Apiary Project Technical Documentation

Apiary Project Technical Documentation

Table of Contents

Introduction

What is the Apiary Project?

The Apiary Project is a fundamental research project with the goal of identifying how human intelligence can be combined with machine processes for effective and efficient transformation of textual museum specimen label information into high-quality machine-processible parsed data.

Chapter 1. Hardware and Software Requirements

Servers, OS requirements, software components and dependencies

Apiary Project Workflow and Image Server

Hardware

1. Dual to Quad core 64 bit processor
2. 4-16GB RAM
3. Sizeable storage space, we used a 100GB drive for original images and another 100GB for converted images

Software

Operating System

1. Ubuntu 8.04 64-bit or later, primarily 10.04 Desktop

Ubuntu aptitude packages

Applications

1. Apache 2
2. Mysql-server
3. PHP 5
4. Sun-java6-jdk
5. Subversion
6. curl
7. mercurial
8. scons
9. chkconfig
10. vim-nox
11. flashplugin-installer
12. varnish (*optional)

Libraries

1. libapache2-mod-auth-mysql
2. Php5-mysql

3. Php5-curl
4. Php5-xsl
5. Php5-imagick
6. Php5-gd
7. Php5-dev
8. Php-soap
9. libjpeg-progs

Additional required applications

1. Fedora-commons digital repository 3.2
2. Tomcat 5.5
3. Adore-Djatoka JPEG2000 Image Server
4. Apache-SOLR 1.4.1
5. Drupal content management 6.22
6. Islandora (fedora repository Drupal module)
7. OCRopus 0.4.1
8. GOCR
9. Ocrad

Apiary Project Image Processing and Conversion Server

Hardware

1. Dual to Quad core 64 bit processor
2. 4+GB RAM
3. Sizeable storage space. Note: We connect to a shared image drive from the Apiary Project Server.

Software

Operating System

1. Ubuntu 10.04 64-bit server

Ubuntu aptitude packages

Applications

1. ImageMagick

2. OpenJPEG

Libraries

1. build-essential
2. libpng12-dev
3. libjpeg62-dev
4. libtiff4-dev
5. libjpeg-progs
6. libopenjpeg-dev

Additional required applications

1. OpenJPEG version 1.4 sources

Chapter 2. Installation and configuration documentation

A step by step guide to completly installing the Apiary Project workflow

Install Ubuntu Desktop 10.04 [http://releases.ubuntu.com/lucid/]

Note: The installation of the OS is outside the scope of this documentation, it is listed here to note the starting point of the installation

1. username: apiary
2. full name: password is apiary
3. password: apiary

   Note: These documents will assume all passwords to be 'apiary'

Go to Synaptics and install

Note: These can also be installed using aptitude from the command, i.e. apt-get install {package}

1. Apache 2 (and its automatically added required packages)
2. Mysql-server

   Note: These documents assume the root password is apiary

3. PHP 5 (and its automatically added required packages)
4. Sun-java6-jdk (and its automatically added required packages)
5. Subversion
6. curl
7. mercurial
8. scons
9. chkconfig
10. libapache2-mod-auth-mysql
11. flashplugin-installer
12. Php5-mysql
13. Php5-curl
14. Php5-xsl
15. Php5-imagick
16. Php5-gd

17.Php5-dev

18.Php-soap

19.libjpeg-progs

20.vim-nox (this fixes the weird behavior vi has with Ubuntu)

Configure server hostname

```
vi /etc/hostname
```

change hostname

Enable Apache Modules

```
a2enmod rewrite

a2enmod auth_mysqlvi

/etc/init.d/apache2 restart
```

Configure PHP

```
vi /etc/php5/apache2/php.ini
```

Replace ‘memory_limit = 16M’ with ‘memory_limit = 256M’

```
/etc/init.d/apache2 restart
```

Create Mysql Drupal and Fedora users and databases

```
mysql -u root -p
```

Accesses mysql command line, password is apiary

Create Drupal user and database

```
create database drupal;

grant all on drupal.* to ‘drupalAdmin’@’localhost’ identified by
‘apiary’;

grant all on drupal.* to ‘drupalAdmin’@’%’ identified by ‘apiary’;
```

Create Fedora user and database

```
create database fedora;

grant all on drupal.* to ‘fedoraAdmin’@’localhost’ identified by
‘apiary’;

grant all on drupal.* to ‘fedoraAdmin’@’%’ identified by ‘apiary’;
```

Leave MySQL

```
exit;
```

Install Drupal [http://drupal.org/project/drupal]

