

DJIGZO EMAIL ENCRYPTION

---

# Djigzo Gateway Installation Guide

---



*Author:* Martijn BRINKERS

March 29, 2010, Rev: 4097

---

Copyright © 2008-2010, Martijn Brinkers.

**Acknowledgments:** I would like to thank Andreas Hödle for feedback and input on gateway security and Christine Karman for input, feedback and proof-reading.

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Install Djigzo on Ubuntu</b>	<b>3</b>
2.1	Configure PostgreSQL . . . . .	4
2.2	Install Djigzo . . . . .	4
2.3	Update sudoers . . . . .	5
2.4	Configure Postfix . . . . .	6
2.5	Install Tomcat . . . . .	8
2.6	Finalize . . . . .	9
<b>3</b>	<b>Install Djigzo on Red Hat/CentOS</b>	<b>11</b>
3.1	Configure PostgreSQL . . . . .	11
3.2	Install Djigzo . . . . .	12
3.3	Update sudoers . . . . .	13
3.4	Configure Postfix . . . . .	14
3.5	Install Tomcat . . . . .	17
3.6	Finalize . . . . .	19
<b>A</b>	<b>Configure Tomcat on Debian 5</b>	<b>21</b>
<b>B</b>	<b>Using Jetty 6</b>	<b>21</b>
<b>C</b>	<b>Adding Tomcat HTTPS connector</b>	<b>23</b>
<b>D</b>	<b>Memory usage</b>	<b>24</b>
<b>E</b>	<b>Securing the gateway</b>	<b>25</b>
E.1	Port usage . . . . .	25
E.2	Passwords . . . . .	25
E.3	SSL certificate . . . . .	26
E.4	Prevent spoofing the From header . . . . .	26
E.5	Securing the database . . . . .	26
E.6	Block access to pages . . . . .	26

## 1 Introduction

This install guide will give you step-by-step instructions on how to install Djigzo. If you are installing Djigzo on Ubuntu, Red Hat or CentOS you are advised to use the quick install guide which explains how to install Djigzo using the .deb or .rpm packages. If you do not want to use the .deb or .rpm packages or, if you need to install Djigzo on another system than Ubuntu, Red Hat or CentOS you must use this guide. Even though this guide assumes that Djigzo is installed on Ubuntu, Red Hat or CentOS, installation on other system will be similar and require in most cases only minor changes. You are recommended to install Djigzo on a dedicated and clean machine.

### Requirements

- Ubuntu, Red Hat or CentOS
- PostgreSQL
- Postfix
- OpenJDK 6
- ANT, ANT-optional
- Tomcat (or Jetty)

**Note:** commands that should be executed by the user are shown on lines starting with a \$ sign (the \$ sign is not part of the command to execute). You can copy and paste the commands to the command line.

**WARNING** do not install Djigzo on a live email system!

## 2 Install Djigzo on Ubuntu

This section explains how to install Djigzo on Ubuntu 8.04.

### Install required packages

```
$ sudo apt-get install postgresql postfix openjdk-6-jre \  
openjdk-6-jre-headless tzdata-java ant ant-optional \  
mktemp wget libsasl2-modules
```

**Note:** during the installation of postfix you need to choose a configuration. Select “No Configuration”.

**Make OpenJDK the default** If there are multiple Java runtimes installed we must be sure that OpenJDK is the default JRE.

```
$ sudo update-java-alternatives -s java-6-openjdk
```

**Check default Java version** Before continuing make sure that Java is properly installed. The following command should report that the default Java version is OpenJDK.

```
$ java -version
```

the output should look similar to:

```
java version "1.6.0_0"  
OpenJDK Runtime Environment (build 1.6.0_0-b11)  
OpenJDK Client VM (build 1.6.0_0-b11, mixed mode, sharing)
```

## 2.1 Configure PostgreSQL<sup>1</sup>

Djigzo stores all settings in a PostgreSQL database.

**Create database user** Create the database user *djigzo* with password *djigzo*<sup>2</sup>.

```
$ echo "CREATE USER djigzo NOCREATEUSER NOCREATEDB ENCRYPTED PASSWORD \  
'md5b720bc9de4ca53d53a4059882a0868b9';" | sudo -u postgres psql
```

**Create database** Create the database *djigzo* owned by database user *djigzo*.

```
$ sudo -u postgres createdb --owner djigzo djigzo
```

## 2.2 Install Djigzo

User and group *djigzo* with home dir `/usr/local/djigzo` must be created. Djigzo will be installed in the *djigzo* home dir and Djigzo will be running as user *djigzo*.

```
$ sudo adduser --system --group --home /usr/local/djigzo \  
--disabled-password --shell /bin/false djigzo
```

Add user *djigzo* to the *adm* group to allow user *djigzo* to read the Postfix log files<sup>3</sup>.

```
$ sudo usermod -a -G adm djigzo
```

Create a directory for Djigzo-web which will be owned by *djigzo*.

```
$ sudo mkdir /usr/local/djigzo-web  
$ sudo chown djigzo:djigzo /usr/local/djigzo-web
```

**Download Djigzo** A full installation of Djigzo requires the encryption engine and the Web admin. Both can be downloaded from <http://www.djigzo.com>. You need to download the following two files<sup>4</sup>:

```
djigzo_1.3.2-1.tar.gz  
djigzo-web_1.3.2-1.tar.gz
```

<sup>1</sup>Djigzo should work with all databases supported by Hibernate. Installation instructions for different databases however is beyond the scope of this manual.

<sup>2</sup>The encoded password is equal to 'md5' concatenated with the MD5 hash of the username and password.

<sup>3</sup>Only required if Djigzo Web admin log page should be able to show Postfix log file content.

<sup>4</sup>The exact version will be different when a new version is released.

**Untar the files**

```
$ sudo -u djigzo tar xzf djigzo_*.tar.gz --directory \  
/usr/local/djigzo/
```

```
$ sudo -u djigzo tar xzf djigzo-web_*.tar.gz --directory \  
/usr/local/djigzo-web/
```

**Run post install script** Some initialization will be done with an ANT script.

```
$ cd /usr/local/djigzo  
$ sudo -u djigzo ant
```

**Importing the database schema** The database schema must be imported into PostgreSQL.

```
$ sudo -u djigzo psql djigzo < /usr/local/djigzo/conf/djigzo.sql
```

**Start Djigzo** Manually start Djigzo to make sure that it is correctly installed.

```
$ sudo -u djigzo ./start-djigzo.sh
```

Starting Djigzo will result in a large number of output lines. The final lines should look similar to:

```
....  
SMTP Service started plain:10025//127.0.0.1  
FetchMail Disabled
```

To continue with the installation kill Djigzo by pressing CTRL+C.

