/shared/classes/alfresco-global.properties

```
db.schema.update=true
db.driver=oracle.jdbc.OracleDriver
db.name=crafter
db.url=jdbc:oracle:thin:@localhost:1521:orcl
db.username=crafter
db.password=crafter
db.pool.initial=10
db.pool.max=20
hibernate.default_schema=CRAFTER

hibernate.dialect=org.alfresco.repo.domain.hibernate.dialect.AlfrescoOracle9Dialect
```

⚠ The system does not actually use Hibernate, but it does use this class

Microsoft SQL Server

/opt/alfresco/apache-tomcat/shared/classes/alfresco-global.properties

```
db.name=crafter
db.username=crafter
db.password=crafter
db.host=192.168.1.54
db.port=1433
db.driver=net.sourceforge.jtds.jdbc.Driver
db.url=jdbc:jtds:sqlserver://${db.host}:${db.port}/${db.name}
db.txn.isolation=4096
db.pool.validate.query=select 1

hibernate.dialect=org.alfresco.repo.domain.hibernate.dialect.AlfrescoSQLServerDialect
```

⚠ The system does not actually use Hibernate, but it does use this class

⚠ **Microsoft SQL Driver**
Please note that the SQL Server JDBC driver is not currently included in the bundle. It can be downloaded from http://sourceforge.net/projects/jtds/files/jtds/. The specific driver supported for Alfresco 4.x is **jtds-1.2.5.jar**. It should be placed in the `/opt/crafter/apache-tomcat/lib` directory.

PostgreSQL

/opt/alfresco/apache-tomcat/shared/classes/alfresco-global.properties

```
db.driver=org.postgresql.Driver
db.username=crafter
db.password=crafter
db.name=crafter
db.url=jdbc:postgresql://localhost:5432/${db.name}

hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
```

> ⚠ The system does not actually use Hibernate, but it does use this class

> ⚠ **PostgreSQL JDBC Driver**
> Please note that the PostgreSQL JDBC driver is not currently included in the bundle. It can be downloaded from http://jdbc.postgresql.org/download.html. The specific driver supported for Alfresco 4.x is **postgresql-9.0-8 02.jdbc4.jar**. It should be placed in the `/opt/crafter/apache-tomcat/lib` directory.

Configure Day Zero Production Settings

Day Zero is a term used by Alfresco referring to production level configurations that should be made and verified on the "first day" of any production installation. Note that the default configuration that typically ships/is found on-line with Alfresco is not production ready but rather intended to work on a laptop or simple desktop class machine for the purpose of demo. The settings displayed below are derived from the *Day Zero Configuration Guide*.

/opt/alfresco/apache-tomcat/shared/classes/alfresco-global.properties

```
db.pool.max=275
db.pool.idle=-1
db.pool.validate.query=/* See table below */
hibernate.jdbc.fetch_size=150
lucene.maxAtomicTransformationTime=0
system.usages.enabled=false
```

Database	db.pool.validate.query value
MySQL	SELECT 1
PostgreSQL	SELECT VERSION()
Oracle	SELECT 1 FROM DUAL
Microsoft SQL Server	SELECT 1

Configure Email

/opt/alfresco/apache-tomcat/shared/classes/alfresco-global.properties

```
mail.host=smtp.gmail.com
mail.port=465
mail.protocol=smtps
mail.username=myemailaddress@email.com
mail.password=emailpassword
mail.encoding=UTF-8
mail.smtps.starttls.enable=true
mail.smtps.auth=true
```

Install and Configure ImageMagick

ImageMagick is used by Alfresco to perform image transformations. This component is optional but recommended.

ImageMagick source and binary code is available on the following page: http://www.imagemagick.org/script/download.php.

On Ubuntu, ImageMagick is available as an APT package, and can be installed as follows: `sudo apt-get install imagemagick`.

Note that ImageMagick you may already have ImageMagick installed. To verify if ImageMagick is installed on the system, do the following:

```
which convert
/usr/bin/convert
```

/opt/alfresco/apache-tomcat/shared/classes/alfresco-global.properties

```
img.root=/usr
img.dyn=${img.root}/lib
img.exe=${img.root}/bin/convert
```

Linux

Once ImageMagick is installed, configure Alfresco as follows so it knows how to call it:

/opt/alfresco/apache-tomcat/shared/classes/alfresco-global.properties

```
img.root=/usr
```

Install License file

1. An enterprise license file can be obtained by opening a support ticket in the Alfresco Support Portal (https://support.alfresco.com/)
2. Copy the license file onto the server into the Alfresco license location

```
cp ~/myco-ent40.lic
/opt/alfresco/apache-tomcat/shared/classes/alfresco/extension/license
```

Install Alfresco's Solr

Alfresco uses Apache Solr to search within the Alfresco repository.

Configure Solr

Note that the Solr WAR file used by Alfresco is included in the CMS/Authoring Node Bundle (/opt/alfresco/alf_data/solr/apache-solr-1.4.1.war).

Set the `docBase` in the Apache Tomcat context file as well as the `solr/home` environment variable:

/opt/alfresco/apache-tomcat/conf/Catalina/localhost/solr.xml

```
<?xml version="1.0" encoding="utf-8"?>
<Context docBase="/opt/alfresco/alf_data/solr/apache-solr-1.4.1.war"
debug="0" crossContext="true">
<Environment name="solr/home" type="java.lang.String"
value="/opt/alfresco/alf_data/solr" override="true"/>
</Context>
```

Set the `data.dir.root` path for the `archive-SpacesStore` Solr core

/opt/alfresco/alf_data/solr/archive-SpacesStore/conf/solrcore.properties

```
data.dir.root=/opt/alfresco/alf_data/solr
```

Set the `data.dir.root` path for {{workspace-SpacesStore Solr core

/opt/alfresco/alf_data/solr/workspace-SpacesStore/conf/solrcore.properties

```
data.dir.root=/opt/alfresco/alf_data/solr
```

Install Alfresco's Share with Crafter CMS UX Plugins

Install Solr Web Application Archive

```
cd /opt/alfresco/apache-tomcat/webapps
wget
http://downloads.craftercms.org/craftercms/community/2-4-0/webapps/share.war
```

Configure Alfresco Share

1. Set the locations where website files are to be deployed to for previewing.

   /opt/alfresco/apache-tomcat/shared/classes/crafter/cstudio/extension/custom-cstudio-services-cfg.properties

   ```
   crafter.deployer.deployRoot=/opt/alfresco/preview-content/data
   crafter.deployer.metaDataRoot=/opt/alfresco/preview-content/metadata
   ```

2. Set the environment name. Example values are: `dev`, `stage`, `prod`.

   /opt/alfresco/apache-tomcat/shared/classes/crafter/cstudio/extension/custom-cstudio-services-cfg.properties

   ```
   environmentConfig.environment=prod
   ```

   ⚠ Site configuration needs to have defined environment overrides configuration for environment name defined in properties file.

3. Set the domain name that will be used to provide browser access Crafter Studio

   /opt/alfresco/apache-tomcat/shared/classes/crafter/cstudio/extension/server-config.properties

   ```
   authoringDomain=cstudio.mycompany.com
   ```

   ⚠ Domain defined as value for authoringDomain needs also to be defined in environment overrides configuration for cookie-domain property:

   ```
   <environment-config>
    // Other config ...
    <cookie-domain>cstudio.mycompany.com</cookie-domain>
    // Other config ...
   </environment-config>
   ```

Verify the Alfresco installation

http://www.slideshare.net/alfresco/alfresco-environment-validation-and-day-zero-configuration

Install Crafter Engine Preview Server

Install Crafter Engine Web Application Archive

```
cd /opt/alfresco/apache-tomcat/webapps
wget
http://downloads.craftercms.org/craftercms/community/2-4-0/webapps/ROOT.war
```

Configure Preview Mode

Crafter Engine preview mode is enabled by the following two Spring context files. They are provided in the CMS/Authoring Node Bundle.

/opt/alfresco/apache-tomcat/shared/classes/crafter/engine/extension/rendering-context.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:context="http://www.springframework.org/schema/context"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd">

 <import
resource="classpath*:crafter/engine/mode/preview/rendering-context.xml" />
</beans>
```

/opt/alfresco/apache-tomcat/shared/classes/crafter/engine/extension/services-context.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:context="http://www.springframework.org/schema/context"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd">

 <import
resource="classpath*:crafter/engine/mode/preview/services-context.xml" />
</beans>
```

Set the file system path where website files are to be previewed from. This path is represented by the `crafter.engine.site.default.root`

`Folder.path` property. Please note that the {`siteName`} parts in this path are substituted at runtime with the actual name of the site being previewed.