Download

```
wget http://ftp.drupal.org/files/projects/drupal-6.22.tar.gz
```

Extract drupal-6.22

```
tar -zxvf drupal-6.22.tar.gz
```

Move drupal-6.22 to /var/www/

```
mv drupal-6.22 /var/www
```

Create symbolic link to drupal dir

Note: This preserves versioning as drupal updates are released

```
ln -s /var/www/drupal-6.22 /var/www/drupal
```

Edit Drupal .htaccess

```
vi /var/www/drupal/.htaccess
```

uncomment REWRITEs for '/drupal'

Create writeable Drupal files directory outside the scope of the main .htaccess

```
mkdir /var/www/drupal/sites/default/files
```

```
chmod a+w /var/www/drupal/sites/default/files
```

```
vi /var/www/drupal/sites/default/files/.htaccess
```

Save blank file!

Edit Apache2 sites

```
vi /etc/apache2/sites-enabled/000-default
```

change AllowOverride None to AllowOverride All for / and /var/www directories

Create writeable settings file for Drupal install

```
cp   /var/www/drupal/sites/default/default.settings.php   /var/www/
drupal/sites/default/settings.php
```

```
chmod a+w /var/www/drupal/sites/default/settings.php
```

Install Drupal via web browser

Go to http://hostname/drupal

Click Install Drupal in English

Enter values

Database name: drupal

Database username: drupalAdmin

Database password: apiary

Click Save and continue

Configure site

Site information

Site name: hostname

Site e-mail address: apiary@hostname.org

Administrator Account

Username: apiary

E-mail address: apiary@hostname.org

Password: apiary

Site e-mail address: apiary@hostname.org

Click Save and continue

Remove write permission on settings file

```
Chmod a-w /var/www/drupal/sites/default/settings.php
```

Setup Environment Variables

Locate java home directory

```
locate /rt.jar
```

Create symbolic link for java-home

Note: This preserves versioning

```
ln -s /usr/lib/jvm/java-6-sun-1.6.0.15 /usr/lib/jvm/java-home
```

Edit user profile

```
vi /etc/profile
```

Add the following lines (Note: no spaces around equal signs!)

export FEDORA_HOME=/usr/local/fedora

export JAVA_HOME=/usr/lib/jvm/java-home

export PATH=$JAVA_HOME/bin:$PATH

export PATH=$FEDORA_HOME/server/bin:$PATH

export PATH=$FEDORA_HOME/client/bin:$PATH

export CATALINA_HOME=$FEDORA_HOME/tomcat

export LD_LIBRARY_PATH=/usr/lib/jvm/java-home

export LD_LIBRARY_PATH=/usr/local/lib:$LD_LIBRARY_PATH

Restart machine

Install Fedora [http://www.fedora-commons.org/]

Download

```
wget        http://downloads.sourceforge.net/fedora-commons/fedora-
installer-3.2.1.jar
```

Create symbolic link to fedora dir

Note: This preserves versioning

```
ln -s /usr/local/fedora-3.2 /usr/local/fedora
```

Create a keystore for SSL support

```
$JAVA_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA
```

Password:apiary

Run Fedora java installer

```
java -jar fedora- installer-3.2.1.jar
```

Installation type: custom

Fedora home directory: /usr/local/fedora-3.2

Fedora administrator password: apiary

Fedora server host: hostname

Fedora application server context: fedora

Authentication requirement for API-A: false

SSL availability: true

SSL required for API-A: false

SSL required for API-M: false

Servlet engine: included

Tomcat home directory: /usr/local/fedora-3.2/tomcat

Tomcat HTTP port: 8080

Tomcat shutdown port: 8005

Tomcat Secure HTTP port: 8443

Keystore file: default

Keystore password: apiary

Keystore type: JKS

Database: mysql

MySQL JDBC driver: included

Database username: fedoraAdmin

Database password: apiary

JDBC URL: jdbc:mysql://hostname/fedora?useUnicode=true&characterEncoding=UTF-8&autoReconnect=true

JDBC DriverClass: com.mysql.jdbc.Driver

Policy enforcement enabled: true

Enable Resource Index: true

Enable Messaging: false

Deploy local services and demos: true

Verify Fedora's Resource Index activation

Note: Some Ubuntu installs had this variable set to zero even using "Enable Resource Index: true"

```
vi $FEDORA_HOME/server/config/fedora.fcfg
```

Set fedora.server.resourceIndex.ResourceIndex to 1

Startup Tomcat (which will install Tomcat Fedora features)

```
$FEDORA_HOME/tomcat/bin/startup.sh
```

Shutdown Tomcat

```
$FEDORA_HOME/tomcat/bin/shutdown.sh
```

Copy crossdomain file

```
cp $FEDORA_HOME/tomcat/webapps/fedora/admin/crossdomain.xml /usr/
local/fedora/tomcat/webapps
```

Setup anonymous user access

Note: Allows anonymous users to be able to view public objects later while using Islandora.

```
vi $FEDORA_HOME/server/config/fedora-users.xml
```

Note: Islandora will send the username of fedora_anonymous and password of anonymous for all unauthenticated use.

