Deploying Fedora - Complete Guide

See the Quick Start guide to getting Fedora up and running as quickly as possible.
Although deploying Fedora is as easy as downloading the WAR file and copying to your servlet container's `webapps` directory, this document details the process.

Downloads

See the latest release for Fedora WAR files to download.

- Java 8

```
java -version
java version "1.8.0_05"
Java(TM) SE Runtime Environment (build 1.8.0_05-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.5-b02, mixed mode)
```

- Servlet 3.0 container such as:
    - Tomcat 8 or later
    - Jetty 9.x or later

Deploying with Tomcat 8

1. Download and install Tomcat
2. Set the Java properties for Tomcat (see: Application Configuration and Catalina Java Properties sections below)
3. Copy the Fedora WAR file into Tomcat's "webapps" directory (e.g. /var/lib/tomcat8/webapps)
4. Start the server
5. Go to the browser page that matches your Fedora WAR file name (e.g. http://localhost:8080/fcrepo-webapp-x.x.x/rest)

Deploying with Jetty 9

1. Download and install Jetty
2. Set the Java properties for Jetty (see: Application Configuration and Catalina Java Properties sections below)
3. Copy the Fedora WAR file into Jetty's "webapps" directory (e.g. /var/lib/jetty/webapps)
4. Start the server
5. Go to the browser page that matches your Fedora WAR file name (e.g. http://localhost:8080/fcrepo-webapp-x.x.x/rest)

Application Configuration

The Fedora web-application supports several deploy-time, system-level configuration options. These configuration elements are set via the definition of System Properties.

See: Best Practices - Fedora Configuration

Deployments

Four means of deploying Fedora have been verified

- Tomcat 8 servlet container
- Jetty 9 servlet container
- Maven jetty:run plugin - *for testing*
- One-Click Run - *for testing*

Each of these deployment approaches has its own way of setting System Properties.

Tomcat 8

On Debian Linux systems, the typical way of setting System Properties is to update the following file:

```
/etc/default/tomcat8
```

Within that file, new properties can be added per the example below:

```
JAVA_OPTS="${JAVA_OPTS} -Dfcrepo.modeshape.configuration=classpath:/config/file-simple/repository.json -Dfcrepo.
home=/mnt/fedora-data"
```

The `/config/file-simple/repository.json` configuration is good for test deployments. For production we recommend a MySQL/PostgreSQL backend. See Best Practices - Fedora Configuration

Additional information regarding the configuration of System Properties in Tomcat 8 can be found here.

Windows notes

Alternatively on Windows systems you can set the following file:

```
CATALINA_BASE/bin/setenv.bat (windows)
```

Within that file, new properties can be added per the example below:

```
set CATALINA_OPTS=%CATALINA_OPTS% -Dfcrepo.modeshape.configuration=classpath:/config/file-simple/repository.json
```

The `/config/file-simple/repository.json` configuration is good for test deployments. For production we recommend a MySQL/PostgreSQL backend. See Best Practices - Fedora Configuration

Jetty 9

On Debian Linux systems, one way of setting System Properties is to update the following file:

```
/etc/default/jetty
```

Within that file, new properties can be added per the example below (note the use of JAVA_OPTIONS instead of JAVA_OPTS):

```
JAVA_OPTIONS="${JAVA_OPTIONS} -Dfcrepo.modeshape.configuration=classpath:/config/file-simple/repository.json -
Dfcrepo.home=/mnt/fedora-data"
```

The `/config/file-simple/repository.json` configuration is good for test deployments. For production we recommend a MySQL/PostgreSQL backend. See Best Practices - Fedora Configuration

Additional information regarding the configuration of System Properties in Jetty 9 can be found here.

Windows notes

Alternatively on Windows systems you can set the following file:

```
{JETTY_DIST}/start.ini
```

Within that file, new properties can be added per the example below:

```
--exec
-Dfcrepo.home=/mnt/fedora-data
-Dfcrepo.modeshape.configuration=classpath:/config/file-simple/repository.json
```

The `/config/file-simple/repository.json` configuration is good for test deployments. For production we recommend a MySQL/PostgreSQL backend. See Best Practices - Fedora Configuration

Maven jetty:run

System Properties can be set when using the Maven jetty:run plugin by passing them per the example below:

```
mvn -Dfcrepo.home=/mnt/fedora-data -Dfcrepo.modeshape.configuration=classpath:/config/file-simple/repository.
json jetty:run
```

The `/config/file-simple/repository.json` configuration is good for test deployments. For production we recommend a MySQL/PostgreSQL backend. See Best Practices - Fedora Configuration

One-Click Run

If the One-Click Run is started from the command line, System Properties can be passed in per the example below:

```
java -Dfcrepo.home=/mnt/fedora-data -jar fcrepo-webapp-jetty-console.war
```

Configuration Elements

There are a number of configuration elements that can be optionally be set when starting the Fedora web-application, noted below within brackets: <>. The only configuration element that is required to be set is "`fcrepo.modeshape.configuration`".

```
fcrepo.home=<cwd/fcrepo4-data>
```

