CMS Container

{ Installation Guide }

[v 1.4]

CONTENTS

1 Introduction....3

2 Prerequisites....4

3 MySQL configuration....5

3.1 Changes in my.cnf....5
3.2 Creating the database(s)....5
3.3 User rights for the database....6
3.4 Testing MySQL....6

4 Tomcat installation....7

4.1 Creating a Linux account for Tomcat....7
4.2 Installation of Tomcat....7
4.3 Adding third party libraries needed by Tomcat....7
4.4 Starting and stopping Tomcat....7
4.5 Testing tomcat....7

5 Installation of the CMS Container application....8

5.1 Import databases (optional)....8
5.2 Starting the CMS Container for the first time....8
5.3 Modifying the context files....9
5.4 Testing the CMS Container application via Tomcat....10
5.5 Connecting Tomcat to Apache HTTP Server (Optional)....10
5.5.1 mod_proxy....10
5.5.2 mod_jk....10
5.6 Testing the CMS Container application using Apache....11
5.7 Log files....11

Appendix A: Resources....12

Appendix B: Init.d script for Tomcat (Linux)....13

Appendix C: Example context file....14

1 INTRODUCTION

This manual describes the installation of the CMS Container on Linux and all third party software needed by it. The manual is intended for system administrators with a basic understanding of Linux, MySQL and Apache web server.

Author, Date	A. van der Elst, April 19th, 2007
Last update by	J. Stroomer, Jan 20th, 2008
Version	v1.4
Status	Tested with CMSC 1.5.0

- **Read me first!**
  The CMS Container can be used in two modes, depending on your situation:
  - A **standalone** mode, where changes are published immediately on the site. This mode uses only one WAR-file, such as `cmsc-webapp-1.3.2.war`. This document is based on the preview-live mode, and where two configuration files are needed, you only need one!
  - A **preview-live** mode, with both a Staging and a Live web application. At the staging web application, the editor section updates the content and previews their work. After the content is verified and published, it becomes visible at the live web application. Useful for large projects with multiple persons. This mode uses two WAR-files, such as `cmsc-staging.war` and `cmsc-live.war`.

2 PREREQUISITES

The following software has to be installed prior to the installation of the CMS Container:

- Apache Tomcat 5.5.x
- MySQL version 5.0.x
- MySQL Connector/J 5.1 or higher
- Java 5.0.x
- ImageMagick 6.3.x (or higher)
- Third party libraries from Sun and the Apache Software Foundation

Optional

- Apache HTTP Server 2.x with mod_so and mod_proxy modules
- Apache Tomcat Connector JK-1.2.26 or higher
- A mail server

See the list of resources at the end of this document for the locations where this software can be acquired. Some of this software might already be included in your Operating System.

This manual describes the specific steps to install and configure the CMS Container software. For more information please consult the manuals included with the software.

3 MYSQL CONFIGURATION

Install MySQL according to the general MySQL installation manual. The CMS Container has been tested with the MySQL 5.0.x database. See http://dev.mysql.com/doc/refman/5.0/en/ for more information.

3.1 Changes in my.cnf

The following settings in my.cnf (or my.ini in Windows) have to be verified & changed if necessary when using a default MySQL 5.x installation:

```
[mysql]
max_allowed_packet = 16M


[mysqld]
default-character-set=utf8
port            = 3306
max_allowed_packet = 16M

innodb_data_home_dir = /srv/mysql/var/
innodb_data_file_path = ibdata1:64M:autoextend
innodb_log_group_home_dir = /srv/mysql/var/
innodb_log_arch_dir = /srv/mysql/var/
# You can set .._buffer_pool_size up to 50 - 80 %
# of RAM but beware of setting memory usage too high
innodb_buffer_pool_size = 128M
innodb_additional_mem_pool_size = 16M
# Set .._log_file_size to 25 % of buffer pool size
innodb_log_file_size = 32M
innodb_log_buffer_size = 32M
#innodb_flush_log_at_trx_commit = 1
#innodb_lock_wait_timeout = 50

[mysqldump]
max_allowed_packet = 16M
```

Notes about the innodb settings:

- The autoextend value of the innodb_data_file should be at least twice the value of max_allowed_packet, all other values depend on the amount of available memory and could impact database performance.
- The innodb data & logfiles are created by MySQL on first startup, when changing settings they have to be recreated, if you currently have no databases using InnoDB tabletypes you can remove the ibdata1, ib_logfile0 & ib_logfile1 file, if you currently have databases using InnoDB tabletypes, you have to backup & reimport these databases after changing the InnoDB settings.