/opt/alfresco/apache-tomcat/shared/classes/crafter/engine/extension/server-config.properties

```
crafter.engine.site.default.rootFolder.path=file:/opt/alfresco/preview-content/content/wem-projects/{siteName}/{siteName}/work-area
crafter.engine.proxy.authoring.baseServiceUrl=http://localhost:8080/share
```

Install Crafter Search

This step is to support search capabilities in Crafter Engine Preview Mode

Install Crafter Search Web Application Archives

Download the Crafter Search web application archive.

```
cd /opt/alfresco/apache-tomcat/webapps
wget
http://downloads.craftercms.org/craftercms/community/2-4-0/webapps/crafter-search.war
```

Download the Solr Crafter web application archive.

```
cd /opt/alfresco/apache-tomcat/webapps
wget
http://downloads.craftercms.org/craftercms/community/2-4-0/webapps/solr-crafter.war
```

Download and unzip the Solr Crafter Solr configuration home folder.

```
TODO Review this step.  There seems to be an error here.
cd /opt/alfresco/apache-tomcat
http://downloads.craftercms.org/craftercms/community/2-4-0/webapps/solr-crafter.war
unzip solr-crafter.zip
```

Configure Crafter Search

Set the `docBase` in the Apache Tomcat context file as well as the `solr/home` environment variable.

/opt/alfresco/apache-tomcat/conf/Catalina/localhost/solr-crafter.xml

```
<?xml version="1.0" encoding="utf-8"?>
<Context docBase="/opt/alfresco/apache-tomcat/webapps/solr-crafter.war"
debug="0" crossContext="true">
<Environment name="solr/home" type="java.lang.String"
value="/opt/alfresco/apache-tomcat/solr-crafter" override="true"/>
</Context>
```

Configure document indexing

Set solr data folder property.

/opt/alfresco/apache-tomcat/solr-crafter/conf/solrcore.properties

```
solr.data.dir=/opt/alfresco/apache-tomcat/solr-crafter
```

Create `lib` folder inside `/opt/alfresco/apache-tomcat/solr-crafter` and copy solr-cell library from solr apache bundle (e.g. apache-solr-cell-3.6.0.jar. Create folder `extraction` inside `lib` folder and copy extraction libraries from solr apache bundle

> ⚠ Solr apache bundle can be downloaded from apache solr website http://lucene.apache.org/solr/downloads.html. Crafter search is using solr 3.6.0 version.
> solr-cell library can be found inside `dist` folder of unpacked bundle. Extraction libraries can be found inside `contrib/extraction/lib` folder of unpacked bundle

Configure solr to load solr-cell and extraction libraries.

/opt/alfresco/apache-tomcat/solr-crafter/conf/solrconfig.xml

```
...
<config>
... <!-- add lib directives -->
<lib dir="${solr.data.dir}/lib/" />
<lib dir="${solr.data.dir}/lib/" regex="apache-solr-cell-\d.*\.jar" />
<lib dir="${solr.data.dir}/lib/extraction" regex=".*\.jar" />
...
```

Configure Solr Cell update request handler

/opt/alfresco/apache-tomcat/solr-crafter/conf/solrconfig.xml

```
...
<!-- Solr Cell Update Request Handler

http://wiki.apache.org/solr/ExtractingRequestHandler

-->
<requestHandler name="/update/extract"
startup="lazy"
class="solr.extraction.ExtractingRequestHandler" >
<lst name="defaults">
<!-- All the main content goes into "text"... if you need to return
the extracted text or do highlighting, use a stored field. -->
<str name="fmap.content">text</str>
<str name="lowernames">false</str>
<str name="uprefix">ignored_</str>

<!-- capture link hrefs but ignore div attributes -->
<str name="captureAttr">true</str>
<str name="fmap.a">links</str>
<str name="fmap.div">ignored_</str>
</lst>
</requestHandler>
...
```

It is important to set `lowernames` property to false. By default it is set to true and that configuration will make `crafterSite` and `localId` attributes as lowercase fields in solr index.
`fmap.content` property sets field name for document content. By default it is set to `text` which will make document content indexed but not stored (search result will not have content snippet or content highlight features). If document content needs to be stored set value for this property to some field name that represents stored field (e.g. document_content_txt - *_txt are dynamic stored fields in default solr config)

Add search attachment processor to deployer target configuration and add it to target processors

```
...
<bean id="searchAttachmentProcessor"
class="org.craftercms.cstudio.publishing.processor.SearchAttachmentProcess
or">
<property name="searchService" ref="SearchService"/>
<property name="siteName"><value>preview</value></property>
<property name="supportedMimeTypes">
<list>
<value>application/pdf</value>
</list>
</property>
</bean>

...
```

Install Crafter Deployment Agent and Preview Solr Sync

The deployment agent is included in the bundle, so it doesn't need to be installed separately. If you are **not** using the bundle, then install it as follows:

```
mkdir -p /opt/alfresco/crafter-deployer
cd /opt/alfresco/crafter-deployer
wget
http://downloads.craftercms.org/craftercms/community/2-4-0/webapps/publish
ing-recivercstudio-publishing-receiver.zip
unzip cstudio-publishing-receiver.zip
```

By default preview sync to remote deployer is disabled. To enable preview sync on remote deployer following properties need to be added to `/opt/alfresco/apache-tomcat/shared/classes/crafter/cstudio/extension/custom-cstudio-services-cfg.properties`

custom-cstudio-services-cfg.properties

```
crafter.deployer.remote.enabled=true
crafter.deployer.server=localhost
crafter.deployer.port=9191
crafter.deployer.target=preview
crafter.deployer.password=admin
```

Add preview target configuration for crafter deployer `/opt/alfresco/crafter-deployer/conf/preview-target-context.xml`. This configuration will allow preview target to support multiple sites.

preview-target-context.xml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:context="http://www.springframework.org/schema/context"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
 http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd">
    <bean id="PreviewTarget"
class="org.craftercms.cstudio.publishing.target.PublishingTarget"
init-method="register">
     <property name="name"><value>preview</value></property>
     <property name="manager" ref="TargetManager"/>
     <property name="postProcessors">
         <list>
    <ref bean="PreviewSamplePostProcessor"/>
    <ref bean="PreviewSearchFlattenXmlProcessor"/>
    <!-- <ref bean="PreviewSearchAttachmentProcessor"/> -->
      </list>
     </property>
        <property name="params">
            <map>
                <entry key="root"><value>/opt/alfresco/preview-content</value></entry>
                <entry
key="contentFolder"><value>content/wem-projects/{siteId}/{siteId}/work-area</value></e
ntry>
                <entry
key="metadataFolder"><value>metadata/wem-projects/{siteId}/{siteId}/work-area</value><
/entry>
            </map>
        </property>
    </bean>
    <bean id="PreviewSearchProcessor"
class="org.craftercms.cstudio.publishing.processor.SearchUpdateProcessor">
  <property name="searchService" ref="PreviewSearchService"/>
  <property name="siteName"><value>preview</value></property>
 </bean>

    <bean id="PreviewSearchFlattenXmlProcessor"
class="org.craftercms.cstudio.publishing.processor.SearchUpdateFlattenXmlProcessor">
  <property name="searchService" ref="PreviewSearchService"/>
  <property name="siteName"><value>preview</value></property>
  <property name="includeElementXPathQuery" value="//include" />
 </bean>

    <bean id="PreviewSearchAttachmentProcessor"
class="org.craftercms.cstudio.publishing.processor.SearchAttachmentProcessor">
  <property name="searchService" ref="PreviewSearchService"/>
  <property name="siteName"><value>preview</value></property>
  <property name="supportedMimeTypes">
   <list>
    <value>application/pdf</value>
   </list>
  </property>
 </bean>
    <bean id="PreviewSearchService"
class="org.craftercms.search.service.impl.RestClientSearchService">
        <property name="serverUrl" value="http://localhost:8080/crafter-search"/>
    </bean>
    <bean id="PreviewSamplePostProcessor"
```

```
class="org.craftercms.cstudio.publishing.processor.SamplePostProcessor"/>

</beans>
```

Crafter Studio Configuration

In the repository, either in your blue-prints for the site or in the site configuration itself you need to modify the environment override files: environment-config.xml. This file can be find in alfresco repository `Company Home->cstudio->config->sites-># your site #->environment-overrides-># your environment #`. Environment name is defined in `custom-cstudio-services-cfg.properties.`

