

Djatoka

Overview

The Djatoka JPEG 2000 Image Server is a plug-in image server that images can be sent to for processing. It can function with an installation of Apache Tomcat as a web app. Islandora uses Djatoka to prepare TIFF images for viewing, particularly with the Internet Archive Bookreader

Dependencies

- A working web app server to install it to (Tomcat - bundled with Fedora - in a default installation)
- [Oracle Java JDK 8](#) (Djatoka requires Sun/Oracle Java, not [OpenJDK 8](#))

Provisions

- [Islandora Internet Archive Bookreader](#)
- [Large Image Solution Pack](#)
- [Open Seadragon](#)

Downloads

Djatoka can be downloaded from the project's Sourceforge page at <http://sourceforge.net/projects/djatoka/files/djatoka/>.

Installation

Installation Steps:

1. Download and install Djatoka on your Fedora server by following the installation instructions at: <http://sourceforge.net/p/djatoka/code/HEAD/tree/INSTALL.txt>.
2. For this example, we have installed Djatoka in /opt/adore-djatoka-1.1. Once Djatoka has been verified and installed, you will need to change the environment variables in the \$CATALINA_HOME/bin/startup.sh script. At the very end of the file, insert the following:

```
. /opt/adore-djatoka-1.1/bin/env.sh

export JAVA_OPTS

echo $JAVA_OPTS

exec "$PRGDIR"/"$EXECUTABLE" start "$@"
```

3. Then, modify the env.sh file found in /opt/adore-djatoka-1.1/bin/

a) At the top of the file:

Change this:	To this:
<pre>#!/bin/sh # setup environment variables for shell script # Define DJATOKA_HOME dynamically LAUNCHDIR=\$PWD cd .. DJATOKA_HOME=`pwd` LIBPATH=\$DJATOKA_HOME/lib</pre>	<pre>#!/bin/sh # setup environment variables for shell script CURRENTDIR=\$PWD # Define DJATOKA_HOME dynamically LAUNCHDIR=/opt/adore-djatoka-1.1/bin #cd .. DJATOKA_HOME=/opt/adore-djatoka-1.1 LIBPATH=\$DJATOKA_HOME/lib</pre>

b) Then, at the bottom of the file:

Change this:	To this:

<pre> KAKADU_HOME=\$DJATOKA_HOME/bin/\$PLATFORM export KAKADU_HOME cd \$LAUNCHDIR for line in `ls -1 \$LIBPATH   grep '.jar'` do classpath="\$classpath:\$LIBPATH/\$line" done #DEBUG="-Xdebug -Xrunjwp:transport=dt_socket,address=8000, server=y,suspend=n" CLASSPATH=../build/:\$classpath JAVA_OPTS="\$DEBUG -Djava.awt.headless=true -Xmx512M -Xms64M -Dkakadu.home=\$KAKADU_HOME -Djava.library.path=\$LIBPATH /\$PLATFORM \$KAKADU_LIBRARY_PATH" </pre>	<pre> KAKADU_HOME=\$DJATOKA_HOME/bin/\$PLATFORM #cd \$LAUNCHDIR #for line in `ls -1 \$LIBPATH   grep '.jar'` # do # classpath="\$classpath:\$LIBPATH/\$line" #done #go back to tomcat dir #cd \$CURRENTDIR #DEBUG="-Xdebug -Xrunjwp:transport=dt_socket,address=8000, server=y,suspend=n" #CLASSPATH=../build/:\$classpath JAVA_OPTS="\$JAVA_OPTS -Djava.awt.headless=true -Dkakadu. home=\$KAKADU_HOME -Djava.library.path=\$LIBPATH /\$PLATFORM \$KAKADU_LIBRARY_PATH" </pre>
--	---

- Restart Fedora and test the Djatoka application by going to: [http://\[fedora-server\]:8080/adore-djatoka](http://[fedora-server]:8080/adore-djatoka)

To test your installation, click the 'Update Image' button (make sure the BaseURL field is correct) If you do not see part of an image from the Library of Congress there is a problem with your Djatoka installation. Go back and review the installation instructions to make sure that they were followed correctly.

Though not required, it is a good idea to set the path for Djatoka log files. You can do this by adding a line to the log4j.properties file. For example:

 log.dir=/usr/local/fedora/server/logs/

Accommodating Viewing Restrictions for Djatoka

The Djatoka image viewer must send authentication credentials to Fedora in order to get titles back. For most set-ups, this communication between Djatoka and Fedora will be direct. However, if you wish to put Djatoka behind an authentication wall to restrict the view of objects to certain users in Fedora, you will need to make some additional configurations.

The islandora-servlet-filter.jar, which is part of the earlier Drupal Server Filter download, is required to facilitate this communication. The islandora-servlet-filter.jar should reside in \$CATALINA_HOME/webapps/fedora/WEB-INF/lib

Installation Steps:

- Make the following change to the web.xml located in \$CATALINA_HOME/webapps/fedora/WEB-INF/

```

<filter>
<filter-name>DrupalFilter</filter-name>
<filter-class>ca.upei.roblib.fedora.servletfilter.FilterDrupal</filter-class>
</filter>

<filter>
<filter-name>IslandoraAuthFilter</filter-name>
<filter-class>ca.upei.roblib.fedora.servletfilter.FilterIslandoraAuth</filter-class>
</filter>

<filter>
<filter-name>RestApiAuthnFilter</filter-name>
<filter-class>fedora.server.security.servletfilters.FilterRestApiAuthn</filter-class>
</filter>

...

<filter-mapping>
<filter-name>DrupalFilter</filter-name>
<url-pattern>/*</url-pattern>
</filter-mapping>

<filter-mapping>
<filter-name>IslandoraAuthFilter</filter-name>
<url-pattern>/*</url-pattern>
</filter-mapping>

<filter-mapping>
<filter-name>EnforceAuthnFilter</filter-name>
<servlet-name>AxisServlet</servlet-name>
</filter-mapping>

```

2. Then, modify the \$FEDORA_HOME/server/config/filter-drupal.xml file by adding the following the bottom of the file, just before </FilterDrupal_Connection>

```
<service_requests>
<service name="djatoka">

<!-- space-separated list of allowed ips. use 'local' to allow requests from same ip as fedora -->
<allowed_ips>local</allowed_ips>

<!-- request uri must match this regex to be eligible for service access -->
<allowed_uri_pattern>^.*(JP2\|TEI)$</allowed_uri_pattern>

<!-- <allowed_uri_pattern>^.*$</allowed_uri_pattern> -->

<!-- space-separated list of roles granted to service requests -->
<roles>administrator fedora_anonymous</roles>

</service>
</service_requests>
```

3. Restart Fedora to apply the changes.
4. To test the filter, access a Fedora with large image objects that have restricted security policies.