

DJIGZO EMAIL ENCRYPTION

Djigzo Gateway Installation Guide



November 4, 2011, Rev: 6572

Copyright © 2008-2011, djigzo.com.

Acknowledgments: Thanks goes out to Andreas Hödle for feedback and input on gateway security.

Contents

1	Introduction	3
2	Install Djigzo on Ubuntu/Debian	3
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
2.5.1	Install Tomcat 5	8
2.5.2	Install Tomcat 6	10
2.6	Finalize	11
3	Install Djigzo on Red Hat 5/CentOS 5	13
3.1	Configure PostgreSQL	13
3.2	Install Djigzo	15
3.3	Update sudoers	16
3.4	Configure Postfix	17
3.5	Install Tomcat	20
3.6	Finalize	21
4	Install Djigzo on Red Hat 6/CentOS 6	23
4.1	Configure PostgreSQL	23
4.2	Install Djigzo	24
4.3	Update sudoers	25
4.4	Configure Postfix	26
4.5	Install Tomcat	29
4.6	Finalize	30
A	Configure Tomcat on Debian 5	32
B	Using Jetty 6	32
C	Adding Tomcat HTTPS connector	34
C.1	Tomcat 5	34
C.2	Tomcat 6	35
D	Memory usage	36
E	Securing the gateway	37
E.1	Port usage	37
E.2	Passwords	37
E.3	SSL certificate	38
E.4	Prevent spoofing the From header	38
E.5	Securing the database	38
E.6	Block access to pages	38

1 Introduction

This installation guide provides step-by-step instructions on how to install Djigzo. If Djigzo is going to be installed on Ubuntu, Debian, 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 Djigzo need to be installed on a system not supported by the .deb or .rpm packages, use this guide. Although this guide assumes that Djigzo will be installed on Ubuntu, Debian, Red Hat or CentOS, installation on other system will be similar and typically require only minor changes. You are recommended to install Djigzo on a dedicated and clean machine.

Requirements

- 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). It is recommended to copy and paste the commands directly to the command line.

WARNING do not install Djigzo on a live email system!

2 Install Djigzo on Ubuntu/Debian

This section explains how to install Djigzo on Ubuntu and Debian.

Install required packages¹

```
$ 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, select “No Configuration”.

Make OpenJDK the default If there are multiple Java runtimes installed, OpenJDK must be set as the default JRE.

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

¹The sudo package is required by Djigzo. Debian does not install sudo by default. If installing on Debian, sudo must be installed prior to installing Djigzo.

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 PostgreSQL²

Djigzo stores all settings in a PostgreSQL database.

Create database user Create the database user *djigzo* with password *djigzo*³.

```
$ 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* should 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 files⁴.

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

Create a directory for Djigzo-web 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 Djigzo encryption back-end and the Web GUI front-end. Both can be downloaded from <http://www.djigzo.com>. The following two .tar.gz files are required⁵:

```
djigzo_2.3.1-7.tar.gz  
djigzo-web_2.3.1-7.tar.gz
```

²Djigzo should work with all databases supported by Hibernate. Installation instructions for different databases however is beyond the scope of this manual.

³The encoded password is equal to 'md5' concatenated with the MD5 hash of the username and password.

⁴Only required if Djigzo Web GUI should be allowed to show Postfix log file content.

⁵The 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 Import the database schema 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
```

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

Update location of Djigzo Djigzo should be automatically started at system startup. The startup script should 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 directory and /etc/rc?.d will be updated.

```
$ 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 scripts for which root access is required. To allow Djigzo to start these scripts, certain lines should be added to the sudoers file.

Note: Djigzo will function even if these scripts are not added to the sudoers file. You will however not be allowed to execute the following functions from the Web GUI: configure Postfix, manage Postfix queues, backup and restore, restart and installation of SSL certificates.

Make root owner of scripts The scripts will be run as root. They therefore should be owned by root and not be 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)⁶. Djigzo functions as a Postfix “after queue filter”. Postfix should therefore be configured to work with the encryption back-end.