```
<environment-config>
// Other config ...
<cookie-domain>cstudio.mycompany.com</cookie-domain>
// Other config ...
</environment-config>
```

Delivery Node Installation

OS Application level User / Run As

1. Create a user that the application will run under such as '`cengine`'
2. Make sure '`cengine`' is the owner of all files in this setup

Disk Space / Locations

Content Storage

Our recommendation is to create a mount specifically for hosting site content

⚠ For the purpose of this document we'll put everything on a single environment. This is just for simplicity. We could create a more sophisticated topology but it would represent only one of many possibilities. The key here is that we will demonstrate where you can set up the various paths in the configuration. The work of mounting etc will depend on your infrastructure and will not be represented here.

Application Server

The recommended location for installing the delivery node software is the following:

```
/opt/crafter
```

Install the Bundle

Linux

1. Create an install location: `/opt/crafter`
2. Grant the `cengine` account ownership
3. Unzip the delivery node bundle inside this install location (see The Bundle)

Java

Your machine likely has Java installed.

Determine which version of JAVA: java -version

```
java version "1.6.0_26"
Java(TM) SE Runtime Environment (build 1.6.0_26-b03)
Java HotSpot(TM) 64-Bit Server VM (build 20.1-b02, mixed mode)
```

1. Crafter is only certified for 1.6.x at this time
2. Note that 64 bit JVMs will allow you to utilize more memory for the JVM

Configure JAVA HOME

1. Determine where java is installed by entering the following command

   which java

   ```
   /usr/bin/java
   ```

2. Modify (if required) the location of `JAVA HOME` for Apache Tomcat

   /opt/crafter/apache-tomcat/bin/setenv.sh

   ```
   export JAVA_HOME=/usr
   ```

Configure Apache Tomcat Ports

The default ports in the bundle are as follows:

Purpose	Port
Server	8005
HTTP/1.1 Connector	8080
AJP/1.3 Connector	8009

For the sake of users, the server is usually made available on port default HTTP port 80 ,which means no port is shown in the browser location bar. To accomplish this, use one of the two following approaches:

- Use Apache HTTPD in front of Crafter Engine. Apache HTTPD (or other web server) will run on port 80 and forward to Apache Tomcat on 8080 via modProxy or the like. This approach is preferred as it is more commonly understood and offers the other performance enhancements that Apache can bring to the environment.
- Use port forwarding (IPTABLES)

Edit the Apache Tomcat server configuration to set ports. Search and replace sever XML values. We recommend using a pattern:

/opt/crafter/apache-tomcat/conf/server.xml

```
8005 -> 9005
8443 -> 9443
8080 -> 9080
```

Automatic Start-up of Delivery Application Server

Linux

/etc/init.d/crafterengine

```
#!/bin/sh
export CENGINE_HOME_PATH='/opt/crafter'
export JAVA_OPTS='-Xms128m -Xmx512m -XX:MaxPermSize=512m -server
-Dfile.encoding=UTF8'
export JAVA_OPTS="${JAVA_OPTS} -Dalfresco.home=${CENGINE_HOME_PATH}
-Dcom.sun.management.jmxremote"

#Uncomment to connect with a debugger
#export JAVA_OPTS="$JAVA_OPTS -Xdebug -Xnoagent -Djava.compiler=NONE
-Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=5605"

tomcatPID=`ps -ef|grep "${CENGINE_HOME_PATH}/apache-tomcat"|grep java| awk
'{print $2}'`;
cur_dir=`pwd`;

case "$1" in
start)
cd ${CENGINE_HOME_PATH}/apache-tomcat;
sh bin/startup.sh
cd $cur_dir;
;;
stop)
cd ${CENGINE_HOME_PATH}/apache-tomcat;
sh ./bin/shutdown.sh;
if ! test -z $tomcatPID;
then
echo "Killing Tomcat process $tomcatPID";
kill -9 $tomcatPID;
fi
cd $cur_dir;
;;
restart)
$0 stop;
sleep 1;
$0 start
;;
*)
echo "Usage: $0 {start|stop|restart}";
exit 1;
;;
esac

exit 0
```

Install Crafter Search Server

Crafter Search is required to allow site search within Crafter Engine

Install Crafter Search Web Application Archives

Crafter Search

```
cd /opt/crafter/apache-tomcat/webapps
wget
http://downloads.craftercms.org/craftercms/community/2-4-0/webapps/crafter-search.war
```

Solr Crafter

```
cd /opt/crafter/apache-tomcat/webapps
wget
http://downloads.craftercms.org/craftercms/community/2-4-0/webapps/solr-crafter.war
```

Configure Crafter Search

Set the `docBase` in the Apache Tomcat context file as well as the `solr/home` environment variable.

/opt/crafter/apache-tomcat/conf/Catalina/localhost/solr-crafter.xml

```
<?xml version="1.0" encoding="utf-8"?>
<Context docBase="/opt/crafter/apache-tomcat/webapps/solr-crafter.war"
debug="0" crossContext="true">
<Environment name="solr/home" type="java.lang.String"
value="/opt/crafter/apache-tomcat/solr-crafter" override="true"/>
</Context>
```

Configure document indexing

Set solr data folder property.

/opt/crafter/apache-tomcat/solr-crafter/conf/solrcore.properties

```
solr.data.dir=/opt/crafter/apache-tomcat/solr-crafter
```

Create `lib` folder inside `/opt/crafter/apache-tomcat/solr-crafter` and copy solr-cell library from solr apache bundle (e.g. apache-solr-cell-3.6.0.jar. Create folder `extraction` inside `lib` folder and copy extraction libraries from solr apache bundle

> Solr apache bundle can be downloaded from apache solr website http://lucene.apache.org/solr/downloads.html. Crafter search is using solr 3.6.0 version.
> solr-cell library can be found inside `dist` folder of unpacked bundle. Extraction libraries can be found inside `contrib/extraction/lib` folder of unpacked bundle

Configure solr to load solr-cell and extraction libraries.

/opt/crafter/apache-tomcat/solr-crafter/conf/solrconfig.xml

```
...
<config>
... <!-- add lib directives -->
<lib dir="${solr.data.dir}/lib/" />
<lib dir="${solr.data.dir}/lib/" regex="apache-solr-cell-\d.*\.jar" />
<lib dir="${solr.data.dir}/lib/extraction" regex=".*\.jar" />
...
```

Configure Solr Cell update request handler

/opt/crafter/apache-tomcat/solr-crafter/conf/solrconfig.xml

```
...
<!-- Solr Cell Update Request Handler

http://wiki.apache.org/solr/ExtractingRequestHandler

-->
<requestHandler name="/update/extract"
startup="lazy"
class="solr.extraction.ExtractingRequestHandler" >
<lst name="defaults">
<!-- All the main content goes into "text"... if you need to return
the extracted text or do highlighting, use a stored field. -->
<str name="fmap.content">text</str>
<str name="lowernames">false</str>
<str name="uprefix">ignored_</str>

<!-- capture link hrefs but ignore div attributes -->
<str name="captureAttr">true</str>
<str name="fmap.a">links</str>
<str name="fmap.div">ignored_</str>
</lst>
</requestHandler>
...
```

It is important to set `lowernames` property to false. By default it is set to true and that configuration will make `crafterSite` and `localId` attributes as lowercase fields in solr index.
`fmap.content` property sets field name for document content. By default it is set to `text` which will make document content indexed but not stored (search result will not have content snippet or content highlight features). If document content needs to be stored set value for this property to some field name that represents stored field (e.g. document_content_txt - *_txt are dynamic stored fields in default solr config)

Add search attachment processor to deployer target configuration and add it to target processors

```
...
<bean id="searchAttachmentProcessor"
class="org.craftercms.cstudio.publishing.processor.SearchAttachmentProcessor">
<property name="searchService" ref="SearchService"/>
<property name="siteName"><value>preview</value></property>
<property name="supportedMimeTypes">
<list>
<value>application/pdf</value>
</list>
</property>
</bean>

...
```

Install Crafter Engine

Install Crafter Engine Web Application Archive

Download the Crafter Engine war file into the Apache Tomcat `webapps` directory.