Add the following <user> lines:

<user name="fedora_anonymous" password="anonymous">

<attribute name="fedoraRole">

<value>fedoraUser</value>

</attribute>

</user>

Install Islandora Drupal filter

Download DrupalFilter_3.jar from http://sourceforge.net/projects/islandora/files/

```
wget http://softlayer.dl.sourceforge.net/project/islandora/
islandora/Islandora_Dru6_Fed_3.1_beta2009-04-03/DrupalFilter_3.jar
```

Move Drupal filter to the fedora tomcat webapp

```
cp DrupalFilter_3.jar $FEDORA_HOME/tomcat/webapps/fedora/WEB-INF/
lib/DrupalFilter.jar
```

Create a fedora server filter-drupal file

```
vi $FEDORA_HOME/server/config/filter-drupal.xml
```

Caution

The xml file needs a blank line at the end

Add the following lines:

<?xml version="1.0" encoding="UTF-8"?> <!--File to hold drupal connection info for the FilterDrupal servlet filter. For multisite drupal installs you can include multiple connection elements. We will

query all the databases and assume any user in any drupal db with the same username and password are the same user. We will gather all roles for that user from all databases. This is a potential security risk if a user in one drupal db has the same username and password as another user in a seperate drupaldb. We are also assuming all drupal dbs to be mysql. This file should be located in the same directory as the fedora.cfcg file--><FilterDrupal_Connection> <connection server="hostname" dbname="drupal" user="drupalAdmin" password="apiary" port="3306"> <sql> SELECT distinct u.uid as userid, u.name as Name, u.pass as Pass, r.name as role FROM drupal.users u, drupal.role r, drupal.users_roles where u.name=? and u.pass=? and r.rid=drupal.users_roles.rid and u.uid=drupal.users_roles.uid; </sql> </connection> </FilterDrupal_Connection>

Edit the fedora tomcat webapp web.xml file

```
vi $FEDORA_HOME/tomcat/webapps/fedora/WEB-INF/web.xml
```

After

<filter>

<filter-name>XmlUserfileFilter</filter-name>

<filter-class>fedora.server.security.servletfilters.xmluserfile.FilterXmlUserfile</filter-class>

</filter>

Add

<filter>

<filter-name>DrupalFilter</filter-name>

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

</filter>

After

<filter-mapping>

<filter-name>XmlUserfileFilter</filter-name>

<url-pattern>/*</url-pattern>

</filter-mapping>

Add

<filter-mapping>

<filter-name>DrupalFilter</filter-name>

<url-pattern>/*</url-pattern>

</filter-mapping>

Enable and rebuild Fedora’s Resource Index

Enable Fedora’s Resource Index activation

Note: Some Ubuntu installs had this variable set to zero even using “Enable Resource Index: true”

```
vi $FEDORA_HOME/server/config/fedora.fcfg
```

Set fedora.server.resourceIndex.ResourceIndex to 1

Rebuild Fedora's Resource Index activation

```
$FEDORA_HOME/server/bin/fedora-rebuild.sh
```

Choose "Rebuild the Resource Index"

Verify selection

Restart Fedora

```
$FEDORA_HOME/tomcat/bin/shutdown.sh
```

```
$FEDORA_HOME/tomcat/bin/startup.sh
```

Install Adore-Djatoka Image Server [http://sourceforge.net/projects/djatoka/]

Download

```
wget    http://iweb.dl.sourceforge.net/project/djatoka/djatoka/1.1/
adore-djatoka-1.1.tar.gz
```

Extract adore-djatoka-1.1

```
tar -zxvf adore-djatoka-1.1.tar.gz
```

Move adore-djatoka-1.1 to /usr/local

```
mv adore-djatoka-x.x /usr/local
```

Create symbolic link to adore-djatoka dir

Note: This preserves versioning

```
ln -s /usr/local/adore-djatoka-1.1 /usr/local/adore-djatoka
```

Test functionality

```
cd /usr/local/adore-djatoka/bin
```

```
./compress.sh -i ../etc/test.jpg -o ../etc/test.jp2
```