**Update location of Djigzo** Djigzo must be automatically started at system startup. The startup script must know the path where Djigzo is installed.

```
$ sudo bash -c 'echo "DJIGZO_HOME=/usr/local/djigzo" >> \  
/etc/default/djigzo'
```

**Add startup to init.d** A softlink to the startup script will be added to /etc/init.d. The script will be added to the /etc/rc?.d directories to make Djigzo automatically start at reboot.

```
$ sudo ln -s /usr/local/djigzo/scripts/djigzo /etc/init.d/  
$ sudo update-rc.d djigzo defaults
```

## 2.3 Update sudoers

For some of its functionality, for example managing the Postfix mail queues, Djigzo should be allowed to start some specialized script files for which root access is required. To allow Djigzo to start these script files, the script files should be added to the `sudoers` file.

**Note:** Djigzo will function even when these script files are not added to the `sudoers` file. You will however lose the following functionality from the Web admin: configure Postfix, manage Postfix queues, backup and restore, restart and installing the JCE policy file.

**Make root owner of scripts** The script files will be run as root so they should be owned by root and not writable by others.

```
$ sudo chown root:root /usr/local/djigzo/scripts/*
```

**Edit sudoers** The following lines should be added to the `sudoers` file:

---

```
User_Alias DJIGZO_USERS = djigzo
Cmnd_Alias DJIGZO_COMMANDS = \
    /usr/local/djigzo/scripts/docopy-postfix-main-config.sh, \
    /usr/local/djigzo/scripts/dosmtp-client-passwd-config.sh, \
    /usr/local/djigzo/scripts/docopy-jce-policy.sh, \
    /usr/local/djigzo/scripts/dopostfix.sh, \
    /usr/local/djigzo/scripts/dobackup.sh, \
    /usr/local/djigzo/scripts/dorestart.sh
DJIGZO_USERS ALL=(ALL) NOPASSWD: DJIGZO_COMMANDS
```

---

The `sudoers` file should be edited with `visudo`.

```
$ sudo visudo
```

## 2.4 Configure Postfix

Djigzo uses Postfix for sending and receiving of email (MTA)<sup>5</sup>. Djigzo functions as a Postfix “after queue filter”. Postfix needs to be configured in such a way that incoming email is sent to Djigzo for encryption and decryption. In `/usr/local/djigzo/conf/system` you will find two Postfix configuration files (`main.cf` and `master.cf`) ready to be used with Postfix. We recommend that you use these Postfix configuration files. If Postfix is already configured and the settings should not be overridden you should manually merge the most important settings of the provided configuration files with your own Postfix configuration files. We will now highlight some of the most important Postfix config settings from `main.cf` and `master.cf` required by Djigzo.

**main.cf configuration** Postfix main config file should contain at minimal the `content_filter` setting which tells Postfix that all email should be filtered by Djigzo. The `content_filter` setting tells Postfix that the service running on `127.0.0.1:10025` will function as an “after queue filter”<sup>6</sup>.

<sup>5</sup>It is possible to use another MTA instead of Postfix, like for example Exim, but that’s beyond the scope of this manual.

<sup>6</sup>If you already configured a `content_filter` you should configure additional filters in `master.cf`. This will however not be explained in this guide.

```
content_filter = djigzo:127.0.0.1:10025
```

Most other settings in the main.cf provided by Djigzo are only required when you want to configure Postfix using the Djigzo web admin. Settings starting with **djigzo\_** will be replaced when the changes are applied. The **djigzo\_...** settings are used by main.cf and master.cf (the settings are referenced as `${djigzo_...}`).

**master.cf configuration** Postfix master config requires at least the following lines:

---

```

djigzo unix          -      -      n      -      4      smtp
  -o smtp_send_xforward_command=yes
  -o disable_dns_lookups=yes
  -o smtp_generic_maps=

127.0.0.1:10026 inet n      -      n      -      10     smtpd
  -o content_filter=
  -o receive_override_options=no_unknown_recipient_checks,
    no_header_body_checks,no_milters
  -o smtpd_helo_restrictions=
  -o smtpd_client_restrictions=
  -o smtpd_sender_restrictions=
  -o smtpd_recipient_restrictions=permit_mynetworks,reject
  -o mynetworks=127.0.0.0/8
  -o smtpd_authorized_xforward_hosts=127.0.0.0/8
  -o smtpd_authorized_xclient_hosts=127.0.0.0/8

```

---

Because Djigzo functions as an “after queue filter” it can happen that a message is increased in size after filtering (for example the message will be larger after signing). The *after queue message size limit* should therefore be larger than the message size limit before filtering (otherwise Postfix will reject the message after filtering). To make sure that the before filter size limit is lower than the after filter size limit, a limit must be set on the *smtpd* service.

```

smtp inet           n      -      -      -      -      smtpd
  -o message_size_limit=${djigzo_before_filter_message_size_limit}

```

**Copy postfix config files** Djigzo comes with a modified Postfix main and master configuration file containing all the required settings. You are advised to use these configuration files by copying them to the Postfix config directory.

**WARNING! THIS WILL OVERWRITE ALL SETTINGS IN THE ORIGINAL POSTFIX CONFIG FILES SO ONLY DO THIS IF THE ORIGINAL SETTINGS MAY BE OVERWRITTEN. IF EXISTING POSTFIX SETTINGS MUST BE KEPT YOU SHOULD MERGE THE REQUIRED CHANGES MANUALLY.**

```

$ sudo cp /usr/local/djigzo/conf/system/main.cf /etc/postfix/main.cf
$ sudo cp /usr/local/djigzo/conf/system/master.cf /etc/postfix/master.cf

```

**Update aliases** Postfix uses `/etc/aliases` as the alias file. Make sure that the alias file is available and up-to-date.

```
$ sudo newaliases
```

**Restart postfix**

```
$ sudo /etc/init.d/postfix restart
```

## 2.5 Install Tomcat<sup>7</sup>

Install the required Tomcat package

```
$ sudo apt-get install tomcat5.5
```

**Note for Debian users:** Tomcat fails on Debian 5 because a suitable JDK cannot be found. See Appendix A for instructions on how to set the JDK path.