Two pre-configured Postfix configuration files, main.cf and master.cf, can be found in /usr/local/djigzo/conf/system. It is recommended that these Postfix configuration files are used. If Postfix is already configured and the existing settings should not be overwritten, the existing Postfix configuration files should be manually merged with the configuration files provided by Djigzo. The most important Postfix configuration settings required by Djigzo will be discussed next.

main.cf configuration The Postfix main configuration file should contain at minimal the **content_filter** setting which tells Postfix that all email should be

⁶It is possible to use another MTA instead of Postfix, like for example Exim, but that's beyond the scope of this manual.

handled by the Djigzo encryption back-end. The `content_filter` setting tells Postfix that the service running on `127.0.0.1:10025` will function as an “after queue filter”⁷.

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

The other settings in the pre-configured `main.cf` file are only required for the MTA configuration page of Djigzo Web GUI. Settings starting with **`djigzo_`** will be replaced when changes on the MTA page are applied. The **`djigzo_...`** settings are used by `main.cf` and `master.cf` (the settings are referenced as `$(djigzo_...)`).

master.cf configuration The Postfix master configuration file 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

```

message_size_limit Because Djigzo functions as an “after queue filter” the message size can be increased after being handled by the encryption back-end. For example, signing a message will slightly increase the message because of the S/MIME signature. 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 should be set on the `smtpd` service.

```

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

```

⁷If you already configured a `content_filter` you should configure additional filters in `master.cf`. This will however not be explained in this guide.

Copy postfix config files It is advised to use the pre-configured Postfix configuration files. The pre-configured configuration files should be copied 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 Tomcat⁸

If Djigzo is installed on Ubuntu 8.04 or Debian 5, Tomcat 5 should be used. If Djigzo is installed on Ubuntu 10.04 or Debian 6, Tomcat 6 should be used.

2.5.1 Install Tomcat 5

Install the required Tomcat package (for Tomcat 6, see next section)

```
$ 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 GUI is stored. The property will be added to the Tomcat default config file.

```
$ sudo bash -c 'echo "JAVA_OPTS=\"\$JAVA_OPTS -Ddjigzo-web.home=\
/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 setup with max. 256 MB heap space.

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

⁸if you would like to use Jetty instead of Tomcat skip the installation of Tomcat. See Appendix B for instructions on installing Jetty.

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 Djigzo Web GUI allows new SSL certificates for the Web GUI to be uploaded using the SSL page. To support this functionality, Tomcat should be allowed to read and write the SSL certificate.

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

Adding an HTTPS connector An HTTPS connector should be added to the Tomcat server configuration. If Tomcat installation is only used by Djigzo, it's 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 should 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 the Web admin context A context should be added to Tomcat to enable the Web admin application.

```
$ 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 context file to ROOT.xml (overwriting the existing file) instead of to djigzo.xml⁹.

Adding the Web portal context If the portal functionality is required, a specific portal context should be added to Tomcat.

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

Restart Tomcat Tomcat must be restarted to make it use the new Tomcat configuration.

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

⁹the 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>

2.5.2 Install Tomcat 6

Install the required Tomcat package

```
$ sudo apt-get install tomcat6
```

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

```
$ sudo bash -c 'echo "JAVA_OPTS=\"\$JAVA_OPTS -Ddjigzo-web.home=\
/usr/local/djigzo-web\"" >> /etc/default/tomcat6'
```

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 setup with max. 256 MB heap space.

```
$ sudo bash -c 'echo "JAVA_OPTS=\"\$JAVA_OPTS \
-Djava.awt.headless=true -Xmx256M\"" >> /etc/default/tomcat6'
```

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 "TOMCAT6_SECURITY=no" >> /etc/default/tomcat6'
```

Allow reading and writing of SSL certificate Djigzo Web GUI allows new SSL certificates for the Web GUI to be uploaded using the SSL page. To support this functionality, Tomcat should be allowed to read and write the SSL certificate.