Compress jpg file into jpeg2000 format

```
./extract.sh -i ../etc/test.jp2 -o ../etc/test-size1.jpg -l 1
```

extracts a jpg file from a jpeg2000 file

```
ls -l ../etc
```

verify test.jp2 and test-size1.jpg files were created

Add Adore-Djatoka to tomcat webapps

Shutdown running tomcat

```
$FEDORA_HOME/tomcat/bin/startup.sh
```

Copy adore-djatoka.war file to tomcat webapps

```
cp /usr/local/adore-djatoka/dist/adore-djatoka.war /$CATALINA_HOME/
webapps
```

Start tomcat from Adore-Djatoka bin directory

Note: Must be started while currently in the /usr/local/adore-djatoka/bin directory

```
cd /usr/local/adore-djatoka/bin
```

```
./tomcat.sh start
```

Configure Referent Resolver

Edit Djatoka Properties

```
vi            $CATALINA_HOME/webapps/adore-djatoka/WEB-INF/classes/
djatoka.properties
```

Add the following in the Referent Resolver Properties section

SimpleListResolver.maxRemoteCacheSize=10000

Setup JP2 images

Create Image directory

```
mkdir -p /var/www/images/jpeg2000
```

Download JP2 images

```
cd /var/www/images/jpeg2000
```

```
wget      http://research.apiaryproject.org/images/apiary-aquarius-
jpeg2000-vbox.tar.gz
```

Extract images

```
tar –zxvf apiary-aquarius-jpeg2000-vbox.tar.gz
```

Edit simple resolver list location

```
vi /var/www/images/jpeg2000/imgIndex.txt
```

Replace all /mnt/converted_images with /var/www/images

Move list to appropriate location

```
mv   /var/www/images/jpeg2000/imgIndex.txt   $CATALINA_HOME/webapps/
adore-djatoka/WEB-INF/classes
```

Stop and Start Tomcat from adore-djatoka bin directory

```
cd /usr/local/adore-djatoka/bin

./tomcat.sh stop

./tomcat.sh start
```

Install OCRopus [http://code.google.com/p/ocropus/]

Download

```
mkdir /home/apiary/ocropus

cd /home/apiary/ocropus

hg clone http://iulib.googlecode.com/hg iulib

hg clone http://ocropus.googlecode.com/hg ocropus
```

Install prerequisites

```
sh -x ocropus/ubuntu-packages
```

Install iulib library

```
cd /home/apiary/ocropus/iulib

scons

scons install
```

Install OCRopus library

```
cd /home/apiary/ocropus/ocropus

scons

scons install
```

Configure Ubuntu for OCRopus

Add /usr/local/bin to PATH

```
PATH=/usr/local/bin:$PATH
```

Add library reference

```
ldconfig
```

```
LD_LIBRARY_PATH=/usr/local/lib:$LD_LIBRARY_PATH
```

Test install

```
ocropus page data/testimages/simple.png
```

Note: should output the ocr of image simple.png

Install Ocrad [http://www.gnu.org/software/ocrad/]

Download

```
mkdir /home/apiary/ocrad
```

```
cd /home/apiary/ocrad
```

```
wget http://ftp.gnu.org/gnu/ocrad/ocrad-0.21.tar.gz
```

Extract ocrad-0.21

```
tar -zxvf ocrad-0.21.tar.gz
```

Configure and make ocrad

```
cd /home/apiary/ocrad/ocrad-0.21
```

```
./configure
```

```
make
```

```
make installd
```

Note: adds ocrad file to /usr/local/bin

Install GOCR [http://jocr.sourceforge.net/]

Download

```
mkdir /home/apiary/gocr
```

```
cd /home/apiary/gocr
```

```
wget http://www-e.uni-magdeburg.de/jschulen/ocr/gocr-0.49.tar.gz
```

Extract ocrad-0.21

```
tar -zxvf gocr-0.49.tar.gz
```

Configure and make ocrad

```
cd /home/apiary/gocr/gocr-0.49
```

```
./configure
```

```
make

make install
```

Note: adds gocr file to /usr/local/bin

Configure Drupal to work with Apiary using fedora_repository (Islandora) and apiary_project

Allow user access for druapl ajax calls

Edit drupal/includes/menu.inc

```
/var/www/drupal/includes/menu.inc
```

In function _menu_check_access(&$item, $map)

Replace:

if ($callback == 'user_access') { $item['access'] = (count($arguments) == 1) ? user_access($arguments[0]) : user_access($arguments[0], $arguments[1]); }