3.2 Creating the database(s)

Create the database for the CMS Container, the CMS Container is using "UTF-8" as it's character set:

Note: The default CMSc configuration assumes that you use one database for both the staging and live environment. If you prefer two seperate databases, please make sure you adjust the context XML files accordingly (see paragraph 5.3).

```
CREATE DATABASE cmsc DEFAULT CHARACTER SET 'utf8';
```

where *cmsc* is the name of the database. The name of the database is usually changed to something

that is a little closer to the CMS Container project name. This information is usually provided (outside of this general document) by the team lead developer that worked on the project.

3.3 User rights for the database

Grant user rights to the database for the account used by the CMS Container:

```
GRANT ALL PRIVILEGES ON cmsc.* TO 'cmsc'@'localhost' IDENTIFIED BY 'cmsc'
WITH GRANT OPTION;
```

where *cmsc* is the name of the database, the CMS Container account and its password.

Both username and password can be changed to something more secure, however you will have to edit the context files (see paragraph 5.3) to match the changes. Specifically the *username* and *password* attribute of the <Resource /> with the name *jdbc/cmsc*.

3.4 Testing MySQL

Verify that you can connect to the MySQL database with the user name and password by entering the following command in a console:

```
mysql -u username -p databasename
```

You should connect successfully to the created database.

4 TOMCAT INSTALLATION

4.1 Creating a Linux account for Tomcat

On Linux, create a group and user account for Tomcat with the following commands:

```
groupadd tomcat
useradd -d <installation-directory>/tomcat -g tomcat -m tomcat
```

where installation-directory is the directory where you want to install Tomcat. Adding an account is not necessary in Windows.

4.2 Installation of Tomcat

Install Tomcat according to the general Tomcat installation manual. See http://tomcat.apache.org/tomcat-5.5-doc/index.html for more information. The installation directory of Tomcat is referred to as $CATALINA_HOME.

4.3 Adding third party libraries needed by Tomcat

A number of third party libraries used for the database connection are not included in the CMS Container distribution and have to be installed in the common library directory of Tomcat. Newer versions of the packages likely work too.

($CATALINA_HOME/common/lib):

- mysql-connector-java-5.1.x-bin.jar (MySQL® Connector/J)
- mail.jar (JavaMail 1.4.1)
- activation.jar (JavaBeans Activation Framework (JAF) 1.1.1)
- commons-dbcp-1.2.2.jar (Database connection pooling services.)
- commons-pool-1.4.jar (Generic object pooling component.)
- commons-collections-3.2.1.jar (Java Collection Framework extension)

See the list of resources for the locations where this software can be acquired. They can also be downloaded from the downloads page on the cmscontainer.org site.

4.4 Starting and stopping Tomcat

In linux

Use an inet.d script to start and stop Tomcat, see Appendix B for an example.

```
/etc/init.d/tomcat start
/etc/init.d/tomcat stop
```

In Windows

In the installation directory of Tomcat: `.\bin\startup.bat`
To stop Tomcat, press Ctrl+Break in the Tomcat console window.

4.5 Testing tomcat

Verify the Tomcat installation with a web browser by browsing to http://localhost:8080/. You should see the general Tomcat information page. Please stop the running Tomcat after you verified it works.

5 INSTALLATION OF THE CMS CONTAINER APPLICATION

The CMS Container application is provided as a standard web application archive (WAR).

5.1 Import databases (optional)

If you are a customer and got a database dump, you can import the database, before deploying the application:

```
mysql -u username -p databasename < dump.sql
```

Now is also a good time to check the MySQL error log any errors/warnings, this log file is usually located in the mysql/var directory. When an empty database is used, the CMS Container application will create the necessary database tables by itself.