```
cd /opt/crafter/apache-tomcat/webapps
wget
http://downloads.craftercms.org/craftercms/community/2-4-0/webapps/ROOT.war
```

Configure a Shared Classloader Location in Apache Tomcat

Configure a shared classloader location in which Crafter Engine configuration files will reside. Let's assume this location to be `/opt/crafter/apache-tomcat/shared/classes`. Open the `catalina.properties` file and modify the following line:

/opt/crafter/apache-tomcat/conf/catalina.properties

```
shared.loader=${catalina.base}/shared/classes
```

Configure the Crafter Engine Default Site Root Folder

Configure the location from which Crafter Engine serves the website files. For the purpose of this documentation, let's assume this location to be `/opt/crafter/site-root/content`. Create a `server-config.properties` file for this as shown below:

/opt/crafter/apache-tomcat/shared/classes/crafter/engine/extension/server-config.properties

```
crafter.engine.site.default.rootFolder.path=file:/opt/crafter/crafter-depl
oyer/target/sample/content
```

The website that Crafter Engine will serve will live directly under this path (/opt/crafter/crafter-deployer/target/sample/content). Once the site is deployed using the Crafter Deployment Agent, a directory structure like this would be in place:

```
/opt/crafter/crafter-deployer/target/sample/content
??? site
??? static-assets
??? templates
```

Configure the Default Site

Configure what the name of the default site is. This is used to retrieve site specific content from Crafter Search.

/opt/crafter/apache-tomcat/shared/classes/crafter/engine/extension/server-config.properties

```
crafter.engine.site.default.name=mysite
```

Configure the Crafter Search Server URL used by Crafter Engine

The following property has to be set so that Crafter Engine can interact with Crafter Search. In the example below, Crafter Search runs on the same host as Crafter Engine (`localhost`) and port `8080`.

/opt/crafter/apache-tomcat/shared/classes/crafter/engine/extension/server-config.properties

```
crafter.engine.search.server.url=http://localhost:8080/crafter-search
```

⚠ If tomcat port numbers for delivery node have been changed, apply port number change for property value accordingly.

Configure the Solr URL used by Crafter Search

The following property has to be set so that Crafter Search can connect to Solr. In the example below, Solr runs on the same host as Crafter Search (`localhost`) and port `8080.`

/opt/crafter/apache-tomcat/shared/classes/crafter/search/extension/server-config.properties

```
crafter.search.solr.server.url=http://localhost:8080/solr-crafter
```

If tomcat port numbers for delivery node have been changed, apply port number change for property value accordingly.

Install Crafter Deployment Agent

The Deployment Agent is included in the bundle, so you do not need to install it. If you are not using the bundle, you can install and configure it as follows:

1. Download the deployment agent from: http://downloads.craftercms.org/craftercms/community/2-2-2/crafter-deployer.zip
2. Create an install directory: `/opt/crafter/crafter-deployer`
3. Unzip the deployment agent into the install directory `/opt/craftercrafter-CMS-deployer`
4. By default the server will listen for HTTP requests on port 9191 and will require the password '`admin`'
5. To customize deployment agent properties and targets, change the following file names (paths relative to install directory):
    a. `conf/custom-receiver-context.xml.sample` to `conf/custom-receiver-context.xml`
    b. `conf/custom-shutdown-context.xml.sample` to `conf/custom-shutdown-context.xml`
    c. `conf/custom-receiver.properties.sample` to `conf/custom-receiver.properties`

    /opt/craftercrafter-wem-deployer/conf/custom-receiver.properties

    ```
    receiver.port:9191
    receiver.password=admin
    ```

6. To configure or override the deployment targets, edit SampleTarget in conf/sample-target-context.xml or add a new target:

/opt/craftercrafter-wem-deployer/conf/sample-target-context.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:context="http://www.springframework.org/schema/context"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd">


<bean id="SampleTarget"
class="org.craftercms.cstudio.publishing.target.PublishingTarget"
init-method="register">
<property name="name"><value>sample</value></property>
<property name="manager" ref="TargetManager"/>
<property name="postProcessors">
<list>
<ref bean="SearchUpdateProcessor" />
<ref bean="SampleCacheInvalidate" />
</list>
</property>
<property name="params">
<map>
<entry key="root"><value>target/sample</value></entry>
<entry key="contentFolder"><value>content</value></entry>
<entry key="metadataFolder"><value>metadata</value></entry>
```

```
</map>
</property>
</bean>

<bean id="SampleCacheInvalidate"
class="org.craftercms.cstudio.publishing.processor.CacheInvalidatePostProcessor">

<property name="cacheInvalidateUrl"
value="http://localhost:8080/api/1/cache/clear_all" />
</bean>

<bean id="SearchService"
class="org.craftercms.search.service.impl.RestClientSearchService">
<property name="serverUrl" value="http://localhost:8080/crafter-search"/>
</bean>

<bean id="SearchUpdateProcessor"
class="org.craftercms.cstudio.publishing.processor.SearchUpdateProcessor">
<property name="searchService" ref="SearchService"/>
<property name="siteName"><value>sample</value></property>
</bean>

<bean id="searchFlattenXmlProcessor"
class="org.craftercms.cstudio.publishing.processor.SearchUpdateFlattenXmlProcesso
r">
<property name="searchService" ref="SearchService"/>
<property name="siteName"><value>sample</value></property>
<property name="includeElementXPathQuery" value="//include" />
<property name="charEncoding" value="UTF-8" />
</bean>

<bean id="SamplePostProcessor"
class="org.craftercms.cstudio.publishing.processor.SamplePostProcessor"/>
```

```
</beans>
```

If tomcat port numbers for delivery node have been changed, apply port number change for property value accordingly.

a. name property is the key to find a deployment target from authoring end.
b. set the deployment target location by updating root property value.
c. add custom post-deployment processors. Custom post-deployment processors must implement org.craftercms.cstudio.publishing.processor.PublishingProcessor interface.

base-context.xml

```
<bean id="SamplePostProcessor"
class="org.craftercms.cstudio.publishing.processor.SamplePostProcessor"/>
```

Automatic Start-up of Crafter Deployment Agent

Linux

/etc/init.d/crafterdeployer

```
#!/bin/sh
export CDEPLOYER_HOME_PATH='/opt/crafter/crafter-deployer'

deployerPID=`ps -ef|grep "${DEPLOYER_HOME_PATH}"|grep java| awk '{print
$2}'`;
cur_dir=`pwd`;

case "$1" in
start)
cd ${CDEPLOYER_HOME_PATH};
sh start-deploy-agent.sh
cd $cur_dir;
;;
stop)
cd ${CDEPLOYER_HOME_PATH};
sh stop-deploy-agent.sh;
if ! test -z $deployerPID;
then
echo "Killing Deployer process $deployerPID";
kill -9 $deployerPID;
fi
cd $cur_dir;
;;
restart)
$0 stop;
sleep 1;
$0 start
;;
*)
echo "Usage: $0 {start|stop|restart}";
exit 1;
;;
esac

exit 0
```

Windows Service

The deployment agent can be registered as Windows Service using Apache Prunsrv. Below is an example command.

```
prunsrv //IS//CrafterDeployer --DisplayName="Crafter Deployer"
--Install=C:\opt\prunsrv\prunsrv.exe --StartMode=java --Jvm=auto
--JavaHome=C:\opt\java\jdk1.7.0 --StartPath="C:\opt\crafter\crafter-deployer"
--StartClass=org.craftercms.cstudio.publishing.PublishingReceiverMain  --StopMode=java
--StopPath="C:\opt\crafter\crafter-deployer"
--StopClass=org.craftercms.cstudio.publishing.StopServiceMain
--Classpath="C:\opt\crafter\crafter-deployer"
++JvmOptions="-Djava.ext.dirs=C:\opt\crafter\crafter-deployer"
--LogPath="C:\opt\crafter\crafter-deployer"
 --LogPrefix="deployment" --LogLevel=Debug --StdOutput=auto --StdError=auto
--PidFile=pid.txt --Startup=auto --Description="Crafter Deployer"
```

Configuring Alfresco Clustering with Crafter Studio

System Architecture

- Three virtual machines running Debian Linux
    - Cluster Node 1 - 2G RAM
        - Alfresco bundle
    - Cluster Node 2 - 2G RAM
        - Alfresco bundle
    - Load Balancer - 512M RAM
        - Apache 2
        - MySQL shared DB
        - Alfresco repo (NFS)

Create Virtual Machines

1. Download Debian linux installation disk image (http://hammurabi.acc.umu.se/debian-cd/6.0.6/amd64/iso-cd/debian-6.0.6-amd64-netinst.iso)
2. Create new VirtualBox appliance using installation image
3. Install Sun JDK 6
4. Install SSH
5. Shutdown Virtual Machine
6. Clone two more Virtual Machines
7. Configure network and memory for all 3 VM boxes

MySQL

1. Install MySQL on Load Balancer Virtual Machine
2. Create `'crafter'` database and `'crafter'` user

Alfresco repository

Install and configure NFS server on Load Balancer

ⓘ **Reference Documentation**
http://www.debianhelp.co.uk/nfs.htm

1. `apt-get install nfs-kernel-server nfs-common portmap`

2. Edit `/etc/exports` file to add `alf_data` repository to NFS

   ```
   /opt/alfresco/alf_data 192.168.56.101(rw,sync,no_subtree_check,no_root_squash)
   192.168.56.103(rw,sync,no_subtree_check,no_root_squash)
   # 192.168.56.101 and 192.168.56.103 are IP addresses for Cluster Node 1 and
   Cluster Node 2
   ```