This can be set to a path (relative to the current working directory or absolute) to which Fedora repository content will be written. Any of the Modeshape configuration options below will default to being within this folder if unset or if set to a relative path. If unset, content will be put in the "fcrepo4-data" directory within the current working directory.

```
fcrepo.spring.configuration=<classpath:/config/spring/fcrepo-config.xml | file:/path/to/fcrepo-config.xml>
```

This specifies the location of the spring context configuration for the Fedora application it defaults to a provided configuration. For more configuration options review the `fcrepo-webapp-plus` supplied spring configuration.

```
fcrepo.modeshape.configuration=<classpath:/config/repository.json | file:/path/to/repository.json>
```

This specifies the configuration for the underlying Modeshape repository and is required to be set. See other available options for this value within the source tree. Example `repository.json` configurations are located within the WAR file within sub-directories under WEB-INF/classes/config. Note: this configuration file must be readable by the servlet container process.

```
java.io.tmpdir=</tmp on Linux, $TMPDIR on MacOSX, and %TEMP% on Windows>
```

This specifies the directory for writing temp files. You may need to set this property to a larger disk/filesystem to upload large files, particularly on Linux where /tmp is sometimes on a small partition.

```
fcrepo.jms.baseUrl=<http://localhost:8080/fcrepo/rest>
```

This specifies the baseUrl to use when generating JMS messages. You can specify the hostname with or without port and with or without path. If your system is behind a NAT firewall you may need this to avoid your message consumers trying to access the system on an invalid port. If this system property is not set, the host, port and context from the user's request will be used in the emitted JMS messages.

Note: If you have multiple instances of Fedora running, the following system properties must be set to avoid messaging port conflicts:

```
fcrepo.dynamic.jms.port=<default-of-61616>
fcrepo.dynamic.stomp.port=<default-of-61613>
```

This specifies the ports used by the embedded JMS-based message broker, both for OpenWire and STOMP protocols.

```
fcrepo.velocity.runtime.log=<$fcrepo.home/velocity.log>
```

The HTML template code uses Apache Velocity, which generates a runtime log called `velocity.log`. By default this is placed inside `fcrepo.home`, but it is possible to override the location to have it written to an alternate location.

```
fcrepo.external.content.allowed=</path/to/allowed.txt>
```

This provides the path to a file defining a list of allowed external binary content paths. If this parameter is not provided, then clients will be disallowed from creating external binary resources. See the external content allowed paths configuration for more details.

Modeshape

The Modeshape configurations mentioned above have a number of more detailed configuration elements that can optionally be set. To see exactly which elements are available to be set, inspect the *repository.json* files specified above. Those files can be browsed in the source tree in:

- the ModeShape configuration directories

Some common elements for Modeshape relate to the directories in which application information is persisted. As mentioned above, if no *fcrepo.home* property is set then application information will be persisted under the directory "fcrepo4-data" in the application's current working directory. There will then be several directories within "fcrepo4-data" that are named by default with the name of the property that can be set to configure each of those directories.

Below are some common examples of these directories:

```
fcrepo.modeshape.index.directory
```

Contains the internal Lucene index used by Modeshape/Admin Search.

```
fcrepo.activemq.directory
```

Contains the reliable messaging information maintained by ActiveMQ.

```
com.arjuna.ats.arjuna.common.ObjectStoreEnvironmentBean.default.objectStoreDir
com.arjuna.ats.arjuna.objectstore.objectStoreDir
```

Contain JBoss JTA transaction engine artifacts.

Configuration Chain

The standard configuration chain is as follows:

1. **fcrepo/fcrepo-webapp/src/main/webapp/WEB-INF/web.xml** contains a **context-param** element with **param-name** "contextConfigLocation". The **param-value** points to a repository.xml which includes either the default `fcrepo-config.xml` or your spring configuration file (as defined by the `fcrepo.spring.configuration` property.
2. Your spring configuration file contains a property **repositoryConfiguration** defining the location of your **repository.json**
3. **fedora-node-types.cnd** is defined in **repository.json**

Catalina Java Properties

fcrepo.home=<some-writable-directory>

Sets the home for Fedora's persisted data. Without this setting Fedora tries to use the current-working-directory as the home of persisted data. If the Tomcat user does not have write access to the installation area (e.g. /var/lib/tomcat8), then Fedora will not deploy. Set this system property to a directory writable by the tomcat process.

JVM Tuning Properties

We have a separate page with suggested VM options for general Java tuning.

Clustering Properties (only effective in a clustered configuration)

```
-Djgroups.tcp.address=<ip-address>
-Dfcrepo.ispn.numOwners=<num-nodes-in-cluster>
-Djava.net.PreferIPv4Stack=true
-Dfcrepo.ispn.replication.timeout=<timeout-in-ms>
```

- fcrepo.ispn.replication.timeout can be used to set the timeout of infinispan replication in a clustered environment