```
$ sudo chown tomcat6:djigzo /usr/local/djigzo-web/ssl/sslCertificate.p12
```

Adding an HTTPS connector An HTTPS connector should be added to the Tomcat server configuration. If Tomcat is only used by Djigzo, it's advised to replace the existing Tomcat configuration file (/etc/tomcat6/server.xml) with the configuration file provided by Djigzo.

```
$ sudo cp /usr/local/djigzo-web/conf/tomcat/server-T6.xml \
/etc/tomcat6/server.xml
```

The path to djigzo-web should be updated

```
$ sudo sed s#/share/djigzo-web/#/local/djigzo-web/# \
/etc/tomcat6/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 the Web admin context A context should be added to Tomcat to enable the Web admin application.

```
$ sudo bash -c 'echo "<Context docBase=\"/usr/local/djigzo-web/djigzo.war\"
\" unpackWAR=\"false\"/>" > /etc/tomcat6/Catalina/localhost/djigzo.xml'
```

Note: if you want Djigzo to use the root context, save the context file to ROOT.xml (overwriting the existing file) instead of to djigzo.xml¹⁰.

Adding the Web portal context If the portal functionality is required, a specific portal context should be added to Tomcat.

```
$ sudo bash -c 'echo "<Context docBase=\"/usr/local/djigzo-web/djigzo-portal.war\"
\" unpackWAR=\"false\"/>" > /etc/tomcat6/Catalina/localhost/web.xml'
```

Restart Tomcat Tomcat should be restarted to make it use the new Tomcat configuration.

```
$ sudo /etc/init.d/tomcat6 restart
```

2.6 Finalize

Create a softlink to the djigzo log file.

```
$ sudo ln -s /usr/local/djigzo/logs/james.wrapper.log /var/log/djigzo.log
```

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 and Djigzo.

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

Open the Web GUI Djigzo should now be running (wait some time for Tomcat to startup). The login page can be accessed using the following URL <https://192.168.178.2:8443/djigzo>¹¹ (change the IP address accordingly)

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

¹⁰the 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>

¹¹if Djigzo was installed as the root context, the URL should be <https://192.168.178.2:8443>

Login Use the following login credentials:

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

Note: the login procedure can take some time after a restart because the Web GUI requires some internal initialization after a restart.

Log output If Djigzo is not running, check the following log files for errors:

Djigzo log

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

Tomcat 5 log

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

Tomcat 6 log

```
$ sudo less /var/log/tomcat6/catalina.out
```

Note: replace * with the current date to view the most recent log file.

3 Install Djigzo on Red Hat 5/CentOS 5

This section explains how to install Djigzo on Red Hat 5 and CentOS 5. 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 block access to most ports. The firewall should therefore be configured to allow access to certain ports used by Djigzo. The following ports should be remotely accessible: 25 (*SMTP*) and 8443¹². The firewall can be configured with the `system-config-securitylevel-tui` tool.

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

Note: port numbers should be postfixed with `.tcp`. For example, to open port 8443, add `8443:tcp` to the port configuration.

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, make 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 PostgreSQL¹³

Djigzo stores all settings in a PostgreSQL database.

Make PostgreSQL autostart PostgreSQL should be started at reboot.

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

¹²See Appendix E.1 for an overview of all ports used by Djigzo.

¹³Djigzo should work with all databases supported by Hibernate. Installation instructions for different databases however is beyond the scope of this manual.

3.1 Configure PostgreSQL 3 INSTALL DJIGZO ON RED HAT 5/CENTOS 5

Start PostgreSQL

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

Enable password authentication By default PostgreSQL does not allow applications to login with user name and password. PostgreSQL should be configured to allow login with user name and password 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

Enable autovacuum With the default install of PostgreSQL on RedHat/CentOS, the *autovacuum* service is not enabled. The PostgreSQL vacuum command should be run on a regular basis to keep the database in optimal shape and to make sure that disk space occupied by updated or deleted rows is automatically recovered. The *autovacuum* service should be enabled by uncommenting and changing the following settings in the PostgreSQL main configuration file.

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

```
stats_start_collector = on
stats_row_level = on
autovacuum = on
autovacuum_naptime = 60
```

After changing these settings, PostgreSQL should be restarted.

