Chapter 9 - Installing Solution Pack & Tool Dependencies

On This Page

- Overview of Dependencies
- Apache Reverse Proxy
- Djatoka
- Accommodating Viewing Restrictions for Djatoka
- ExifTool
- ImageMagick
- Islandora Viewer
- Lame
- Kakadu
- SWF Tools
- Tesseract
- Video Dependencies

Overview of Dependencies

Most Islandora Solution Packs and some modules will require that you have one or more of the following dependencies installed so that Islandora Content Models can perform prescribed actions on your data objects. Review Chapter 5 - Islandora Solution Packs to see what dependencies are required for the solution packs you wish to install.

Apache Reverse Proxy

If you wish to use an iframe with the Islandora Viewer you must configure Apache with a reverse proxy. Note: These instructions are for configurations with Fedora running on the same server as Drupal.

Configuration Steps:

1. Add the following to Apache's httpd.conf file or desired virtual host, if using Apache virtual hosts:

```
ProxyRequests Off
ProxyPreserveHost On
<Proxy \*>
        Order deny,allow
        Allow from all
</Proxy>
ProxyPass /fedora/get http://localhost:8080/fedora/get
ProxyPassReverse /fedora/get http://localhost:8080/fedora/get
ProxyPass /fedora/services http://localhost:8080/fedora/services
ProxyPassReverse /fedora/services http://localhost:8080/fedora/services
ProxyPass /fedora/describe http://localhost:8080/fedora/describe
ProxyPassReverse /fedora/describe http://localhost:8080/fedora/describe
ProxyPass /iiv http://localhost:8080/iiv
ProxyPassReverse /iiv http://localhost:8080/iiv
ProxyPass /fedora/risearch http://localhost:8080/fedora/risearch
ProxyPassReverse /fedora/risearch http://localhost:8080/fedora/risearch
ProxyPass /adore-djatoka http://localhost:8080/adore-djatoka
ProxyPassReverse /adore-djatoka http://localhost:8080/adore-djatoka
```

2. Test your configuration by navigating to http://localhost/fedora/risearch to view the Fedora resource index page.

Djatoka

Djatoka is a Java-based open source image server that provides compression and region extraction of JPEG 2000 images, URI-addressability of regions, and support for a rich set of input/output image formats (e.g., BMP, GIF, JPG, PNG, PNM, TIF, JPEG 2000).

Installation Steps:

1. Download and install Djatoka on your Fedora server by following the installation instructions at: http://sourceforge.net/apps/mediawiki/djatoka/index.php?title=Installation.
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:

#!/bin/sh
# setup environment variables for shell script 	
 	
# Define DJATOKA_HOME dynamically 	
LAUNCHDIR=$PWD
cd ..
 	
DJATOKA_HOME=`pwd`
LIBPATH=$DJATOKA_HOME/lib
	#!/bin/sh
	# setup environment variables for shell script
	CURRENTDIR=$PWD
	# Define DJATOKA_HOME dynamically
	LAUNCHDIR=/opt/adore-djatoka-1.1/bin
	#cd ..
	
	DJATOKA_HOME=/opt/adore-djatoka-1.1
	LIBPATH=$DJATOKA_HOME/lib

   b) Then, at the bottom of the file:

Change this:	To this:

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 -Xrunjdwp: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 $KAKADU_LIBRARY_PATH"
	KAKADU_HOME=$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 -Xrunjdwp: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"

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

   To test your installation, click the 'Update Image' button. 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 resides in $CATALINA_HOME/webapps/fedora/WEB-INF/lib

Installation Steps:

1. 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.

ExifTool

ExifTool is a platform-independent Perl library plus a command-line application for reading, writing and editing metadata in a wide variety of formats. ExifTool is used by the Audio Solution Pack to read and write metadata for audio files. Islandora currently uses ExifTool version 3.5.

ExifTool can be obtained from the project home page: http://www.sno.phy.queensu.ca/%7Ephil/exiftool/. Installation will differ depending on your operating system; please see the installation instructions for details.

ImageMagick

ImageMagick is used to create thumbnail and medium-size JPEF versions of ingested images. Most Linux systems ship with ImageMagick already installed. If your system does not have the command-line program "convert" you will need to install ImageMagick using your OS's package manager, or on Mac OS X install it from source using http://www.macports.org. JPEG2000 support is not required in ImageMagick since we are using Kakadu.

We recommend using ImageMagick version 6.7 and higher and the corresponding Ghostscript installation. You can download ImageMagick (and find installation instructions) on the Binary Release page: http://www.imagemagick.org/script/binary-releases.php.

Islandora Viewer

The Islandora Viewer is a JAVA application served up by Tomcat, which enables the viewing of web-based images and PDFs as a virtual document. It is required by the Large Image Solution Pack, Newspaper Solution Pack, and Book Solution Pack. The viewer provides zoom, scrolling and pagination capabilities.

Installation Steps:

Download the latest version of IIV from the Release Notes and Downloads page and extract the WAR files to the $CATALINA_HOME/webapps directory.
Tomcat should automatically deploy the WAR file, but if it doesn't you can manually unzip the iiv folder from the war file using the "unzip" Unix command.

You must enter the actual IP address of your server in the Fedora URL fields under Administer > Site Configuration > Islandora Configure. Entering 'localhost' or '127.0.0.1' will prevent the viewer from working.

Test the installation by ingesting Fedora service objects in etc/fedora-objects.

If you wish to use an iFrame with the Islandora Viewer, you must configure Apache with a reverse proxy. See Apache Reverse Proxy.