3. Execute: `exportfs -a`

Install and Configure NFS Client on Cluster Nodes

> **Reference Documentation**
> http://www.debianhelp.co.uk/nfs.htm

1. `apt-get install nfs-common portmap`
2. Edit `/etc/fstab` file to add `alf_data` NFS volume as mount point

   ```
   192.168.56.102:/opt/alfresco/alf_data /opt/alfresco/nfs_alf_data nfs
   rw,rsize=4096,hard,intr,async,nodev,nosuid 0 0
   # 192.168.56.102 is the Load Balancer IP address
   ```

3. Execute: `mount -a`

Install Crafter Studio on Cluster Node 1

1. Use the Installation Guide to install Crafter Studio on Cluster Node 1. *Do not start alfresco at any point during installation!*
2. Configure Alfresco to use the MySQL database on the Load Balancer VM Node (`/opt/alfresco/apache-tomcat/shared/classes/alfresco-global.properties`):

   ```
   ### database connection properties ###
   db.driver=org.gjt.mm.mysql.Driver
   db.username=crafter
   db.password=crafter
   db.name=crafter
   db.url=jdbc:mysql://192.168.56.102:3306/crafter?useUnicode=yes&characterEncoding=
   UTF-8
   hibernate.dialect=org.hibernate.dialect.MySQLInnoDBDialect
   ```

3. Configure the Alfresco repository to use the NFS volume for data stores (`/opt/alfresco/apache-tomcat/shared/classes/alfresco-global.properties`):

   ```
   dir.root=/opt/alfresco/nfs_alf_data
   ```

4. Change the keystore path for the Solr indexes (Solr indexes are not shared) (`/opt/alfresco/apache-tomcat/shared/classes/alfresco-global.properties`):

```
dir.keystore=/opt/alfresco/alf_data/keystore
```

5. Start Alfresco on Cluster Node 1

Install Crafter Studio on Cluster Node 2

1. Use the Installation Guide to install Crafter Studio on Cluster Node 1. *Do not start Alfresco at any point during installation!*
2. Configure Alfresco to use MySQL database on Load Balancer (`apache-tomcat/shared/classes/alfersco-global.properties`)

   ```
   ### database connection properties ###
   db.driver=org.gjt.mm.mysql.Driver
   db.username=crafter
   db.password=crafter
   db.name=crafter
   db.url=jdbc:mysql://192.168.56.102:3306/crafter?useUnicode=yes&characterEncoding=UTF-8
   hibernate.dialect=org.hibernate.dialect.MySQLInnoDBDialect
   ```

3. Configure Alfresco repository to use NFS volume for data stores (`/opt/alfresco/apache-tomcat/shared/classes/alfresco-global.properties`):

   ```
   dir.root=/opt/alfresco/nfs_alf_data
   ```

4. Change keystore path for Solr indexes (Solr indexes are not shared) (`/opt/apache-tomcat/shared/classes/alfresco-global.properties`):

   ```
   dir.keystore=/opt/alfresco/alf_data/keystore
   ```

5. Initiate complete index re-build on Cluster Node 2 (`/opt/alfresco/apache-tomcat/shared/classes/alfresco-global.properties`):

   ```
   index.recovery.mode=FULL
   ```

6. Start Alfresco on Cluster Node 2

Configure Alfresco Cluster

1. Stop Alfresco on both nodes
2. Apply the following configuration changes to both nodes:

/opt/alfresco/apache-tomcat/shared/classes/alfresco-global.properties

```
alfresco.cluster.name=cstudiocluster
alfresco.tcp.initial_hosts=192.168.56.101[7800],192.168.56.103[7800]
alfresco.jgroups.defaultProtocol=TCP
index.recovery.mode=AUTO
index.tracking.cronExpression=0/5 * * * * ?
index.tracking.reindexLagMs=5000
```

3. Create Ehcache cluster context

```
cd /opt/alfresco/apache-tomcat/shared/classes/extension
cp ehcache-custom.xml.sample.cluster ehcache-custom.xml
```

4. Edit `ehcache-custom.xml`
    a. Replace the `cacheManagerPeerProviderFactory` bean definition with the following:

```
<cacheManagerPeerProviderFactory
class="net.sf.ehcache.distribution.RMICacheManagerPeerProviderFa
ctory"
properties="peerDiscovery=automatic,
multicastGroupAddress=230.0.0.1,
multicastGroupPort=4446"/>
```

    b. Remove the following default definition of the `cacheManagerPeerListenerFactory` bean:

```
<cacheManagerPeerListenerFactory
class="net.sf.ehcache.distribution.RMICacheManagerPeerListenerFa
ctory"
properties="socketTimeoutMillis=10000"
/>
```

    c. Uncomment the extended definition by removing the comment lines `<!-` and `-!>` before and after the following section:

```
<cacheManagerPeerListenerFactory
class="net.sf.ehcache.distribution.RMICacheManagerPeerListenerFa
ctory"
properties="hostName=${alfresco.ehcache.rmi.hostname},
port=${alfresco.ehcache.rmi.port},
remoteObjectPort=${alfresco.ehcache.rmi.remoteObjectPort},
socketTimeoutMillis=${alfresco.ehcache.rmi.socketTimeoutMillis}"
/>
```

d. Set the Ehcache RMI hostname

apache-tomcat/shared/classes/alfersco-global.properties

```
alfresco.ehcache.rmi.hostname=192.168.56.101
```

5. Share configuration
   a. Edit `/opt/alfresco/apache-tomcat/shared/classes/alfresco/web-extension/share-config-custom.xml`: Hard-wire each Share instance to its own Alfresco instance, bypassing the load balancer.
   This can be achieved by populating each `share-config-custom.xml` file with a host name and port number that is not behind your load balancing mechanism.

Install Apache Load Balancer

1. Install Apache 2 on the Load Balancer Node

```
apt-get install apache2 apache2-doc apache2-utils
```

2. Enable the `proxy`, `proxy_http`, `proxy_balancer` and `headers` Apache modules:

```
a2enmod proxy
a2enmod proxy_http
a2enmod proxy_balancer
a2enmod headers
```

3. Create a new virtual host for the CMS domain:

```
<VirtualHost *:80>
ServerName cms.mydomain.com

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
LogLevel warn

ProxyStatus On
ProxyPreserveHost On

ProxyPass / balancer://cstudioCluster/

Header add Set-Cookie "CLUSTER=.%{BALANCER_WORKER_ROUTE}e; path=/"
env=BALANCER_ROUTE_CHANGED

<Proxy balancer://cstudioCluster>
BalancerMember http://192.168.56.101:8080 route=1
BalancerMember http://192.168.56.103:8080 route=2
ProxySet stickysession=CLUSTER
</Proxy>
</VirtualHost>
```

References

- http://docs.alfresco.com/4.1/index.jsp?topic=%2Fcom.alfresco.enterprise.doc%2Fconcepts%2Fha-intro.html
- http://www.ixxus.com/blog/2012/01/setting-up-alfresco-cluster/
- http://httpd.apache.org/docs/2.2/mod/mod_proxy_balancer.html

Operational Tips

JVM Character Sets

We have found it is best to tell the JVM to use a specific characterset rather than defaulting to the system character set. To do this, add the following to `catalina.sh` in `BASEDIR/tomcat/bin`:

```
JAVA_OPTS="$JAVA_OPTS -Dfile.encoding=UTF8"
```

JVM Debugging

Add the following to `catalina.sh` in `BASEDIR/tomcat/bin` to enable remote debugging:

```
JAVA_OPTS="$JAVA_OPTS -Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=5005
```

In your IDE configure a remote JVM debugging session and configure it to connect to port 5005.