**Set `djigzo-web.home`** The system property `djigzo-web.home` must reference the location where Djigzo web application is stored. The property will be added to the Tomcat default config file.

```
$ sudo bash -c 'echo "JAVA_OPTS=\"\${JAVA_OPTS} -Ddjigzo-web.home=\n/usr/local/djigzo-web\"" >> /etc/default/tomcat5.5'
```

**Configure Tomcat memory usage** In order to allow the import of very large certificate files (.p7b or .pfx files with more than 10's of thousands certificates) Djigzo requires that Tomcat is set to at least 256 MB of heap space.

```
$ sudo bash -c 'echo "JAVA_OPTS=\"\${JAVA_OPTS} \n-Djava.awt.headless=true -Xmx256M\"" >> /etc/default/tomcat5.5'
```

**Disable Java security manager** Djigzo currently does not function properly when the Tomcat Java security manager is enabled. The Tomcat Java security manager should therefore be disabled.

```
$ sudo bash -c 'echo "TOMCAT5_SECURITY=no" >> /etc/default/tomcat5.5'
```

**Allow reading and writing of SSL certificate** If you want to allow the upload of new SSL certificates using the Djigzo web admin SSL manager, Tomcat should be allowed to read and write the SSL certificate.

```
$ sudo chown tomcat55:djigzo /usr/local/djigzo-web/ssl/sslCertificate.p12
```

---

<sup>7</sup>if you would like to use Jetty instead of Tomcat skip the installation of Tomcat. See Appendix B for instructions on installing Jetty.

**Adding a HTTPS connector** A HTTPS connector must be added to the Tomcat server configuration. If the Tomcat installation is only used for Djigzo, you are advised to replace the existing Tomcat configuration file (`/etc/tomcat5.5/server.xml`) with the configuration file provided by Djigzo.

```
$ sudo cp /usr/local/djigzo-web/conf/tomcat/server.xml /etc/tomcat5.5
```

The path to djigzo-web must be updated

```
$ sudo sed s#/share/djigzo-web/#/local/djigzo-web/# \
/etc/tomcat5.5/server.xml --in-place
```

**Note:** if you want to keep the existing server.xml you need to manually add the HTTPS Connector. See Appendix C for more information.

**Adding a context** A Djigzo context must now be added to Tomcat.

```
$ sudo bash -c 'echo "<Context docBase=\"/usr/local/djigzo-web/djigzo.war\
\" unpackWAR=\"false\"/>" > /etc/tomcat5.5/Catalina/localhost/djigzo.xml'
```

**Note:** if you want Djigzo to use the root context save the file to ROOT.xml instead of djigzo.xml<sup>8</sup>.

## 2.6 Finalize

Some softlinks will be created to log and configuration files. Note that these softlinks are not required. You are however advised to add the softlinks because it makes it easier when you need to change some configuration files.

```
$ sudo mkdir /etc/djigzo
$ sudo mkdir /etc/djigzo/james
$ sudo mkdir /etc/djigzo/spring
$ sudo ln -s /usr/local/djigzo/wrapper/djigzo.wrapper.conf /etc/djigzo/
$ sudo ln -s /usr/local/djigzo/conf/charset-aliases.properties /etc/djigzo/
$ sudo ln -s /usr/local/djigzo/conf/djigzo.properties /etc/djigzo/
$ sudo ln -s /usr/local/djigzo/conf/log4j.properties /etc/djigzo/
$ sudo ln -s /usr/local/djigzo/conf/hibernate.cfg.xml /etc/djigzo/
$ sudo ln -s /usr/local/djigzo/conf/spring/djigzo.xml /etc/djigzo/spring/
$ sudo ln -s /usr/local/djigzo/conf/spring/general.xml /etc/djigzo/spring/
$ sudo ln -s /usr/local/djigzo/conf/spring/services.xml /etc/djigzo/spring/
$ sudo ln -s /usr/local/djigzo/conf/spring/soap.xml /etc/djigzo/spring/
$ sudo ln -s /usr/local/djigzo/logs/james.wrapper.log /var/log/djigzo.log

$ sudo ln -s /usr/local/djigzo/conf/james/SAR-INF/config.xml \
/etc/djigzo/james

$ sudo ln -s /usr/local/djigzo/conf/james/SAR-INF/\
custom_processors_config.xml /etc/djigzo/james
```

<sup>8</sup>the root context allows you to access Djigzo using a URL of the form `https://192.168.178.2/` instead of `https://192.168.178.2/djigzo`

```
$ sudo ln -s /usr/local/djigzo/conf/james/SAR-INF/\
internal_remote_delivery_processor.xml /etc/djigzo/james
```

```
$ sudo ln -s /usr/local/djigzo/conf/james/SAR-INF/\
smtp_server_config.xml /etc/djigzo/james
```

```
$ sudo ln -s /usr/local/djigzo/conf/james/SAR-INF/\
smtp_transport_config.xml /etc/djigzo/james
```

**Protect files** Some files containing passwords should only be readable by user *djigzo*.

```
$ sudo chmod 640 /usr/local/djigzo/conf/djigzo.properties
$ sudo chmod 640 /usr/local/djigzo/conf/hibernate.cfg.xml
```

**Restart services** Restart Postfix, Djigzo and Tomcat.

```
$ sudo /etc/init.d/postfix restart
$ sudo /etc/init.d/djigzo restart
$ sudo /etc/init.d/tomcat5.5 restart
```

**Open the Web Admin page** Djigzo should now be running (wait some time for Tomcat to startup). The login page can be accessed using the following URL<sup>9</sup> <https://192.168.178.2:8443/djigzo> (change the IP address accordingly)

**Note:** Djigzo comes with a default SSL certificate which is not trusted by your browser. You should therefore manually accept the HTTPS certificate.

**Login** Use the following login credentials:

```
username: admin
password: admin
```

**Note:** it can take some time to login after a restart because the web application needs to be initialized upon first login.