With:

if ($callback == 'user_access') { if(strpos($item['file'], "modules/apiary_project") > -1) { $item['access'] = user_access("View Apiary Project"); } else { $item['access'] = (count($arguments) == 1) ? user_access($arguments[0]) : user_access($arguments[0], $arguments[1]); } }

Enable Clean URLs

Browse to Administer->Site configuration->Clean URLs

Select Enabled

Click Save Configuration

Install Required Drupal Themes

Install Zen [http://drupal.org/project/zen]

Download

```
cd /var/www/drupal/themes

svn co svn://projects.brit.org/src/svn_apiary/trunk/servers/
applications/drupal/themes/zen
```

Install

Browse to http://hostname/drupal/admin/build/themes

Scroll to Zen

Check Enabled

Click Save configuration

Install Cti-flex-no-title

Note: This is theme altered by the Apiary Project to remove all menus, padding, etc

Download

```
cd /var/www/drupal/themes

svn co svn://projects.brit.org/src/svn_apiary/trunk/servers/
applications/drupal/themes/cti_flex_no_title
```

Install

Browse to http://hostname/drupal/admin/build/themes

Scroll to cti-flex-no-title

Check Enabled

Click Save configuration

Install Cti-flex [http://drupal.org/project/cti_flex]

Download

```
cd /var/www/drupal/themes

svn co svn://projects.brit.org/src/svn_apiary/trunk/servers/
applications/drupal/themes/cti_flex
```

Install

Browse to http://hostname/drupal/admin/build/themes

Scroll to cti-flex

Check Enabled

Click Save configuration

Install Required Drupal Modules

Install ImageAPI [http://drupal.org/project/imageapi]

Download

```
cd /home/apiary/

wget http://ftp.drupal.org/files/projects/imageapi-6.x-1.10.tar.gz
```

Extract

```
tar -zxvf imageapi-6.x-1.10.tar.gz
```

Move

```
mv imageapi-6.x-1.10 /var/www/drupal/modules
```

Install

Browse to http://hostname/drupal/admin/build/modules

Scroll to ImageCache section

Check Enabled for ImageAPI

Click Save configuration

Install ImageAPI GD2

Download

Note: There is no download as it is already included

Install

Browse to http://hostname/drupal/admin/build/modules

Scroll to ImageCache section

Check Enabled for ImageAPI GD2

Click Save configuration

Install Core-optional Modules

Download

Note: There is no download as it is already included with the drupal core

Install

Browse to http://hostname/drupal/admin/build/modules

Scroll to Core-optional section

Check Enabled for Path

Check Enabled for PHP filter

Click Save configuration

Install Fedora_Repository (ISLANDORA) [http://islandora.ca/]

Download

```
cd /var/www/drupal/modules
```

```
svn co http://fedora-commons.org/svn/root/islandora/islandora-module/Islandora-dru6-fed3/trunk/fedora_repository
```

Install

Browse to http://hostname/drupal/admin/build/modules

Scroll to Fedora Repository section

Check Enabled for Digital Repository

Check Enabled for Fedora ImageAPI

Click Save configuration

Create Drupal administrator role

Browse to http://hostname/drupal/admin/user/roles

type ‘administrator’ and click Add role

Edit Drupal role administrator’s permissions

Browse to http://hostname/drupal/admin/user/roles

Click edit permissions for administrator

Check all options for fedora_repository module

Click Save permissions

Configure Fedora Repository settings

Browse to http://hostname/drupal/admin/settings/fedora_repository

Default Collection Name: Apiary's Fedora Repository

Default Collection PID: apiary:SpecimenBinders

A user with the Drupal role administrator: apiary

Pid namespaces allowed in this Drupal install: demo: changeme: Islandora: ilives: apiary: ap-specimen: ap-roi: ap-image: ap-model: ap-sdef: ap-sdep: ap-sdefcm:

Fedora Soap Management Url: http://hostname:8080/fedora/services/management?wsdl

Fedora base url: http://hostname:8080/fedora

Fedora RISearch URL: http://hostname:8080/fedora/risearch

Fedora Lucene Search URL: http://hostname:8080/fedoragsearch/rest

Fedora Lucene Index Name: BasicIndex

Fedora Soap Url: http://hostname:8080/fedora/services/access?wsdl

Click Save configuration

Install Apiary Project [http://www.apiaryproject.org]

Download

```
cd /var/www/drupal/modules
```

```
svn co svn://projects.brit.org/src/svn_apiary/branches/mellifera-apiary-1.0.0/apiary_project
```