Connecting Apache

Putting Apache in front of your Apache Tomcat application server has many benefits:

- It is already set to run on default HTTP port 80
- It allows you to run multiple sites with virtual hosting
- It can be used to implement security, redirection, caching and compression

Setting Up Apache Virtual Hosting for a Crafter Site

1. Ensure an AJP 1.3 Connector is configured in Apache Tomcat:

   /opt/crafter/apache-tomcat/conf/server.xml

   ```
   <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
   ```

2. Create an Apache virtual host configuration file:

   /etc/apache2/sites-available/mysite

   ```
   <VirtualHost *:80>

   ServerName mysite
   ServerAlias www.mysite.com

   # -----------
   # Static assets
   # -----------

   # Point this to where the Crafter site is published
   DocumentRoot /path/to/site-root

   # -----------
   # Logging
   # -----------

   LogLevel warn
   CustomLog /var/log/apache2/mysite.log combined
   ErrorLog /var/log/apache2/mysite.log

   # -----------
   # Compression
   # -----------
   ```

```
<Location />
# Insert filter
SetOutputFilter DEFLATE

# Netscape 4.x has some problems...
BrowserMatch ^Mozilla/4 gzip-only-text/html

# Netscape 4.06-4.08 have some more problems
BrowserMatch ^Mozilla/4\.0[678] no-gzip

# MSIE masquerades as Netscape, but it is fine
# BrowserMatch \bMSIE !no-gzip !gzip-only-text/html

# NOTE: Due to a bug in mod_setenvif up to Apache 2.0.48
# the above regex won't work. You can use the following
# workaround to get the desired effect:
BrowserMatch \bMSI[E] !no-gzip !gzip-only-text/html

# Don't compress images
SetEnvIfNoCase Request_URI \.(?:gif|jpe?g|png|mp4|mp4v|zip)$ no-gzip
dont-vary

# Make sure proxies don't deliver the wrong content (don't use this
with a CDN)
# Header append Vary User-Agent env=!dont-vary
</Location>

# -----------
# Expiration
# -----------

ExpiresActive On
ExpiresByType image/jpeg "access plus 1 year"
ExpiresByType image/png "access plus 1 year"
ExpiresByType image/gif "access plus 1 year"
ExpiresByType image/swf "access plus 1 year"
ExpiresByType application/x-javascript "access plus 1 year"
ExpiresByType text/css "access plus 1 year"
Header unset ETag

# -------------
# URL Rewriting
# -------------

RewriteEngine On
# Only rewrite (proxy) for WEB PAGES, so skip anything within
'/static-resources'
RewriteCond ${REQUEST_URI} !^/static-resources/.*$
RewriteRule ^/(.*)$ ajp://localhost:8009/$1 [P]
```

```
</VirtualHost>
```

3. Enable the virtual host

```
a2ensite mysite
```

4. Reload apache2 to activate the virtual host

```
/etc/init.d/apache2 reload
```

When Mod Proxy is not installed

1. sudo apt-get install libapache2-mod-proxy-html
2. apt-get install libxml2-dev

apache2.conf

```
LoadModule proxy_module /usr/lib/apache2/modules/mod_proxy.so
LoadModule proxy_http_module
/usr/lib/apache2/modules/mod_proxy_http.so
LoadModule headers_module /usr/lib/apache2/modules/mod_headers.so
LoadModule deflate_module /usr/lib/apache2/modules/mod_deflate.so
#LoadFile /usr/lib/libxml2.so
```

Alfresco & Share Settings Related to Web Server Proxying

With a web server in front of Alfresco, the browser resolvable host name is going to be different. One might also introduce SSL encryption on the virtual host configuration. Below are some settings that should be changed accordingly.

/opt/alfresco/apache-tomcat/shared/classes/alfresco-global.properties

```
# Change to domain used in web server virtual host
alfresco.host=127.0.0.1
# Change to port 80 or 443 as needed
alfresco.port=8080
# Change to https as needed
alfresco.protocol=http

# Change to domain used in web server virtual host
share.host=127.0.0.1
# Change to port 80 or 443 as needed
share.port=8080
# Change to https as needed
share.protocol=http
```

Database Tuning

TODO

Monitoring

- Load average
- Available memory
- Available file space
- Available file handles
- Java process existence
- Log mining (for errors and warns)

Turning up logging

"/opt/alfresco/apache-tomcat/conf/logging.properties"

```
org.apache.catalina.core.ContainerBase.[Catalina].level = INFO
org.apache.catalina.core.ContainerBase.[Catalina].handlers =
java.util.logging.ConsoleHandler
```

Dropping Alfresco repository

1. Stop Tomcat that runs Alfresco web application
2. Remove Alfresco file system content stores at alf_data/contentstore and alf_data/contentstore.deleted
3. Empty tomcat temp folder at apache-tomcat/temp
4. Drop and create again alfresco database
5. Clear Solr indexes

Clearing Solr indexes for a re-index

1. Stop Tomcat that runs Solr web application

2. Remove index data of archive core at alf_data/solr/archive/SpacesStore
3. Remove index data of workspace core at alf_data/solr/workspace/SpacesStore
4. Remove cached content model info of archive core at alf_data/solr/archive-SpacesStore/alfrescoModels/*
5. Remove cached content model info of workspace core at alf_data/solr/workspace-SpacesStore/alfrescoModels/*
6. Restart Tomcat that runs Solr web application
7. Wait. This takes a while even with a small alfresco repo. Solr polls for updates so some of the wait is in part due to the time Solr is sleeping.

Setting up Log Rotation

Linux

This section describes how to setup log rotation on Linux (Ubuntu) using logrotate. If logrotate is not yet on the system, it can be installed as follows:

Install logrotate

```
sudo apt-get install logrotate
```

Once installed, logrotate will be scheduled to run daily:

/etc/cron.daily/logrotate

```
#!/bin/sh

# Clean non existent log file entries from status file
cd /var/lib/logrotate
head -1 status > status.clean
sed 's/"//g' status | while read logfile date
do
[ -e "$logfile" ] && echo "\"$logfile\" $date"
done >> status.clean
mv status.clean status

test -x /usr/sbin/logrotate || exit 0
/usr/sbin/logrotate /etc/logrotate.conf
```

In the above daily cron entry, one can see that logrotate is executed with the main configuration file `/etc/logrotate.conf`:

/etc/logrotate.conf

```
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress

# packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own wtmp, or btmp -- we'll rotate them here
/var/log/wtmp {
missingok
monthly
create 0664 root utmp
rotate 1
}

/var/log/btmp {
missingok
monthly
create 0660 root utmp
rotate 1
}

# system-specific logs may be configured here
```

This main configuration file defines some global directives (such as `rotate 4` and `create`) as well as an include directory (`/etc/logrotate.d` in which individual configurations can be dropped for inclusion. Note that the global configuration directives can be overridden in the included configuration files.

Below is an example of a logrotate configuration file to be included.

/etc/logrotate.d/cms

```
/opt/alfresco/apache-tomcat/catalina.out {
size 1M
copytruncate
create 640 crafter nogroup
rotate 5
compress
}
```

This will rotate the Apache Tomcat generated `catalina.out` file in the following manner:

- **`size 1M`** - Rotate the file when it has become bigger than 1 MB.
- **`copytruncate`** - Truncate the original log file in place after creating a copy, instead of moving the old log file and optionally creating a new one
- **`create 640 crafter nogroup`** - Immediately after rotation the log file is created with mode `640`, owner `crafter` and group `nogroup`
- **`rotate 5`** - Log files are rotated 5 times before being removed
- **`compress`** - Old versions of log files are compressed (with gzip by default)

Setting up LDAP

> ⚠ LDAP installations vary from organization to organization because the data structures used by organizations vary. Given this the install can only be described in a general way when it comes to these specifics. Installations can require communication between different teams in your organization (IT and Development for example) and should be taken in to account when planning for the setup.

Then acquire the following items:

- JXplorer or similar LDAP browser or administration tool
- The LDAP server domain name or IP address and port number (default 389)
- The LDAP principal name and password for Alfresco
- The Base Distinguish Name (Base DN), for example, "ou=ACME,dc=acmehq,dc=com"

Login to LDAP to validate above items. Note that some LDAP servers require the login name to be in the format of name@domain, for example, *alfresco.ldap@acmehq*.

LDAP Groups vs. Alfresco Groups

Organizations usually prefer administrating all users and groups in LDAP and require Alfresco to honor the LDAP user credentials for authentication and the LDAP groups for permissions. When Crafter Studio creates a site (for example, "ACME"), five Alfresco groups will be created:

- site_ACME
- site_ACME_SiteManager
- site_ACME_SiteCollaborator
- site_ACME_SiteContributor
- site_ACME_SiteConsumer

where "site_ACME" is the parent group of the remaining four. Note that these groups are only visible in Share Admin Console, not visible in Alfresco Admin Console.

Depending on their organizational needs, LDAP administration may create related groups in one of two ways or both:

- Create 5 LDAP groups parallel to the 5 Alfresco groups. This allows permission control on a per site basis.

  > ⓘ Do **NOT** name LDAP groups identical to their Alfresco counterparts. Add a prefix such as "Alfresco_".