5.2 Starting the CMS Container for the first time

1) If you have received WAR files as a customer, you can use these in this section. Otherwise, you can download the latest release of CMS Container at this website: http://www.cmscontainer.org/download

2) Copy the WAR file(s) into the deploy directory of Tomcat, in Linux as follows:

```
cp cmsc-live.war $CATALINA_HOME/webapps
cp cmsc-staging.war $CATALINA_HOME/webapps
```

In Windows, you can use the Explorer.

3) Start Tomcat as shown at section 4.4 Starting and Stopping Tomcat.

The CMS Container application will create a context file for each WAR file you put in the /webapps directory, in the Tomcat root directory.
The configuration file(s) are located in `$CATALINA_HOME/conf/Catalina/localhost` with the name of the web applications:

`cmsc-live.xml` **and** `cmsc-staging.xml`

These context files contain important settings for the database, mail server, ImageMagick etc. and should be modified, as described in the following section.

Two WAR-files situation: If the start up of Tomcat fails, you might get only one configuration file (.xml file). Correct the existing xml-file and when Tomcat successfully deploys one War-file after a restart, it will create the other xml-configuration file. You will likely have to correct both files.

ROOT WAR-file deployment: Oftentimes the live WAR file (`cmsc-live.xml`) is deployed in the ROOT of Tomcat, in this case you can unzip the live WAR file to `$CATALINA_HOME/webapps/ROOT/` instead of copying the WAR file to `$CATALINA_HOME/webapps/`.

5.3 Modifying the context files

Check and modify the database connection settings in the context file(s), in particular the user name, password and the host, port and database name in the url:

```
$CATALINA_HOME/conf/Catalina/localhost/cmsc-live.xml
```

Where `cmsc-live` is the name of the web application (this name depends on the WAR file name). A complete XML file can be found in Appendix C.

```
<Resource name="jdbc/cmsc"
... other attributes
      username="cmsc"
      password="cmsc"
      url="jdbc:mysql://localhost:3306/cmsc?
useUnicode=true&amp;characterEncoding=UTF-8&amp;autoReconnect=true&amp;useSe
rverPrepStmts=false" />
```

Check and modify the ImageMagick settings, in particular the `convert` command and its location. The following code block is a Linux example. In Windows, the location can be for example: `'C:\bin\ImageMagick-6.3.5-Q16'`.

```
<Environment name="mmbase/imaging/ImageConvert.ConverterCommand"
value="convert" type="java.lang.String" />
<Environment name="mmbase/imaging/ImageConvert.ConverterRoot"
value="/usr/local/ImageMagick-6.3.2/bin" type="java.lang.String" />
```

Tip! If you're unsure of the location of ImageMagick, in Linux, type 'which convert' on the command line to see if it's in the system path.

If you have chosen to deploy the live WAR file in the ROOT of tomcat (see paragraph 5.2) make sure the `server/useServerName` property is set to `true` in the live context file.

```
<Environment name="server/useServerName" value="false"
type="java.lang.String" />
```

Optionally, add a SMTP mail server by providing the `mail.smtp.host` attribute with the host name of your mail server:

```
<Resource name="mail/Session"
      ... other attributes
      mail.smtp.host="smtp.mydomain.com" />
```

See http://tomcat.apache.org/tomcat-5.5-doc/config/context.html for more information about context files.

Note: If you are using two WAR files, make sure you modify both the live & staging context files!

Note: Changes in the context file(s) will become active after a restart of Tomcat.

Tip: when searching for the cause of Tomcat errors or start up failures of Tomcat, take a look at section 5.7 of this document: 'Log files'

5.4 Testing the CMS Container application via Tomcat

Verify the CMS Container installation, connecting directly to Tomcat, with a web browser by browsing to http://localhost:8080/cmsc-staging/editors. Port 8080 is the default used by Tomcat. You should see the login page of the administration module.