Create Drupal apiary admin role

Browse to http://hostname/drupal/admin/user/roles

type ‘apiary admin’ and click Add role

Edit Drupal role apiary admin's permissions

Browse to http://hostname/drupal/admin/user/roles

Click edit permissions for administrator

Check all options for apiary_project module

Click Save permissions

Add Drupal roles apiary admin and administrator to Drupal User apiary

Browse to http://hostname/drupal/admin/user/user

Under Operations, click edit for apiary

Scroll to Roles section

Select apiary admin

Select administrator

Click Save

Install

Create Writeable Directories

```
chmod 777 /var/www/drupal/modules/apiary_project/workflow/
templates_c

mkdir /var/www/drupal/sites/default/files/apiary_datastreams

chmod 777 /var/www/drupal/sites/default/files/apiary_datastreams
```

Edit Apiary Project Fedora Database Connector

```
vi /var/www/drupal/modules/apiary_project/fedora_commons/
config_fedora.inc
```

Set database host, username and password as specified in Create Fedora User and Database

Note: This allows for direct reading of the fedora database instead of querying fedora itself through CURL.

Queries like get all pids with a name like ap-specimen is much easy to do from the database, which is done in one line, vs through SPRQL or Resource Index Fedora request.

At one point, we discussed the merits of storing all data for the workflow here but instead went the way of modifying the drupal db.

Edit Externally Used Workflow files

```
vi /var/www/drupal/modules/apiary_project/workflow/index.php
```

Set $drupal_url = "http://hostname/drupal";

Set $djatoka_url = "http://hostname:8080/";

```
vi /var/www/drupal/modules/apiary_project/workflow/comparer.php
```

Set $drupal_url = "http://hostname/drupal";

```
vi /var/www/drupal/modules/apiary_project/workflow/search.php
```

Set $drupal_url = "http://hostname/drupal";

```
vi /var/www/drupal/modules/apiary_project/workflow/workflow.php
```

Set $drupal_url = "http://hostname/drupal";

Browse to http://hostname/drupal/admin/build/modules

Scroll to Apiary Project section

Check Enabled for Apiary Research Project

Click Save configuration

Configure Apiary Project settings

Browse to http://hostname/drupal/admin/settings/apiary_project

Enter variable information

Click Save configuration

Caution

Do this even if no changes are made.

Configure Apiary Project System Startup Script

```
cp apiary_project/apiary_project.sh /etc/init.d/apiary_project

update-rc.d apiary_project defaults

chmod a+rx /etc/init.d/apiary_project

chmod 777 /var/www/drupal/modules/apiary_project/workflow/
templates_c
```

Install ThemeKey [http://drupal.org/project/themekey]

Download

```
cd /home/apiary/

wget http://ftp.drupal.org/files/projects/themekey-6.x-3.3.tar.gz
```

Extract

```
tar -zxvf themekey-6.x-3.3.tar.gz
```

Move

```
mv themekey-6.x-3.3 /var/www/drupal/modules
```

Install

Browse to http://hostname/drupal/admin/build/modules

Scroll to ThemeKey section

Check Enabled for ThemeKey

Click Save configuration

Browse to http://hostname/drupal/admin/settings/themekey

Create New themekey Rule

drupal:path = apiary use CTI Flex theme No Title

check Enabled

Click Save configuration

Install Apache-Solr [http://www.apache.org/dyn/closer.cgi/lucene/solr/]

Download

```
cd /home/apiary

wget       http://apache.tradebit.com/pub/lucene/solr/1.4.1/apache-solr-1.4.1.tgz
```

Extract apache-solr-1.4.1

```
tar -zxvf apache-solr-1.4.1.tar.gz
```

Move apache-solr-1.4.1

```
mv apache-solr-1.4.1 $FEDORA_HOME/tomcat/webapps
```

Copy Apiary Solr Schema

```
cp apiary_project/solr/conf/schema.xml $FEDORA_HOME/tomcat/webapps/apache-solr-1.4.1/example/solr/conf/schema.xml
```

Configure Solr System Startup Script

```
cp apiary_project/solr/solr.sh /etc/init.d/solr

update-rc.d solr defaults

chmod a+rx /etc/init.d/solr
```

Ingest Required Fedora Digital Objects

Ingest SpecimenBinders

```
$FEDORA_HOME/client/bin/fedora-admin.sh
```

Login

Username: fedoraAdmin

Password: apiary

Click File->Ingest->One Object->From File…

Navigate to /var/www/drupal/modules/apiary_project/digital_objects/required/apiary_SpecimenBinders.xml

Click Open

Select FOXML version 1.1

Click OK

dynamicOCR service definition

```
cp    /var/www/drupal/modules/apiary_project/digital_objects/sDefs/ocropus/dynamicOCR.php /var/www/drupal/sites/default/files/

$FEDORA_HOME/client/bin/fedora-admin.sh
```