**Log output** In case Djigzo is not running you can check the log files for errors.

**Djigzo log**

```
$ less /var/log/djigzo.log
```

**Tomcat log**<sup>10</sup>

```
$ sudo less /var/log/tomcat5.5/catalina.*.log
```

---

<sup>9</sup>if Djigzo was installed as the root context the URL should be <https://192.168.178.2:8443>

<sup>10</sup>replace \* with the current date to view the most recent log file.

## 3 Install Djigzo on Red Hat/CentOS

This section explains how to install Djigzo on Red Hat 5.4 and CentOS 5.4. It is assumed that all commands are run as root (i.e. the user is logged in as root).

**Configure firewall** Red Hat and CentOS by default blocks access to most ports. The firewall must be configured to allow access to certain ports used by Djigzo. The following ports must be remotely accessible: *25 (SMTP)* and *8443*<sup>11</sup>. The firewall can be configured with the `system-config-securitylevel-tui` tool.

```
$ system-config-securitylevel-tui
```

**Note:** ports, like port 8443, must be postfixed with `:tcp`. For example: `8443:tcp`.

### Install required packages

```
$ yum install redhat-lsb postgresql postgresql-server postfix \
java-1.6.0-openjdk ant ant-nodeps mktemp wget system-switch-mail
```

**Make OpenJDK the default** If there are multiple Java runtime's installed we must be sure that OpenJDK is the default JRE.

```
$ /usr/sbin/alternatives --set java \
/usr/lib/jvm/jre-1.6.0-openjdk/bin/java
```

**Check default Java version** Before continuing make sure that Java is properly installed. The following command should report that the default Java version is OpenJDK.

```
$ java -version
```

the output should look similar to:

```
java version "1.6.0"
OpenJDK Runtime Environment (build 1.6.0-b09)
OpenJDK Client VM (build 1.6.0-b09, mixed mode)
```

### 3.1 Configure PostgreSQL<sup>12</sup>

Djigzo stores all settings in a PostgreSQL database.

**Make PostgreSQL autostart** PostgreSQL should be started at reboot.

```
$ /sbin/chkconfig postgresql on
```

### Start PostgreSQL

```
$ /sbin/service postgresql start
```

---

<sup>11</sup>See Appendix [E.1](#) for an overview of all ports used by Djigzo.

<sup>12</sup>Djigzo should work with all databases supported by Hibernate. Installation instructions for different databases however is beyond the scope of this manual.

**Enable password authentication** By default PostgreSQL does not allow applications to login with user name and password. PostgreSQL must be configured to allow login with user name and password. This should be done by editing the file `/var/lib/pgsql/data/pg_hba.conf`. The line containing *ident sameuser* should be commented out (or completely removed) and a line with *md5 authentication* should be added:

```
$ vi /var/lib/pgsql/data/pg_hba.conf
```

---

```
#host      all            all            127.0.0.1/32      ident sameuser
host      all            all            127.0.0.1/32      md5
```

---

**Restart PostgreSQL** PostgreSQL must be restarted for the changes to take effect.

```
$ /sbin/service postgresql restart
```

**Create database user** Create the database user *djigzo* with password *djigzo*<sup>13</sup>.

```
$ echo "CREATE USER djigzo NOCREATEUSER NOCREATEDB ENCRYPTED PASSWORD \
'md5b720bc9de4ca53d53a4059882a0868b9';" | sudo -u postgres psql
```

**Create database** Create the database *djigzo* owned by database user *djigzo*.

```
$ sudo -u postgres createdb --owner djigzo djigzo
```

## 3.2 Install Djigzo

User and group *djigzo* with home dir `/usr/local/djigzo` must be created. Djigzo will be installed in the *djigzo* home dir and Djigzo will be running as user *djigzo*.

```
$ /usr/sbin/adduser --home-dir /usr/local/djigzo -m \
--shell /sbin/nologin djigzo
```

Create a directory for Djigzo-web which will be owned by *djigzo*.

```
$ mkdir /usr/local/djigzo-web
$ chown djigzo:djigzo /usr/local/djigzo-web
```

**Download Djigzo** A full installation of Djigzo requires the encryption engine and the Web admin. Both can be downloaded from <http://www.djigzo.com>. You need to download the following two files<sup>14</sup>:

```
djigzo_1.3.2-1.tar.gz
djigzo-web_1.3.2-1.tar.gz
```

<sup>13</sup>The encoded password is equal to 'md5' concatenated with the MD5 hash of the user name and password.

<sup>14</sup>The exact version will be different when a new version is released.

**Untar the files** Untar and make the files owned by user *djigzo*.

```
$ tar xzf djigzo_*.tar.gz --directory /usr/local/djigzo/  
$ chown -R djigzo:djigzo /usr/local/djigzo
```

```
$ tar xzf djigzo-web_*.tar.gz --directory /usr/local/djigzo-web/  
$ chown -R djigzo:djigzo /usr/local/djigzo-web
```

**Run post install script** Some initialization will be done with an ANT script.

```
$ cd /usr/local/djigzo  
$ sudo -u djigzo ant
```

**allow text relocation** if SELinux is enabled we should allow text relocation for the Java wrapper lib.

```
$ chcon -t textrel_shlib_t /usr/local/djigzo/wrapper/libwrapper.so
```

**Importing the database schema** The database schema must be imported into PostgreSQL.

```
$ sudo -u djigzo psql djigzo < /usr/local/djigzo/conf/djigzo.sql
```

**Copy startup script** Red Hat/CentOS requires a modified startup script (overwrite the existing script).

```
$ cp /usr/local/djigzo/dist/redhat/djigzo /usr/local/djigzo/scripts/
```

**Update location of Djigzo** Djigzo must be automatically started at system startup. The startup script must know the path where Djigzo is installed.

```
$ echo "DJIGZO_HOME=/usr/local/djigzo" >> /etc/default/djigzo
```

**Add startup to init.d** A softlink to the startup script will be added to /etc/init.d. The script will added to the /etc/rc?.d directories to make Djigzo automatically start at reboot.

```
$ ln -s /usr/local/djigzo/scripts/djigzo /etc/init.d/  
$ /sbin/chkconfig --add djigzo
```

### 3.3 Update sudoers

For some of it's functionality, for example managing the Postfix mail queues, Djigzo should be allowed to start some specialized script files for which root access is required. To allow Djigzo to start these script files, the script files should be added to the `sudoers` file.