Restart PostgreSQL PostgreSQL should be restarted for the changes to take effect.

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

Create database user Create the database user *djigzo* with password *djigzo*¹⁴.

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

¹⁴The encoded password is equal to 'md5' concatenated with the MD5 hash of the user name and password.

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` should 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 Djigzo encryption back-end and the Web GUI front-end. Both can be downloaded from <http://www.djigzo.com>. The following two .tar.gz files are required¹⁵:

```
djigzo_2.3.1-7.tar.gz  
djigzo-web_2.3.1-7.tar.gz
```

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, text relocation for the Java wrapper lib should be enabled.

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

Importing the database schema Import the database schema into PostgreSQL.

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

¹⁵The exact version will be different when a new version is released.

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 should be automatically started at system startup. The startup script should 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 directory and /etc/rc?.d will be updated.

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

3.3 Update sudoers

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

Note: Djigzo will function even if these scripts are not added to the sudoers file. You will however not be allowed to execute the following functions from the Web GUI: configure Postfix, manage Postfix queues, backup and restore, restart and installation of SSL certificates.

Make root owner of scripts The scripts will be run as root. They therefore should be owned by root and not be 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 *should* 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)¹⁶. Djigzo functions as a Postfix “after queue filter”. Postfix should therefore be configured to work with the encryption back-end.

Two pre-configured Postfix configuration files, main.cf and master.cf, can be found in /usr/local/djigzo/conf/system and /usr/local/djigzo/dist/redhat/. It is recommended that these Postfix configuration files are used. If Postfix is already configured and the existing settings should not be overwritten, the existing Postfix configuration files should be manually merged with the configuration files provided by Djigzo. The most important Postfix configuration settings required by Djigzo will be discussed next.

main.cf configuration The Postfix main configuration file should contain at minimal the **content_filter** setting which tells Postfix that all email should be handled by the Djigzo encryption back-end. The content_filter setting tells Postfix that the service running on 127.0.0.1:10025 will function as an “after queue filter”¹⁷.

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

The other settings in the pre-configured main.cf file are only required for the MTA configuration page of Djigzo Web GUI. Settings starting with **djigzo_** will be replaced when changes on the MTA page are applied. The **djigzo_...** settings are used by main.cf and master.cf (the settings are referenced as `$(djigzo_...)`).

¹⁶It is possible to use another MTA instead of Postfix, like for example Exim, but that’s beyond the scope of this manual.

¹⁷If you already configured a content_filter you should configure additional filters in master.cf. This will however not be explained in this guide.

master.cf configuration The Postfix master configuration file 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

```

message_size_limit Because Djigzo functions as an “after queue filter” the message size can be increased after being handled by the encryption back-end. For example, signing a message will slightly increase the message because of the S/MIME signature. 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 should be set on the *smtpd* service.

```

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

```

Copy postfix config files It is advised to use the pre-configured Postfix configuration files. The pre-configured configuration files should be copied 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.

```

$ 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/aliases` as the alias file. Make sure that the alias file is available and up-to-date.

```
$ newaliases
```

Make Postfix the default MTA A Postfix after queue filter is used for encrypting and decrypting incoming and outgoing email. 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, Sendmail can be removed.

```
$ 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 “rejection” 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 should 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
```

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
```

Set djigzo-web.home The system property **djigzo-web.home** should reference the location where Djigzo Web GUI 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 setup with max. 256 MB heap space.

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

Adding an HTTPS connector An HTTPS connector should be added to the Tomcat server configuration. If Tomcat is only used by Djigzo, it's 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 should 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 the Web admin context A context should be added to Tomcat to enable the Web admin application.

```
$ 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 context file to ROOT.xml (overwriting the existing file) instead of to djigzo.xml¹⁸.

Adding the Web portal context If the portal functionality is required, a specific portal context should be added to Tomcat.

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

Allow reading and writing of SSL certificate Djigzo Web GUI allows new SSL certificates for the Web GUI to be uploaded using the SSL page. To support this functionality, 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
```

3.6 Finalize

Create a softlink to the djigzo log file.