Login

Username: fedoraAdmin

Click File->Ingest->One Object->From File…

Navigate to each of the following separately:

• /var/www/drupal/modules/apiary_project/digital_objects/sDefs/ocropus/ap-sdef_ocropus.xml

• /var/www/drupal/modules/apiary_project/digital_objects/sDefs/ocropus/ap-sdefcm_ocropus.xml

• /var/www/drupal/modules/apiary_project/digital_objects/sDefs/ocropus/ap-sdep_ocropus.xml

Click Open

Select FOXML version 1.1

Click OK

Click File->Exit

Create Apiary Project Drupal items

Create Apiary Pages

homepage

Browse to http://hostname/drupal/node/add/page

Title: Home Page

Menu settings

Menu link title:

Parent item: Primary links

Weight: 0

Body:

<h1>Welcome to the Apiary Project Proof of Concept</h1> <?php if (user_is_logged_in()): ?> To begin the demo, select Workspace in the navigation menu above. <?php else:?> To begin the demo, login with username "demo" and password "demo" then select Workspace in the navigation menu above. <?php endif;?>

Input Format: PHP code

URL path settings: homepage

Set Homepage

Browse to http://hostname/drupal/admin/settings/site-information

Default front page: homepage

Click Save configuration

Set Menus

Remove menus

Browse to http://hostname/drupal/admin/build/block

Set all regions to <none>

Click Save blocks

Configure Drupal Theme for Apiary site

Copy cti_flex_logo.png to files directory

```
cp                    /var/www/drupal/modules/apiary_project/images/
cti_flex_logo.png /var/www/drupal/sites/default/files
```

Configure Zen

Browse to http://hostname/drupal/admin/build/themes

Click configure next to Zen

Scroll to Logo image settings

Uncheck Use the default logo

Click Save configuration

Configure CTI Flex Theme

Browse to http://hostname/drupal/admin/build/themes

Click configure next to CTI Flex theme

Scroll toToggle display

Uncheck Site name

Scroll to Logo image settings

Uncheck Use the default logo

Path to custom logo: sites/default/files/cti_flex_logo.png

Scroll to Theme-specific settings ->Custom color settings

Select color for body background: #FFE8B7

Select color for header and footer backgrounds: #8EB7FE

Select color for main navigation bar and block header backgrounds: #3F4F6B

Select color for block content background: #D4DAE6

Click Save configuration

Configure Drupal Primary links

Browse to http://hostname/drupal/admin/build/menu-customize/primary-links

Home

Click Add item tab

Path: http://hostname/drupal

Menu link title: Home

Description: home page

Parent item: <Primary links>

Weight: 0

Click Save

Worksplace

Note: This link is now added when the module is installed

Weight: 0

Click Save

Login

Click Add item tab

Path: user/login

Menu link title: Login

Description:

Parent item: <Primary links>

Weight: 2

Click Save

Logout

Click Add item tab

Path: logout

Menu link title: Logout

Description:

Parent item: <Primary links>

Weight: 2

Click Save

About us

Click Add item tab

Path: http://hostname/drupal/apiary?ref=about

Menu link title: About us

Description:

Parent item: <Primary links>

Weight: 3

Click Save

Configure Drupal Navigation menu

Add/remove Navigation menu links

Browse to http://hostname/drupal/admin/build/menu-customize/navigation

Uncheck the following items marked Enabled

Digital Repository

Create content->Fedora Repository

Create content->Story

Click Save configuration

Remove Navigation menu from the left panel for the workflow

Browse to http://hostname/drupal/admin/build/block

Click configure for Navigation

Scroll to Page specific visibility settings

Select Show only on the listed pages

Type ‘homepage’ in the Pages: section

Click Save block

Start Apiary

```
/etc/init.d/solr start
```

```
/etc/init.d/apiary_project start
```

Ingest Apiary Demo Objects

Browse to http://hostname/drupal/apiary/admin

Click Batch Create Specimens

Max Specimens:

File name source file: http://hostname/images/jpeg2000/demo_filenames.txt

Referent ID: apiary:jpeg2000

JPEG2000 Url Base: http://hostname/images/jpeg2000

Source Url Base: http://hostname/images/original

Click Submit

Chapter 3. Using the Apiary Project Module

Preface: Chapter 2 entails a lot, so if you have objects ingested then great! It's time to harness the power of the Apiary Project.

(Re)Starting The Apiay Project

Anytime a reboot is felt necessary, use the following two commands to restart the project.