  - Alfresco_site_ACME
  - Alfresco_site_ACME_SiteManager
  - Alfresco_site_ACME_SiteCollaborator
  - Alfresco_site_ACME_SiteContributor
  - Alfresco_site_ACME_SiteConsumer

- Create 5 generic LDAP groups, for example:
  - Alfresco_users
  - Alfresco_SiteManager
  - Alfresco_SiteCollaborator
  - Alfresco_SiteContributor
  - Alfresoc_SiteConsumer

It is recommended to create these LDAP groups under an isolated Organization Unit (OU). This allows the synchronization from LDAP to Alfresco be limited to a small set.

Alfresco LDAP Configuration Settings

Add the following set of lines to *tomcat/shared/classes/alfresco-global.properties*.

```
### LDAP integration ###
authentication.chain=alfrescoNtlm1:alfrescoNtlm,ldap1:ldap-ad
synchronization.synchronizeChangesOnly=false
ldap.authentication.active=true
ldap.authentication.java.naming.security.authentication=simple
ldap.authentication.userNameFormat=
ldap.authentication.java.naming.provider.url=ldap://ldap.acme.com:389
ldap.synchronization.active=true
ldap.synchronization.java.naming.security.authentication=simple
ldap.synchronization.java.naming.security.principal=alfresco.ldap@acmehq
ldap.synchronization.java.naming.security.credentials=************
ldap.synchronization.userSearchBase=ou=ACME,dc=acmehq,dc=com
ldap.synchronization.groupSearchBase=ou=ACME,dc=acmehq,dc=com
ldap.synchronization.userIdAttributeName=sAMAccountName
ldap.synchronization.userFirstNameAttributeName=givenName
ldap.synchronization.userLastNameAttributeName=sn
ldap.synchronization.groupIdAttributeName=extensionAttribute9
ldap.synchronization.modifyTimestampAttributeName=whenChanged
ldap.synchronization.timestampFormat=yyyyMMddHHmmss'.0Z'
ldap.synchronization.groupQuery=(objectclass\=group)
ldap.synchronization.groupDifferentialQuery=(&(objectclass\=group)(!(whenChanged<\={0}
)))
ldap.synchronization.personQuery=(&(objectclass\=user)(userAccountControl:1.2.840.1135
56.1.4.803:=512))
ldap.synchronization.personDifferentialQuery=(&(objectclass\=user)(userAccountControl:
1.2.840.113556.1.4.803:=512)(!(whenChanged<\={0})))
```

The values of most of these settings must be validated in the LDAP browser. The primary documentation on these settings are:

- Alfresco Enterprise - LDAP configuration properties
- Alfresco Authentication Subsystem Wiki
- Alfresco Synchronization Subsystem Wiki
- LDAP Search Filters (query syntax)

Here are some additional notes:

- **authentication.chain=alfrescoNtlm1:alfrescoNtlm,ldap1:ldap-ad**
  If you are using OpenLDAP instead of Active Directory, change "ldap1:ldap-ad" to "ldap1:ldap". Keeping "alfrescoNtlm1:alfrescoNtlm" up front allows Alfresco user credentials take precedence over LDAP credentials. For example, it preserves the 'admin' account created during Alfresco installation.
- **synchronization.synchronizeChangesOnly=false**
  This makes the periodic LDAP sync (default daily at midnight) to perform a full sync. Set to true to perform a differential sync.
- **ldap.authentication.active=true**
  Enables LDAP authentication. Authentication can be enabled or disabled (set to false) independent of synchronization.
- **ldap.authentication.java.naming.security.authentication=simple**
  This is a common authentication method.
- **ldap.authentication.userNameFormat=**
  Setting this value to empty is intentional (do not remove it). This allow Alfresco to query the LDAP server for the correct user name format, which is the most flexible behavior.
- **ldap.authentication.java.naming.provider.url=ldap://ldap.acme.com:389**
  The LDAP server domain name and port number.
- **ldap.synchronization.active=true**
  Enables LDAP synchronization. Synchronization can be enabled or disabled (set to false) independent of authentication.
- **ldap.synchronization.java.naming.security.authentication=simple**

This is a common synchronization method.

- **ldap.synchronization.java.naming.security.principal=alfresco.ldap@acmehq**
  LDAP login principal name for Alfresco.
- **ldap.synchronization.java.naming.security.credentials=**********
  LDAP login principal password.
- **ldap.synchronization.userSearchBase=ou=ACME,dc=acmehq,dc=com**
  The Distinguish Name (DN) base for all LDAP users who will have login access to Alfresco. It is recommended to create an isolated Organization Unit (OU) to limit the access to Alfresco and to reduce the LDAP user synchronization size.
- **ldap.synchronization.groupSearchBase=ou=ACME,dc=acmehq,dc=com**
  The Distinguish Name (DN) base for all LDAP groups that will be available to Alfresco. It is recommended to create an isolated Organization Unit (OU) for all Alfresco related groups to reduce the LDAP group synchronization size.
- **ldap.synchronization.userIdAttributeName=sAMAccountName**
  The LDAP attribute that stores the network username used by people to login to the network, VPN, email, etc. This username will also be used to login to Alfresco. Browse the LDAP and confirm this with the LDAP administrator.
- **ldap.synchronization.userFirstNameAttributeName=givenName**
  The LDAP attribute that stores the first name of a user.
- **ldap.synchronization.userLastNameAttributeName=sn**
  The LDAP attribute that stores the last name of a user.
- **ldap.synchronization.groupIdAttributeName=extensionAttribute9**

  > When this setting is set, only LDAP groups with a non-empty value for that attribute ('extensionAttribute9' in this example) will be synchronized into Alfresco. This is an alternative way to reduce the size of LDAP groups available to Alfresco. Do **NOT** set this LDAP attribute to names identical to their Alfresco counterparts. Add a prefix such as "LDAP_".

  This setting offers a way to rename the LDAP group name when it is synchronized into Alfresco group name. For example, for LDAP group "Alfresco_site_ACME", if its 'extensionAttribute9' value is "LDAP_site_ACME", then the group name shown in Alfresco will be "LDAP_site_ACME", which from the Alfresco point of view, the name makes more sense.
- **ldap.synchronization.modifyTimestampAttributeName=whenChanged**
  The LDAP attribute that stores the timestamp of the last modification to a user or group.
- **ldap.synchronization.timestampFormat=yyyyMMddHHmmss'.0Z'**
  Browse the LDAP to verify this timestamp pattern.
- **ldap.synchronization.groupQuery=**(objectclass\=group)
  Verify this query in the LDAP browser. Remove the backslash when pasting the query into the browser.
- **ldap.synchronization.groupDifferentialQuery=**(&(objectclass\=group)(!(whenChanged<\={0})))
  Verify this query in the LDAP browser. Remove the backslashes when pasting the query into the browser. The 'whenChanged' timestamp name should match the value of modifyTimestampAttributeName.
- **ldap.synchronization.personQuery=**(&(objectclass\=user)(userAccountControl:1.2.840.113556.1.4.803:=512))
  Verify this query in the LDAP browser. Remove the backslash when pasting the query into the browser. The "userAccountControl:1.2.840.113556.1.4.803:=512" element is specific to Active Directory. Confirm that with LDAP administrator.
- **ldap.synchronization.personDifferentialQuery=**(&(objectclass\=user)(userAccountControl:1.2.840.113556.1.4.803:=512)(!(whenChanged<\={0})))
  Verify this query in the LDAP browser. Remove the backslashes when pasting the query into the browser. The 'whenChanged' timestamp name should match the value of modifyTimestampAttributeName. The "userAccountControl:1.2.840.113556.1.4.803:=512" element is specific to Active Directory. Confirm that with LDAP administrator.

Add LDAP Groups to Alfresco Groups

Once synchronization is successful, LDAP groups should be visible in Share Admin Console. Add these LDAP groups into their Alfresco counterparts. For example,

- Add "Alfresco_site_ACME_SiteManager" group into "site_ACME_SiteManager" group
- Add "Alfresco_site_ACME_SiteCollaborator" group into "site_ACME_SiteCollaborator" group
- Add "Alfresco_site_ACME_SiteContributor" group into "site_ACME_SiteContributor" group
- Add "Alfresco_site_ACME_SiteConsumer" group into "site_ACME_SiteConsumer" group.

Finally, login to Share again using a user in the LDAP group. The corresponding site should be visible in the Share dashboard. This concludes the LDAP integration.

Setting up Hot Backup

A hot backup should comprise of the following steps:

1. Backup Search Indexes (alf_data/solrBackup or alf_data/backup-lucene-indexes)
2. Backup SQL
3. Backup files (alf_data/contentstore, alf_data/contentstore.deleted)

⚠ **"Order of Operations"**
These steps should occur in this order, and should NOT overlap, otherwise the system could get in a corrupt state when restored from the backup. The backed up artifacts from step 1 through 3 should then be stored as a package, so that it can easily restored should this become necessary in the future.