**Note:** Djigzo will function even when these script files are not added to the `sudoers` file. You will however loose the following functionality from the Web admin: configure Postfix, manage Postfix queues, backup and restore, restart and installing the JCE policy file.

**Make root owner of scripts** The script files will be run as root so they should be owned by root and not writable by others.

```
$ chown root:root /usr/local/djigzo/scripts/*
```

**Edit sudoers** The following lines should be added to the `sudoers` file:

---

```
User_Alias DJIGZO_USERS = djigzo
Cmnd_Alias DJIGZO_COMMANDS = \
    /usr/local/djigzo/scripts/docopy-postfix-main-config.sh,\
    /usr/local/djigzo/scripts/dosmtp-client-passwd-config.sh,\
    /usr/local/djigzo/scripts/docopy-jce-policy.sh,\
    /usr/local/djigzo/scripts/dopostfix.sh,\
    /usr/local/djigzo/scripts/dobackup.sh,\
    /usr/local/djigzo/scripts/dorestart.sh
DJIGZO_USERS ALL=(ALL) NOPASSWD: DJIGZO_COMMANDS
```

---

By default Red Hat/CentOS enable the `sudoers` setting **requiretty**. This *must* be commented out because Djigzo need to run commands without a tty (the line containing *requiretty* can be found halfway the `sudoers` file).

```
# requiretty must be commented out!
#Defaults requiretty
```

The `sudoers` file should be edited with `visudo`.

```
$ visudo
```

### 3.4 Configure Postfix

Djigzo uses Postfix for sending and receiving of email (MTA)<sup>15</sup>. Djigzo functions as a Postfix “after queue filter”. Postfix needs to be configured in such a way that incoming email is sent to Djigzo for encryption and decryption. In `/usr/local/djigzo/conf/system` and in `/usr/local/djigzo/dist/redhat/` you will find two Postfix configuration files (`main.cf` and `master.cf`) ready to be used with Postfix. We recommend that you use these Postfix configuration files. If Postfix is already configured and the settings should not be overridden you should manually merge the most important settings of the provided configuration files with your own Postfix configuration files. We will now highlight some of the most important Postfix config settings from *main.cf* and *master.cf* required by Djigzo.

---

<sup>15</sup>It is possible to use another MTA instead of Postfix, like for example Exim, but that’s beyond the scope of this manual.

**main.cf configuration** Postfix main config file should contain at minimal the **content\_filter** setting which tells Postfix that all email should be filtered by Djigzo. The `content_filter` setting tells Postfix that the service running on 127.0.0.1:10025 will function as an “after queue filter”<sup>16</sup>.

```
content_filter = djigzo:127.0.0.1:10025
```

Most other settings in the `main.cf` provided by Djigzo are only required when you want to configure Postfix using the Djigzo web admin. Settings starting with **djigzo\_** will be replaced when the changes are applied. The **djigzo\_...** settings are used by `main.cf` and `master.cf` (the settings are referenced as `#{djigzo_...}`).

**master.cf configuration** Postfix master config requires at least the following lines:

---

```

djigzo unix          -      -      n      -      4      smtp
  -o smtp_send_xforward_command=yes
  -o disable_dns_lookups=yes
  -o smtp_generic_maps=

127.0.0.1:10026 inet  n      -      n      -      10     smtpd
  -o content_filter=
  -o receive_override_options=no_unknown_recipient_checks,
    no_header_body_checks,no_milters
  -o smtpd_helo_restrictions=
  -o smtpd_client_restrictions=
  -o smtpd_sender_restrictions=
  -o smtpd_recipient_restrictions=permit_mynetworks,reject
  -o mynetworks=127.0.0.0/8
  -o smtpd_authorized_xforward_hosts=127.0.0.0/8
  -o smtpd_authorized_xclient_hosts=127.0.0.0/8

```

---

Because Djigzo functions as an “after queue filter” it can happen that a message is increased in size after filtering (for example the message will be larger after signing). The *after queue message size limit* should therefore be larger than the message size limit before filtering (otherwise Postfix will reject the message after filtering). To make sure that the before filter size limit is lower than the after filter size limit, a limit must be set on the `smtpd` service.

```

smtp inet           n      -      -      -      -      smtpd
  -o message_size_limit=#{djigzo_before_filter_message_size_limit}

```

**Copy postfix config files** Djigzo comes with a modified Postfix main and master configuration file containing all the required settings. You are advised to use these configuration files by copying them to the Postfix config directory.

<sup>16</sup>If you already configured a `content_filter` you should configure additional filters in `master.cf`. This will however not be explained in this guide.

**WARNING! THIS WILL OVERWRITE ALL SETTINGS IN THE ORIGINAL POSTFIX CONFIG FILES SO ONLY DO THIS IF THE ORIGINAL SETTINGS MAY BE OVERWRITTEN. IF EXISTING POSTFIX SETTINGS MUST BE KEPT YOU SHOULD MERGE THE REQUIRED CHANGES MANUALLY.**

```
$ cp /usr/local/djigzo/conf/system/main.cf /etc/postfix/main.cf
$ cp /usr/local/djigzo/dist/redhat/master.cf /etc/postfix/
```

**Update aliases** Postfix uses `/etc/alias` as the alias file. Make sure that the alias file is available and up-to-date.

```
$ newaliases
```

**Make Postfix the default MTA** Red Hat/CentOS installs Sendmail by default. Because Djigzo requires Postfix we must switch the default MTA from Sendmail to Postfix.

```
$ system-switch-mail
```

Optionally if Sendmail is no longer required you can remove Sendmail.

```
$ yum remove sendmail
```

**Configure SELinux** If SELinux is enabled (which is by default) Postfix is not allowed to bind to port 10026 (which is used by Djigzo as the Postfix “re-injection” port). SELinux should be configured to allow Postfix to bind to port 10026. This can be done by creating a file `djigzo.te` with the following content:

---

```
module djigzo 1.0;

require {
    type postfix_master_t;
    type port_t;
    class tcp_socket name_bind;
}

allow postfix_master_t port_t:tcp_socket name_bind;
```

---

The SELinux module must now be compiled and loaded

```
$ checkmodule -M -m -o djigzo.mod djigzo.te
$ semodule_package -o djigzo.pp -m djigzo.mod
$ semodule -i djigzo.pp
```

**Note:** alternatively you can disable SELinux with the `system-config-securitylevel-tui` tool if you have troubles getting SELinux to work with Postfix and Djigzo.