```
$ ln -s /usr/local/djigzo/logs/james.wrapper.log /var/log/djigzo.log
```

¹⁸the 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>

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 GUI Djigzo should now be running (wait some time for Tomcat to startup). The login page can be accessed using the following URL <https://192.168.178.2:8443/djigzo>¹⁹ (change the IP address accordingly)

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

Login Use the following login credentials:

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

Note: the login procedure can take some time after a restart because the Web GUI requires some internal initialization after a restart.

Log output If Djigzo is not running, check the following log files for errors:

Djigzo log

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

Tomcat log

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

¹⁹if Djigzo was installed as the root context, the URL should be <https://192.168.178.2:8443>

4 Install Djigzo on Red Hat 6/CentOS 6

This section explains how to install Djigzo on Red Hat 6 and CentOS 6. 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 block access to most ports. The firewall should therefore be configured to allow access to certain ports used by Djigzo. The following ports should be remotely accessible: SMTP (25) and 8443²⁰. The firewall can be configured with the `system-config-firewall-tui` tool.

```
$ yum install system-config-firewall-tui
$ system-config-firewall-tui
```

Note: Port 25 can be opened by selecting *Mail (SMTP)* in the *Trusted Services* list. Port 8443 with protocol `tcp` should be added to the "Other Ports".

Install required packages

```
$ yum install redhat-lsb postgresql postgresql-server postfix \
java-1.6.0-openjdk-devel ant ant-nodeps mktemp symlinks wget
```

4.1 Configure PostgreSQL²¹

Djigzo stores all settings in a PostgreSQL database.

Initialize PostgreSQL

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

Make PostgreSQL autostart PostgreSQL should be started at reboot.

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

Enable password authentication By default PostgreSQL does not allow applications to login with user name and password. PostgreSQL should be configured to allow login with user name and password 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

²⁰See Appendix E.1 for an overview of all ports used by Djigzo.

²¹Djigzo should work with all databases supported by Hibernate. Installation instructions for different databases however is beyond the scope of this manual.

Restart PostgreSQL

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

Create database user Create the database user *djigzo* with password *djigzo*²².

```
$ 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
```

4.2 Install Djigzo

User and group *djigzo* with home dir */usr/local/djigzo* should 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 Djigzo encryption back-end and the Web GUI front-end. Both can be downloaded from <http://www.djigzo.com>. The following two .tar.gz files are required²³:

```
djigzo_2.3.1-7.tar.gz  
djigzo-web_2.3.1-7.tar.gz
```

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
```

²²The encoded password is equal to 'md5' concatenated with the MD5 hash of the user name and password.

²³The exact version will be different when a new version is released.

allow text relocation if SELinux is enabled, text relocation for the Java wrapper lib should be enabled.

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

Importing the database schema Import the database schema 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 should be automatically started at system startup. The startup script should 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 directory and /etc/rc?.d will be updated.

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

4.3 Update sudoers

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

Note: Djigzo will function even if these scripts are not added to the `sudoers` file. You will however not be allowed to execute the following functions from the Web GUI: configure Postfix, manage Postfix queues, backup and restore, restart and installation of SSL certificates.

Make root owner of scripts The scripts will be run as root. They therefore should be owned by root and not be 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 *should* 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
```

4.4 Configure Postfix

Djigzo uses Postfix for sending and receiving of email (MTA)²⁴. Djigzo functions as a Postfix “after queue filter”. Postfix should therefore be configured to work with the encryption back-end.

Two pre-configured Postfix configuration files, `main.cf` and `master.cf`, can be found in `/usr/local/djigzo/conf/system` and `/usr/local/djigzo/dist/redhat/`. It is recommended that these Postfix configuration files are used. If Postfix is already configured and the existing settings should not be overwritten, the existing Postfix configuration files should be manually merged with the configuration files provided by Djigzo. The most important Postfix configuration settings required by Djigzo will be discussed next.

main.cf configuration The Postfix main configuration file should contain at minimal the **content_filter** setting which tells Postfix that all email should be handled by the Djigzo encryption back-end. The `content_filter` setting tells Postfix that the service running on `127.0.0.1:10025` will function as an “after queue filter”²⁵.

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

²⁴It is possible to use another MTA instead of Postfix, like for example Exim, but that's beyond the scope of this manual.

²⁵If you already configured a `content_filter` you should configure additional filters in `master.cf`. This will however not be explained in this guide.

The other settings in the pre-configured main.cf file are only required for the MTA configuration page of Djigzo Web GUI. Settings starting with **djigzo_** will be replaced when changes on the MTA page are applied. The **djigzo_...** settings are used by main.cf and master.cf (the settings are referenced as `#{djigzo_...}`).

master.cf configuration The Postfix master configuration file 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