Scheduled Task / Chron

By default you will set up your backup process to execute at 4AM. This is based on the fact that Alfresco backs up its search indexes every day at 3:00am.

Create / Clear your temporary backup location

Create a location where you will store the components of your backup. Inside that location create folders:

- search indexes
- metadata (SQL database)
- content (CMS repository)

Solr or Lucene Backup

On the Alfresco instance, most Crafter Studio installations are configured to use the Solr search engine, with **alf_data/solrBackup** as its index backup folder. Those configured to use the Lucene search engine, the index backup folder is **alf_data/backup-lucene-indexes**. Either folder contains daily backup created by Alfresco via scheduled job within the running application that creates backups daily at 3:00 AM.

To backup the search index, move the contents of the **alf_data/solrBackup** or **alf_data/backup-lucene-indexes** directory to your temporary index backup location.

SQL Backup

Once the search index backup is completed, dump your database (metadata) to disk and move the dump to your temporary backup location/metadata.

Content backup

Once metadata have been stored, copy the contents of your repository (**alf_data/contentstore**, **alf_data/contentstore.deleted**) to your temporary backup location/content

Compress and store

Compress the temporary location

- Initiate a backup management system to collect the new backup asset
- Or store the compressed backup in a safe location according to your retention policies.

Files Backup

Once the database export has been transferred into the backup files location, the script will then use rsync to backup the alf_data/contentstore and alf_data/contentstore.deleted directories to a temporary backup location.

In order to remotely execute the database backup script, we are using public key authentication for the oracle account. A good HOWTO for setting this up can be found here: http://paulkeck.com/ssh/. Below are the notes of the initial setup:

Frequently Asked Questions

Should I use the install script or the installers

- For demos, evaluations, and developer instances use the installers.
- For production installations use the script. This script covers aspects such as HA, backup, hardened configuration, performance tuning, and other topics which the installers do not address – and for the use cases above are generally not appropriate.

I've noticed that Crafter CMS is constructed of many WAR files. Why?

There are two basic reasons for this:

- Simplicity of the overall code base. As individual applications each component is easy to understand and maintain
- Scalability. Each component can easily be tiered out on to separate infrastructure. You can deploy Crafter CMS all on a single box or achieve cloud scale.

I used the Crafter CMS installer, it works locally but I want others to connect to my instance remotely

1. If you want to use your own domain for your authoring servers (not localhost / 127.0.0.1)
    a. Create file *tomcat/shared/classes/crafter/cstudio/extension/server-context.xml* as the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:context="http://www.springframework.org/schema/context"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd">

<bean name="author-config-properties"
class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
<property name="locations">
<list>
<value>classpath*:crafter/server-config.properties</value>
<value>classpath*:crafter/cstudio/extension/server-config.properties</value>
</list>
</property>
<property name="ignoreUnresolvablePlaceholders" value="true" />
</bean>
</beans>
```

    b. Set the following in file *tomcat/shared/classes/crafter/cstudio/extension/server-config.properties*

```
authoringDomain=.craftercms.com
```

    c. In the repository, either in your blue-prints for the site or in the site configuration itself you need to modify the environment override files: environment-config.xml

```
<environment-config>
...
<cookie-domain>craftercms.com</cookie-domain>
...
</environment-config>
```

===================

How to Install the Crafter CMS Server

From Tomcat Zip File

The tomcat zip file will set up a fully functional Crafter Studio instance running against test data stores. It can easily be configured (using the documentation below) to point to Alfresco and external RDBMS.

- Download the tomcat install zip here (TODO ON RELEASE)
    - Source code for this can be found here
- Unzip the tomcat in any directory
- Start the tomcat
    - Linux / Apple: INSTALL-DIR/bin/startup.sh
    - Windows: ToDo
- Login in with admin/admin (any username password will work, the security provider is faux by default)
- Create a site

From Installer Application

TODO

How to Run Crafter on Ports other than 8080

1. Assuming tomcat: for the server, update the ports the server runs on (Example: 9090)

    a. **TOMCAT/conf/server.xml**

    ```
     <Server port="9005" shutdown="SHUTDOWN">
    ...
    <Connector port="9090" URIEncoding="UTF-8" protocol="HTTP/1.1"
                   connectionTimeout="20000"
                   redirectPort="9443"
    ...
    <Connector port="9009" protocol="AJP/1.3" redirectPort="8443" />
        <Connector port="9443"
    protocol="org.apache.coyote.http11.Http11Protocol" SSLEnabled="true"...
    ```

2. If running Alfresco, update Alfresco endpoint configuration

    a. **TOMCAT/shared/classes/crafter/cstudio/extension/server-config.properties**

    ```
    alfrescoUrl=http://ALFRESCO_SERVER_NAME_GOES_HERE:ALFRESCO_PORT_GOES_HERE/alfresco
    ```

3. If Alfresco is running in the same tomcat, change the Alfresco ports: http://docs.alfresco.com/community/tasks/WQS-config-single-app.html
4. Update Crafter Search endpoint configuration (this URL points to the Crafter Search instance supporting preview)

    a. **TOMCAT/shared/classes/crafter/engine/extension/server-config.properties**

    ```
    crafterSeachUrl=http://localhost:9090/crafter-search
    ```

    b. **TOMCAT/shared/classes/crafter/cstudio/extension/server-config.properties**

    ```
    crafterSeachUrl=http://localhost:9090/crafter-search
    ```

5. Update Crafter Social endpoint configuration TODO
6. Update Crafter Profile endpoint configuration TODO

How to Configure the Crafter CMS Database

Derby (testing/eval only) Configured by Default

TOMCAT/shared/classes/crafter/cstudio/extension/server-config.properties

```
studio.db.name=CRAFTER
studio.db.username=crafter
studio.db.password=crafter
studio.db.driver=org.apache.derby.jdbc.EmbeddedDriver
studio.db.derby.path=data/derby/${studio.db.name}
studio.db.url=jdbc:derby:${studio.db.derby.path}
```

MySql

TOMCAT/shared/classes/crafter/cstudio/extension/server-config.properties

```
studio.db.name=crafter
studio.db.username=crafter
studio.db.password=crafter
studio.db.driver=org.gjt.mm.mysql.Driver
studio.db.url=jdbc:mysql://localhost:3306/${studio.db.name}?useUnicode=yes&characterEncoding=UTF-8
```

Oracle

todo

Postgres

todo

MS SQL Server

todo

How to Configure Crafter CMS to Use the Test Content Repository

The test reository is configured by default. To switch back to it follow the instructions below. Remember there is both disk and database components. Switching repositories means switching both of thesse components (or dropping the database.) The default/test database is Derby.

TOMCAT/shared/classes/crafter/cstudio/extension/server-config.properties

```
repositoryType=default
securityType=default
repository.diskImplementation.path=/Users/rdanner/crafter-installs/2-5-0/crafter/data/repo
```

How to Configure Crafter CMS to Use Alfresco as the Content Repository

1. Log in to Alfresco
2. Import the following ZIP file in to COMPANY HOME inside Alfresco
3. Add/Update the following properties:

   TOMCAT/shared/classes/crafter/cstudio/extension/server-config.properties

   ```
   repositoryType=alfresco
   securityType=alfresco
   alfrescoUrl=http://localhost:8080/alfresco
   repositoryJob.password=MYUSER
   repositoryJob.username=MYPASSWORD
   ```

4. Restart Crafter Studio

Note: Your Content Repository and your Crafter Studio Database need to be in sync. When you connect to Alfresco for the first time, it's recommended that you bring these in to sync or that you clear your studio database.

Changing Ports on your Preview Server Deployer

In some cases (when you want to run more than one full studio install side by side) you need to change the ports on the deployer. To do this update the following

1. Tell Studio where to look for it's deployer by setting the properties below. If you're just changing the port, set only the port property. Other properties are listed in case you would like to change them as well. Update the properties file and restart the studio application.

TOMCAT/shared/classes/crafter/cstudio/extension/server-config.properties

```
crafter.deployer.server=localhost
crafter.deployer.port=9595
crafter.deployer.target=sample
crafter.deployer.password=admin
```

1. Configure the deployer to run on the specified port:

   CRAFTER-HOME/crafter-deployer/conf/custom-receiver.properties

   ```
   receiver.port=9595
   ```

Production / Advanced Installation Topics

For topics such as:

- Clustering
- Production performance tuning
- Log rolling
- Backup
- Disaster recovery
- CDN integration
- Complex or WAN deployment topologies
- and other advanced topics
  Please refer to the enterprise installation guide found here (TODO)