**Restart postfix** If Postfix was not yet running “Shutting down postfix” will fail. This can be ignored.

```
$ /sbin/service postfix restart
```

**Make mail logs readable** The mail logs should be readable by user *djigzo*.

```
$ chmod +r /var/log/maillog
```

**Update log paths** Djigzo by default expects the mail logs to be stored in mail.info. Red Hat/CentOS however store the mail logs in maillog. The paths in soap.xml should be updated.

```
$ sed s#/var/log/mail.info.0#/var/log/maillog.1# \
/usr/local/djigzo/conf/spring/soap.xml --in-place
```

```
$ sed s#/var/log/mail.info#/var/log/maillog# \
/usr/local/djigzo/conf/spring/soap.xml --in-place
```

### 3.5 Install Tomcat

```
$ yum install tomcat5.i386
```

**Add xalan to endorsed jars** Djigzo-web requires xalan jars in the Tomcat endorsed directory.

```
$ rebuild-jar-repository /var/lib/tomcat5/common/endorsed \
xalan-j2-2.7.0.jar
```

```
$ rebuild-jar-repository /var/lib/tomcat5/common/endorsed \
xalan-j2-serializer-2.7.0.jar
```

**Update Javamail** Red Hat/CentOS by default installs an older version of Javamail. The newer version of Javamail provided by Djigzo will be added as a new alternative.

```
$ alternatives --install /usr/share/java/javamail.jar javamail \
/usr/local/djigzo/lib/mail/mail.jar 20000
```

```
$ alternatives --install /usr/share/java/jaf.jar jaf \
/usr/local/djigzo/lib/mail/activation.jar 20000
```

**Set djigzo-web.home** The system property **djigzo-web.home** must reference the location where Djigzo web application is stored. The property will be added to the Tomcat default config file.

```
$ echo "JAVA_OPTS=\"\$JAVA_OPTS -Ddjigzo-web.home=\
/usr/local/djigzo-web\"" >> /etc/sysconfig/tomcat5
```

**Configure Tomcat memory usage** In order to allow the import of very large certificate files (.p7b or .pfx files with more than 10's of thousands certificates) Djigzo requires that Tomcat is set to at least 256 MB of heap space.

```
$ echo "JAVA_OPTS=\"\$JAVA_OPTS \
-Djava.awt.headless=true -Xmx256M\"" >> /etc/sysconfig/tomcat5
```

**Adding a HTTPS connector** A HTTPS connector must be added to the Tomcat server configuration. If the Tomcat installation is only used for Djigzo, you are advised to replace the existing Tomcat configuration file (/etc/tomcat5/server.xml) with the configuration file provided by Djigzo.

```
$ cp /usr/local/djigzo-web/conf/tomcat/server.xml /etc/tomcat5
```

The path to djigzo-web must be updated to make sure the SSL certificate is loaded from /usr/local/djigzo-web.

```
$ sed s#/share/djigzo-web/#/local/djigzo-web/# \
/etc/tomcat5/server.xml --in-place
```

**Note:** if you want to keep the existing server.xml you need to manually add the HTTPS Connector. See Appendix C for more information.

**Adding a context** A Djigzo context must now be added to Tomcat.

```
$ echo "<Context docBase=\"/usr/local/djigzo-web/djigzo.war\" unpackWAR=\
>false\"/>" > /etc/tomcat5/Catalina/localhost/djigzo.xml
```

**Note:** if you want Djigzo to use the root context save the file to ROOT.xml instead of djigzo.xml<sup>17</sup>.

**Allow reading and writing of SSL certificate** If you want to allow the upload of new SSL certificates using the Djigzo web admin SSL manager, Tomcat should be allowed to read and write the SSL certificate.

```
$ chown tomcat:djigzo /usr/local/djigzo-web/ssl/sslCertificate.p12
```

**Make Tomcat start at reboot** Tomcat should be automatically started at reboot.

```
$ /sbin/chkconfig tomcat5 on
```

<sup>17</sup>the root context allows you to access Djigzo using a URL of the form https://192.168.178.2/ instead of https://192.168.178.2/djigzo

### 3.6 Finalize

Some softlinks will be created to log and configuration files. Note that these softlinks are not required. You are however advised to add the softlinks because it makes it easier when you need to change some configuration files.

```
$ mkdir /etc/djigzo
$ mkdir /etc/djigzo/james
$ mkdir /etc/djigzo/spring
$ ln -s /usr/local/djigzo/wrapper/djigzo.wrapper.conf /etc/djigzo/
$ ln -s /usr/local/djigzo/conf/charset-aliases.properties /etc/djigzo/
$ ln -s /usr/local/djigzo/conf/djigzo.properties /etc/djigzo/
$ ln -s /usr/local/djigzo/conf/log4j.properties /etc/djigzo/
$ ln -s /usr/local/djigzo/conf/hibernate.cfg.xml /etc/djigzo/
$ ln -s /usr/local/djigzo/conf/spring/djigzo.xml /etc/djigzo/spring/
$ ln -s /usr/local/djigzo/conf/spring/general.xml /etc/djigzo/spring/
$ ln -s /usr/local/djigzo/conf/spring/services.xml /etc/djigzo/spring/
$ ln -s /usr/local/djigzo/conf/spring/soap.xml /etc/djigzo/spring/
$ ln -s /usr/local/djigzo/logs/james.wrapper.log /var/log/djigzo.log

$ ln -s /usr/local/djigzo/conf/james/SAR-INF/config.xml \
/etc/djigzo/james

$ ln -s /usr/local/djigzo/conf/james/SAR-INF/\
custom_processors_config.xml /etc/djigzo/james

$ ln -s /usr/local/djigzo/conf/james/SAR-INF/\
internal_remote_delivery_processor.xml /etc/djigzo/james

$ ln -s /usr/local/djigzo/conf/james/SAR-INF/\
smtp_server_config.xml /etc/djigzo/james

$ ln -s /usr/local/djigzo/conf/james/SAR-INF/\
smtp_transport_config.xml /etc/djigzo/james
```

**Protect files** Some files containing passwords should only be readable by user *djigzo*.

```
$ chmod 640 /usr/local/djigzo/conf/djigzo.properties
$ chmod 640 /usr/local/djigzo/conf/hibernate.cfg.xml
```

**Restart services** Restart Postfix, Djigzo and Tomcat.

```
$ /sbin/service postfix restart
$ /sbin/service djigzo restart
$ /sbin/service tomcat5 restart
```

**Open the Web Admin page** Djigzo should now be running (wait some time for Tomcat to startup). The login page can be accessed using the following URL<sup>18</sup> <https://192.168.178.2:8443/djigzo> (change the IP address accordingly)

<sup>18</sup>if Djigzo was installed as the root context the URL should be <https://192.168.178.2:8443>

**Note:** Djigzo comes with a default SSL certificate which is not trusted by your browser. You should therefore manually accept the HTTPS certificate.