Add the following line to your System Environment to let Tomcat use the proper file encoding and MaxPermSize, and restart Tomcat after the change in order to use it.

```
JAVA_OPTS=-Dfile.encoding=utf-8 -XX:MaxPermSize=128m
```

5.5 Connecting Tomcat to Apache HTTP Server (Optional)

Apache HTTP Server is usually used as a web server front-end for Tomcat, only the dynamic content is handled by Tomcat. This manual assumes that you have a working Apache HTTP server installed.

There are two ways to forward requests from Apache to Tomcat:

- mod_proxy
- mod_jk

5.5.1 mod_proxy

1) `mod_proxy` is an Apache module which is often installed, but turned off by default. There are two options to turn it on, depending on your Apache configuration:

1. If you have the Dynamic Shared Object configuration, as written in `httpd.conf`: You need to uncomment the following lines:

```
LoadModule proxy_module modules/mod_proxy.so
....
LoadModule proxy_http_module modules/mod_proxy_http.so
```

2. If the modules are loaded through dynamic links, as located in `apache2/mods-enabled/`, in linux, make 3 dynamic links in `/etc/apache2/mods-enabled/` to the following three files: `../mods-available/{proxy.conf,proxy.load,proxy_http.load}`

2) To forward requests to Tomcat the following lines have to be added to the Apache configuration file (httpd.conf)

```
<Proxy *>
      Order deny,allow
      Allow from all
</Proxy>

ProxyPass /cmsc-live/ http://localhost:8080/cmsc-live/
ProxyPassReverse /cmsc-live/ http://localhost:8080/cmsc-live/
ProxyPass /cmsc-staging/ http://localhost:8080/cmsc-staging/
ProxyPassReverse /cmsc-staging/ http://localhost:8080/cmsc-staging/
ProxyPreserveHost On
```

5.5.2 mod_jk

The Apache Tomcat Connector or mod_jk can also be used to forward requests to Tomcat. See http://tomcat.apache.org/connectors-doc/ for more information.

5.6 Testing the CMS Container application using Apache

Verify the CMS Container installation, connecting via Apache HTTP Server, with a web browser by browsing to http://localhost/cmsc-staging/editors. You should see the login page of the administration module. At http://localhost/cmsc-staging/ will see an empty page if you installed an empty CMS Container project, because no site is configured.

5.7 Log files

As every good administrator knows, log-files are the essential part of installation. The log-files of the CMS Container can by default be found in the $CATALINA_HOME/logs. To change the location or names of the log-files see the file /cmsc-webapp/WEB-INF/config/log/log4j.xml

APPENDIX A: RESOURCES

Java J2SE 5.0.x: http://java.sun.com/j2se/1.5.0/

JavaMail 1.4.x: http://java.sun.com/products/javamail/downloads/index.html
JAF: http://java.sun.com/javase/technologies/desktop/javabeans/jaf/downloads/index.html

Apache HTTP Server: http://httpd.apache.org/

Apache Tomcat 5.5: http://tomcat.apache.org/download-55.cgi
Apache Tomcat Connector 1.2 or higher: http://tomcat.apache.org/connectors-doc/

MySQL Community Edition 5.0.x: http://dev.mysql.com/downloads/mysql/5.0.html
MySQL Connector/J 5.1 or higher: http://dev.mysql.com/downloads/connector/j/5.1.html

ImageMagick: http://www.imagemagick.org/

commons-collections 3.2.1 or higher: http://commons.apache.org/downloads/download_collections.cgi
commons-dbcp 1.2.2 or higher: http://commons.apache.org/downloads/download_dbcp.cgi
commons-pool 1.3 or higher: http://commons.apache.org/pool/download_pool.cgi

APPENDIX B: INIT.D SCRIPT FOR TOMCAT (LINUX)

```
#!/bin/bash
#
### BEGIN INIT INFO
# Provides:     tomcat
# Required-Start: mysql
# X-UnitedLinux-Should-Start:
# Required-Stop:
# X-UnitedLinux-Should-Stop:
# Default-Start:  2 3
# Default-Stop:   0 1 5 6
# Description:    cmsc site
### END INIT INFO
#
# description: start stops tomcat
#

# Start/stop/reset tomcat
case "$1" in

'start')
#
#
su - tomcat -c "cd ~/apache-tomcat-5.5.20/bin;./startup.sh"
   ;;
'stop')
pkill -u tomcat
sleep 10
pkill -9 -u tomcat
rm -Rf /srv/tomcat/apache-tomcat-5.5.20/work/*

   ;;
*)
   echo "Usage: /etc/init.d/tomcat { start | stop }"
   ;;
esac
```