```

message_size_limit Because Djigzo functions as an “after queue filter” the message size can be increased after being handled by the encryption backend. For example, signing a message will slightly increase the message because of the S/MIME signature. 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 should be set on the *smtpd* service.

```

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

```

Copy postfix config files It is advised to use the pre-configured Postfix configuration files. The pre-configured configuration files should be copied 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.

4.4 Configure Postfix 4 INSTALL DJIGZO ON RED HAT 6/CENTOS 6

```
$ cp /usr/local/djigzo/conf/system/main.cf /etc/postfix/main.cf
$ cp /usr/local/djigzo/dist/redhat/master-2.6.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.

```
$ newaliases
```

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 “rejection” 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 should 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
```

4.5 Install Tomcat

```
$ yum install tomcat6
```

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
```

Set djigzo-web.home The system property **djigzo-web.home** should reference the location where Djigzo Web GUI 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/tomcat6
```

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 setup with max. 256 MB heap space.

```
$ echo "JAVA_OPTS=\"\${JAVA_OPTS} \  
-Djava.awt.headless=true -Xmx256M\"" >> /etc/sysconfig/tomcat6
```

Adding an HTTPS connector An HTTPS connector should be added to the Tomcat server configuration. If Tomcat is only used by Djigzo, it's advised to replace the existing Tomcat configuration file (/etc/tomcat6/server.xml) with the configuration file provided by Djigzo.

```
$ cp /usr/local/djigzo-web/conf/tomcat/server-T6.xml /etc/tomcat6/server.xml
```

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

```
$ sed s#/share/djigzo-web/#/local/djigzo-web/# \  
/etc/tomcat6/server.xml --in-place
```

Due to a bug in Tomcat²⁶, the HTTP NIO connector sometimes fails when Firefox 7 is used. If support for Firefox 7 is required, the HTTP NIO connector should be replaced with the HTTP/1.1 connector.

```
$ sed s#org.apache.coyote.http11.Http11NioProtocol#HTTP/1.1# \
/etc/tomcat6/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 the Web admin context A context should be added to Tomcat to enable the Web admin application.

```
$ echo "<Context docBase=\"/usr/local/djigzo-web/djigzo.war\" unpackWAR=
\"false\"/>" > /etc/tomcat6/Catalina/localhost/djigzo.xml
```

Note: if you want Djigzo to use the root context, save the context file to ROOT.xml (overwriting the existing file) instead of to djigzo.xml²⁷.

Adding the Web portal context If the portal functionality is required, a specific portal context should be added to Tomcat.

```
$ echo "<Context docBase=\"/usr/local/djigzo-web/djigzo-portal.war\" unpackWAR=
\"false\"/>" > /etc/tomcat6/Catalina/localhost/web.xml
```

Allow reading and writing of SSL certificate Djigzo Web GUI allows new SSL certificates for the Web GUI to be uploaded using the SSL page. To support this functionality, 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 tomcat6 on
```

4.6 Finalize

Create a softlink to the djigzo log file.

```
$ ln -s /usr/local/djigzo/logs/james.wrapper.log /var/log/djigzo.log
```