**Login** Use the following login credentials:

```
username:  admin
password:  admin
```

**Note:** it can take some time to login after a restart because the web application need to be initialized upon first login.

**Log output** In case Djigzo is not running you can check the log files for errors.

**Djigzo log**

```
$ less /var/log/djigzo.log
```

**Tomcat log**

```
$ less /var/log/tomcat5/catalina.out
```

**Note:** OpenJDK installed by Red Hat/CentOS 5.4 already contains the “unlimited strength JCE policy files”.

## A Configure Tomcat on Debian 5

Tomcat on Debian 5 cannot start because a suitable JDK is not found:

```
no JDK found - please set JAVA_HOME failed!
```

The JDK path should be set in `/etc/default/tomcat`:

```
$ sudo bash -c 'echo "JAVA_HOME=/usr/lib/jvm/java-6-openjdk" >> \
/etc/default/tomcat5.5'
```

## B Using Jetty 6

This appendix will explain how to configure Jetty 6 for Djigzo. This guide will only explain how to install Jetty on Ubuntu. For installation instructions on installing Jetty on non-Ubuntu systems please see <http://jetty.codehaus.org/jetty/>. Configuration of Jetty for Djigzo should be similar for all Jetty installations.

**Note:** the latest .deb releases of Jetty can only be installed on Ubuntu 8.10 and higher. If you need to install Jetty 6 on a previous version of Ubuntu you either need to use an older version of Jetty (for example 6.1.17) or use the non-deb version.

### Install the required packages

```
$ sudo apt-get install libservlet2.5-java
```

### Download Jetty<sup>19</sup>

```
$ wget http://www.djigzo.com/downloads/jetty6_6.1.22-1_all.deb
$ wget http://www.djigzo.com/downloads/libjetty6-java_6.1.22-1_all.deb
```

### Install the deb files

```
$ sudo dpkg -i jetty6-*_all.deb libjetty6-java_6.*_all.deb
```

**Enable automatic startup** By default Jetty is not automatically started at reboot. To make sure that Jetty is started at system startup replace `NO_START=1` with `NO_START=0` in file `/etc/default/jetty6`.

```
$ sudo sed s/NO_START\s*=\s*1/NO_START=0/ /etc/default/jetty6 --in-place
```

### Configure Jetty

 Copy the required Jetty configuration files

```
$ sudo cp /usr/local/djigzo-web/conf/jetty/djigzo-jetty-ssl.xml \
/etc/jetty6/
```

```
$ sudo cp /usr/local/djigzo-web/conf/jetty/djigzo-jetty-context.xml \
/etc/jetty6/contexts/
```

---

<sup>19</sup>other Jetty releases can be downloaded from <http://dist.codehaus.org/jetty/>

**Set djigzo-web.home** The system property **djigzo-web.home** must reference the location where Djigzo web application is stored. The property will be added to the Jetty default config file.

```
$ sudo bash -c 'echo "JAVA_OPTIONS=\\"$JAVA_OPTIONS -Ddjigzo-web.home=\
/usr/local/djigzo-web\\"" >> /etc/default/jetty6'
```

**Load SSL config**

```
$ sudo sed $a\ /etc/jetty6/djigzo-jetty-ssl.xml \
/etc/jetty6/jetty.conf --in-place
```

**Allow reading and writing of SSL certificate** If you want to allow the upload of new SSL certificates using the Djigzo web admin SSL manager, Jetty should be allowed to read and write the SSL certificate.

```
$ sudo chown jetty /usr/local/djigzo-web/ssl/sslCertificate.p12
$ sudo chmod 660 /usr/local/djigzo-web/ssl/sslCertificate.p12
```

**Restart Jetty**

```
$ sudo /etc/init.d/jetty6 restart
```

**Open the Web Admin page** Djigzo should now be running (wait some time for Jetty to startup). The login page can be accessed using the following URL <https://192.168.178.2:8443> (change the IP address accordingly)

**Note:** Djigzo comes with a default SSL certificate which is not trusted by your browser. You should therefore manually accept the HTTPS certificate.

**Login** Use the following login credentials:

```
username: admin
password: admin
```

**Note:** it can take some time to login after a restart because the web application will be initialized upon first login.

## C Adding Tomcat HTTPS connector

The version of Tomcat server.xml that comes with Djigzo:

```
<Server>
  <Service name="Catalina">
    <Connector port="8443" maxHttpHeaderSize="8192"
      maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
      enableLookups="false" disableUploadTimeout="true"
      acceptCount="100" scheme="https" secure="true"
      clientAuth="false" sslProtocol="TLS"
      keystoreFile="/usr/local/djigzo-web/ssl/sslCertificate.p12"
      keystorePass="djigzo"
      keystoreType="PKCS12"
      ciphers="TLS_RSA_WITH_AES_128_CBC_SHA,
        TLS_RSA_WITH_AES_256_CBC_SHA,
        SSL_RSA_WITH_3DES_EDE_CBC_SHA,
        TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA,
        TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA,
        TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA,
        TLS_ECDH_RSA_WITH_AES_128_CBC_SHA,
        TLS_ECDH_RSA_WITH_AES_256_CBC_SHA,
        TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA,
        TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA,
        TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA,
        TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA,
        TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,
        TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,
        TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA"
    />

    <Engine name="Catalina" defaultHost="localhost">
      <Host name="localhost" appBase="webapps" unpackWARs="false"/>
    </Engine>
  </Service>
</Server>
```

If you want to use an existing server.xml you must add the Connector for port 8443 to the existing server.xml.