If you are using Fedora 3.5, you will need to edit FEDORA_HOME/server/config/spring/web/security.xml and comment out the following line: <security:filter-chain pattern="/risearch*" filters="ChannelFilter,AuthFilterJAAS" />.

Lame

Lame is distributed as source code only and requires the ability to use a C compiler. To learn more about the LAME MP3 Encoder, visit http://lame.sourceforge.net/

Install using the following command (for Ubuntu)

```
sudo apt-get install lame
```

Kakadu

Source: Kakadu Software - Description

A comprehensive, heavily optimized, fully compliant software toolkit for JPEG2000 developers. Now with more extensive and convenient support for Java native interfaces. Also now automatically builds bindings for C# and Visual Basic programmers.

Kakadu supports multi-threaded processing to fully utilize parallel processing resources (multiple CPUs, multi-core CPUs or hyperthreading). You can select the single-threaded processing model from v5.0 and before, or a new multi-threaded processing model (requires only a few extra lines of code in your application).

Kakadu provides a carefully engineered thread scheduler so once you have created a multi-threaded environment and populated it with one thread for each physical/virtual processor on your system, close to 100% utilization of all computational resources is typically achieved.

Kakadu is a complete implementation of the JPEG2000 standard, Part 1, -- i.e., ISO/IEC 15444-1. This new image compression standard is substantially more complex than the existing JPEG standard, both from a computational and a conceptual perspective.

Kakadu also provides a comprehensive implementation for several of the most useful features from Part 2 of the JPEG2000 standard, including general multi-component transforms and arbitrary wavelet transform kernels.

The Kakadu software framework provides a solid foundation for a range of commercial and non-commercial applications. By making a consistent and efficient implementation of the standard widely available for both academic and commercial applications, the goal is to encourage the widespread adoption of JPEG2000.

The Kakadu software toolkit is required if you wish to automatically convert TIFF files into JPEG2000 during the ingest object process.

Installation Steps:

Download the Kakadu package from the project download page on http://www.kakadusoftware.com/ and follow the installation instructions for your platform. You should be able to run "kdu_compress" from the command line and successfully convert a TIFF file to a JPEG2000.

- Djatoka includes Kakadu executable files. If Drupal and Fedora are on the same server you should be able to copy or symlink kdu_compress to somewhere on the Apache users classpath.

SWF Tools

This suite of Drupal modules is used primarily by the Audio Solution Pack. You can download if from the Drupal project page.

SWF Tools version 6.x-3.0-beta5 or higher is required.

JW Player is included in this package, but you need to download and install the library separately. You can get the latest JW Player library at http://www.longtailvideo.com/players/jw-flv-player/.

1. Install to /sites/all/libraries/mediaplayer-*x.x* (the name of the directory will depend on the version you install).
2. Create a symlink to this directory in sites/all/modules called **mediaplayer4**.

Finally, enable the following SWF Tools modules:

- SWF Tools
- JW Media Player 5
- SWF Object 2

Tesseract

Tesseract is an OCR engine that was developed at HP Labs between 1985 and 1995 - it is currently being developed at Google. Recognized as one of the most accurate open source OCR engines available, Tesseract will read binary, grey, or colour images and output text. A TIFF reader that will read uncompressed TIFF images is also included.

Islandora currently uses Tesseract version 3.0, which can be obtained from http://code.google.com/p/tesseract-ocr/. Installation will differ depending on your operating system; please see the http://code.google.com/p/tesseract-ocr/wiki/ReadMe for detailed instructions.

Automatic page orientation detection was added in Tesseract 3.01. For more information, see the Tesseract Project Home.

Video Dependencies

A number of video conversion dependencies are required by the Video Solution Pack. Most of the conversion work is handled by FFmpeg, a complete, cross-platform solution to record, convert and stream audio and video. However, conversion to the Ogg Vorbis format requires ffmpeg2theora, libvorbis, libtheora, and libogg.

The following command will install FFmpeg, libvorbis, libtheora, and libogg on CentOS:

```
yum -y install ffmpeg libtheora libvorbis libogg
```

The process is slightly different on Ubuntu; support for the AAC codec has been removed from the version of FFmpeg available from the standard Ubuntu repositories. Instead, you can install from a third-party repository using these steps:

1. Install, update, and authenticate the Mediabuntu repository

```
sudo wget http://www.medibuntu.org/sources.list.d/$(lsb_release -cs).list --output-document=/etc/apt/sources.
list.d/medibuntu.list && sudo apt-get -q update && sudo apt-get --yes -q --allow-unauthenticated install
medibuntu-keyring && sudo apt-get -q update
```

2. Install FFmpeg

Ubuntu Oneiric Ocelot 11.10:

```
sudo apt-get install ffmpeg libavcodec-extra-53
```

Ubuntu Natty Narwhal 11.04, Ubuntu Maverick Meerkat 10.10, and Ubuntu Lucid Lynx 10.04:

```
sudo apt-get install ffmpeg libavcodec-extra-52
```

Ubuntu Hardy Heron 8.04:

```
sudo apt-get install ffmpeg
```

You can then use **wget** to install ffmpeg2theora. The command, which works on both CentOS and Ubuntu, will vary depending on the version and installation path; here is an example of the command we use to download and install the 64bit Linux version:

```
wget http://v2v.cc/~j/ffmpeg2theora/ffmpeg2theora-x.xx.linux64.bin
chmod +x ffmpeg2theora-x.xx.linux64.bin
ln -s /opt/ffmpeg2theora-x.xx.linux64.bin /usr/bin/ffmpeg2theora
```