²⁶https://issues.apache.org/bugzilla/show_bug.cgi?id=50072

²⁷the 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>

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 tomcat6 restart
```

Open the Web GUI Djigzo should now be running (wait some time for Tomcat to startup). The login page can be accessed using the following URL <https://192.168.178.2:8443/djigzo>²⁸ (change the IP address accordingly)

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

Login Use the following login credentials:

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

Note: the login procedure can take some time after a restart because the Web GUI requires some internal initialization after a restart.

Log output If Djigzo is not running, check the following log files for errors:

Djigzo log

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

Tomcat log

```
$ less /var/log/tomcat6/catalina.out
```

²⁸if Djigzo was installed as the root context, the URL should be <https://192.168.178.2:8443>

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 newer versions of Ubuntu. 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 Jetty²⁹

```
$ 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*/NO_START=0/ /etc/default/jetty6 --in-place
```

²⁹other Jetty releases can be downloaded from <http://dist.codehaus.org/jetty/>

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/
```

Set djigzo-web.home The system property **djigzo-web.home** should reference the location where Djigzo Web GUI 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: the login procedure can take some time after a restart because the Web GUI requires some internal initialization after a restart.

C Adding Tomcat HTTPS connector

Djigzo uses the following Tomcat server.xml configuration files.

C.1 Tomcat 5

```
<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>
```

C.2 Tomcat 6

```
<?xml version="1.0" encoding="UTF-8"?>
<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"
      SSLEnabled="true"
      sslProtocol="TLS"
      protocol="org.apache.coyote.http11.Http11NioProtocol"
      keystoreFile="/usr/local/djigzo-web/ssl/sslCertificate.p12"
      keystorePass="djigzo"
      keystoreType="PKCS12"
    />

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

Note: If an existing server.xml should be used, the Connector for port 8443 should be added to the existing server.xml.

D Memory usage

Because of some limitations of Javamaail, 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`³⁰:

```
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)

³⁰`/etc/default/djigzo` that comes with Djigzo already contains these lines. They are however commented out

E Securing the gateway

E.1 Port usage

Djigzo uses the following ports:

external → internal

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*

* By default the back-end SOAP service is only accessible from localhost (i.e., it is bound to localhost)

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

When the encryption back-end and Web GUI front-end are installed on the same machine, remote access to port 9000 is not required. It is advised to block remote access to all ports which are not used by Djigzo.

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 a different password should be used, the database password for user *djigzo* should be changed (see PostgreSQL documentation). The

database password in the database configuration file `/usr/local/share/djigzo/conf/hibernate.cfg.xml` should be changed accordingly.

Back-end The front-end (Web GUI) communicates with the back-end (encryption engine) using password authenticated SOAP messages. If the back-end and front-end are not installed on the same machine, it is advised to change the SOAP password.

For the back-end, the password can be changed by modifying the property **protected.system.soap.password** in file `/usr/local/share/djigzo/conf/djigzo.properties`. If the password for the back-end is changed, the password used by the front-end should be changed accordingly. The password for the front-end can be changed by adding a property **soap.password** with the password as the property value to `/etc/default/tomcat5.5`³¹ in a similar way as **djigzo-web.home** was set (see 4.5).

E.3 SSL certificate

Access to the administration page is protected with an encrypted HTTPS connection. Djigzo comes with a default SSL certificate. It is advised to install a new SSL certificate using the “SSL certificate manager” from the Djigzo Web GUI.

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 (i.e., relaying email), 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 PostgreSQL are installed on the same machine, the djigzo database user should only be allowed to access the database locally. 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). It is advised to only allow access to the PDF reply pages and block access to all other pages.

³¹If Jetty is used instead you should add the property to `/etc/default/jetty6`. If Tomcat 6 is used add the property to `/etc/default/tomcat6`

Access to the following URLs should be granted for all IP addresses: <https://192.168.178.24:8443/web/portal/>* (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 a firewall is used and the firewall is capable of blocking access at the HTTP(s) level, a rule should be added to block access to all URLs with the exception of the PDF reply page URL.

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/tomcat6`³² in a similar way as **djigzo-web.home** was set (see 4.5).

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`

³²For 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`