## D Memory usage

Because of some limitations of Javamail, Djigzo requires a lot of memory when it needs to encrypt large messages. By default Djigzo is setup with a maximum heap size of 512 MB which should be enough for sending messages up to 50 MB. If you need to send larger messages you need to increase the maximum heap size by modifying the property `wrapper.java.maxmemory` in file `/etc/djigzo/djigzo.wrapper.conf`. Alternatively you can set the memory based on the total available memory by adding the following lines to `/etc/default/djigzo`:

```
RESERVE=320
# On 32 bits Linux Java max mem is little less than 2048.
# on 64 bits Linux you can use much more memory.
# Set DJIGZO_MAX_MEM to the memory you want Djigzo to use
DJIGZO_MAX_MEM=$(( $(free -m | grep "Mem:" | awk '$2 > (2024 + \
'$RESERVE') { $2 = (2024 + '$RESERVE') } { print $2 }') - $RESERVE ))

echo "* Djigzo max memory setting: $DJIGZO_MAX_MEM MB"

# Additional options that are passed to the Daemon.
WRAPPER_OPTS=$WRAPPER_OPTS" wrapper.java.maxmemory=$DJIGZO_MAX_MEM"
```

This will ensure that the maximum heap size of Djigzo is set to the total available memory minus 320 MB (you can change the value of `RESERVE` if other processes on the system require more memory)

## E Securing the gateway

### E.1 Port usage

Djigzo uses the following ports for external → internal (listening ports)

Port	Service	Description
22	SSH	Console access
25	SMTP	Send/Receive email
8080	HTTP	Web manager
8443	HTTPS	Web manager
9000	SOAP (HTTP)	Back end

and for internal → external

Port	Service	Description
25	SMTP	Send/Receive email
80	HTTP	CRL download
139	SMB/CIFS	remote backup and restore
398	LDAP	CRL download
443	HTTPS	CRL download
445	SMB/CIFS	remote backup and restore

With a default setup of Djigzo, ie. the encryption back-end and web front-end are running on the same local machine, remote access to port 9000 is not required. You are advised to block remote access to all ports that need not be externally accessible.

**Enable Ubuntu firewall** Ubuntu can be protected by installing the “Uncomplicated Firewall” (UFW) with the following commands:

```
$ sudo apt-get install ufw
$ sudo ufw allow smtp/tcp
$ sudo ufw allow ssh/tcp
$ sudo ufw allow 8443/tcp
$ sudo ufw allow 8080/tcp
$ sudo ufw enable
```

Red Hat/CentOS already comes with a pre-installed firewall.

### E.2 Passwords

**Database** By default Djigzo creates a database user *djigzo* with the password *djigzo*. If you would like to change the password you should change the database password for user *djigzo* (see PostgreSQL documentation) and change the password in the file `/user/share/djigzo/conf/hibernate.cfg.xml`.

**Back-end** The front-end (Web application) communicates with the back-end (encryption engine) using password authenticated SOAP messages. If the back-end and front-end are running on separate systems you are advised to change

the SOAP password. For the back-end the password should be changed in the file `/usr/local/djigzo/conf/djigzo.properties` (see property **protected.system.soap.password**). You can change the password for the front-end by adding a property **soap.password** with the password as the property value to `/etc/default/tomcat5.5`<sup>20</sup> in a similar way as **djigzo-web.home** was set (see 3.5).

### E.3 SSL certificate

Access to the administration page is protected with an encrypted HTTPS connection. Djigzo comes with a default SSL certificate. You are advised to install your own SSL certificate using the “SSL certificate manager” from the Djigzo web administration page.

### E.4 Prevent spoofing the From header

Djigzo uses the *From* header as the identity of the sender. If the Djigzo gateway is used for sending email to external recipients (ie. relaying email) you should make sure that internal users cannot ‘spoof’ the *From* header.

### E.5 Securing the database

Unless a “Hardware Security Module” (HSM) is used, all private keys used for signing and decrypting of email are stored in the database. The database therefore has to be protected against unauthorized access. If Djigzo and the database are on the same machine the djigzo database user should not be allowed to access the database across the network. This is done by making sure that only localhost (127.0.0.1) can login with the username *djigzo*. The PostgreSQL config file `pg_hba.conf` should contain a line similar to:

```
host djigzo djigzo 127.0.0.1/32 md5
```

### E.6 Block access to pages

If the PDF reply functionality is used, external access to the gateway should be granted to all external IP addresses (otherwise the recipients of the encrypted PDF message cannot open the reply page). You are advised to only allow access to the PDF reply pages and block access to all other pages. Access to the following URLs should be granted for all IP addresses: <https://192.168.178.24:8443/external/pdf/>\* (the IP address should be the external IP address and \* means that access should be granted to all parent URLs). There are multiple ways to block access to most of the gateway pages while allowing access to the PDF reply page:

**Block access with a firewall** If your firewall is capable of blocking access at the HTTP(s) level you should add a rule to block access to all URLs except to the PDF reply page URL.

<sup>20</sup>For Red Hat/CentOS the property should be added to `/etc/sysconfig/tomcat5`. If Jetty is used you should add the property to `/etc/default/jetty6`

**Use Apache as a front-end** Use Apache as a front-end to the gateway. Apache will handle all HTTP(s) access. Apache can be setup to only allow access to certain URLs. Add a rule to block access to all URLs except to the PDF reply page URL.

**Enable the built-in IP filter** Djigzo can be setup to only allow access to the management pages from certain IP ranges. A property `djigzo.ipfilter.network` with the IP filter as the property value should be added to `/etc/default/tomcat5.5`<sup>21</sup> in a similar way as `djigzo-web.home` was set (see 3.5).

**Example:** `-Ddjigzo.ipfilter.network=192.168.178.20`

**Filter** The IP filter should be a comma separated list of IP ranges. Some example filters:

- a) `192.168.*`
- b) `192.168.*, 127.*, 222.0.0.0/8`

**Use a dedicate WAR** Another option is to install an alternative Web Application Archive (WAR) with only support for the PDF reply functionality. Instead of using `djigzo.war` in the web context (see section 2.5) `djigzo-external.war` should be used instead.

---

<sup>21</sup>For Red Hat/CentOS the property should be added to `/etc/sysconfig/tomcat5`. If Jetty is used you should add the property to `/etc/default/jetty6`