APPENDIX C: EXAMPLE CONTEXT FILE

```
<!--    Context configuration file for the Tomcat CMS Container Web App -->
<Context debug="5" reloadable="true">

        <Logger className="org.apache.catalina.logger.FileLogger" prefix="localhost_cmsc_log."
suffix=".txt"
           timestamp="true" />

        <!--
           Remove this if you want to use the default StandarManager. The standard
           manager try to save the the session when you restart the tomcat it may cause
           problem as not all of your session attribute is not serializable
        -->
        <Manager className="org.apache.catalina.session.PersistentManager" saveOnRestart="false" />

        <!--
           mail.smtp.host             The SMTP server to connect to.
           mail.smtp.port             The SMTP server port to connect to, if the connect() method
doesn't explicitly specify one. Defaults to 25.
           mail.smtp.localhost     Local host name used in the SMTP HELO or EHLO command.
           Defaults to InetAddress.getLocalHost().getHostName().
           Should not normally need to be set if your JDK and your
           name service are configured properly.
           mail.debug                         The initial debug mode. Default is false.
        -->
        <Resource name="mail/Session" auth="Container" type="javax.mail.Session"
           factory="org.apache.naming.factory.MailSessionFactory"
           mail.smtp.host="192.168.0.44" />

        <Resource name="jdbc/cmsc" auth="Container" type="javax.sql.DataSource"
           removeAbandoned="true"
           removeAbandonedTimeout="60"
           logAbandoned="true"
           maxActive="10"
           maxIdle="1"
           maxWait="10000"
           username="cmsc"
           password="cmsc"
           driverClassName="com.mysql.jdbc.Driver"
           url="jdbc:mysql://localhost:3306/cmsc_staging?
useUnicode=true&amp;characterEncoding=UTF-8&amp;autoReconnect=true&amp;useServerPrepStmts=false" />

        <Environment name="mmbase/mmbaseroot/datasource-context" value="java:comp/env"
type="java.lang.String" />
        <Environment name="mmbase/mmbaseroot/datasource" value="jdbc/cmsc" type="java.lang.String" /
>
   <!--<Environment name="mmbase/mmbaseroot/basename" value="staging" type="java.lang.String" />-->
        <Environment name="mmbase/imaging/ImageConvertClass"
           value="org.mmbase.module.builders.ConvertImageMagick" type="java.lang.String" />
        <Environment name="mmbase/imaging/ImageConvert.ConverterCommand" value="convert"
type="java.lang.String" />
        <Environment name="mmbase/imaging/ImageConvert.ConverterRoot" value=""
type="java.lang.String" />

        <Environment name="mmbase/rmmci/port" value="2111" type="java.lang.String" />
        <Environment name="mmbase/rmmci/stubport" value="2112" type="java.lang.String" />
        <Environment name="mmbase/rmmci/bindname" value="staging" type="java.lang.String" />
        <!-- if RMIRegistryServer is not set RemoteMMCI looks for a RMIRegistryServer
           on the host defined in the mmbaseroot.xml module. -->
        <!-- <Environment name="mmbase/rmmci/RMIRegistryServer" value="localhost"
type="java.lang.String" /> -->


        <Environment name="mmbase/remotepublishing/remotecloud" value="live.server"
type="java.lang.String" />
        <!-- Contains value 'live' or 'staging' to determine whether this server is Live or Staging.
-->
        <Environment name="server/LiveOrStaging" value="staging" type="java.lang.String" />
        <!-- This entry defines whether this server includes the servername to lookup portal pages.
-->
        <Environment name="server/useServerName" value="false" type="java.lang.String" />

   <Environment name="cmsc/security/jcifs.http.domainController" value="192.168.0.3"
type="java.lang.String" />
</Context>
```