Note: start, stop and restart can all be passed in, but restart will not die if either are not started so it is an all-safe command to use regardless.

```
/etc/init.d/solr restart
```

```
/etc/init.d/apiary_project restart
```

Administrating the Apiary Project

Browse to http://hostname/drupal/apiary/admin

Create, Edit and Delete Workflows

Workflow Name

Title Name of the workflow

Workflow Description

Information about the workflow

Workflow Permissions

Permissions to features users of the workflow will be allowed to access.

canAnalyzeSpecimen

Can users assigned to the workflow use Image Analysis features, like creating ROIs.

canTranscribe

Can users assigned to the workflow use Transcribe features.

canParseL1

Can users assigned to the workflow use Parse Level 1 features.

canParseL2

Can users assigned to the workflow use Parse Level 2 features.

canParseL3

Can users assigned to the workflow use Parse Level 3 features.

canQC

Can users assigned to the workflow use Quality Control features.

Workflow Users

Assign Users to this workflow

Select Drupal Apiary Project User

Select one or many or all druapal users who can view the Apiary Project and assign them to the workflow

Create New Drupal Apiary Project User

This feature creates a new drupal user with all the roles and permissions needed to view an Apiary Project workflow and automatically adds that user to the workflow list.

Workflow Strategy

Select Object Pool

Select an Object Pool from list of previous created Specimen Pools.

Create New Object Pool

This feature creates a new object pool based on a Resource Index or Solr Query.

Object Pool Resource Index Query Example

select $sp_pid from <#ri> where $sp_pid<fedora-rels-ext:isMemberOf> <info:fedora/apiary:SpecimenBinders>

Object Pool Solr Query Example

imageMetadata_sourceURL:("http://research.apiaryproject.org/images/original/vol062.tif")+imageMetadata_sourceURL:("http://research.apiaryproject.org/images/original/vol065.tif")&fl=parent_id

Create and Edit GroundTruth

Groundtruth is the expected result of a Specimen after being accurately analyzed, transcibed and parsed. This values do not necessarily associate to any real fedora objects.

Display Specimen Data

Displays the existing Groundtruth datastream for a requested Specimen pid. Can be manually entered or passed as a specimen_pid=ap-specimen:Specimen-X variable

Save Specimen Data

Create or overwrites the Groundtruth datastream for the requested Specimen pid.

Create one or more Specimen from the Apiary Project Research Server

A demonstration of the simplicity of sharing images from the djatoka image server. Any installation can ingest these images.

Select Specimen Images

Select any number of images to ingest.

Assign Specimen Metadata to it to be saved during ingestion

Click Add Specimens

Upon successful ingestion, the Image and Specimen Pids are displayed

Batch Create Specimen Using a Source File

Ingest a number of specimen using the same rft-id base, jp2 url base and source url base. See Ingest Apiary Demo Objects [27]

Select Specimen Images

Select any number of images to ingest.

Assign Specimen Metadata to it to be saved during ingestion

Click Add Specimens

Upon successful ingestion, the Image and Specimen Pids are displayed

Text Comparison

Compare two text strings and see the resulting levenshtein and simple text distances displayed using Daisydiff.

Search Metadata and Object Statuses

ROI specimenMetadata Search

Search using specimenMetadata keywords in throughout all or one specific metadata field.

Image or ROI Status Search

Search for all objects with workflow status and/or specimenMetadata keyword.

Note: A search containing only Image status will return only Image/Specimen pairs.

Re-Index all Fedora Objects into Solr

If changes have been made to Fedora Objects outside the Apiary Project, i.e. directly in Fedora, this useful tool will re-index all Apiary Project objects.

Example Links to Fedora Objects

Example links to:

Main Digital Object overview

View an object's datastream

View a list of datastreams for an object

View a list of service methods (i.e. dynamicOCR) for an object

Chapter 4. Using the Apiary Project Workflow

Preface: Now that we have covered how to create workflows, it's time to use them.

Click Workspace or browse to http://hostname/drupal/apiary

Select Workflow to begin work

Load Workflow Queue

Click + in Southern Pane

Loads the next available Specimen in to the Workflow Queue

Glide Mouse to left to reveal Workflow Queue

Click Add Items to Bring Up the Image Bowser

Select Specimen to load into the Workflow Queue

Click Add to queue

Select Specimen from Workflow Queue

Click Navigation Arrow in Southern Pane

Loads Specimen immediately to the right or left of the current Specimen in the Workflow Queue

Glide Mouse to left to reveal Workflow Queue

Click Specimen from List

View Specimen with Open Layers [http://code.google.com/p/jquery-openlayers/]

Zoom

Use scroll wheel

Use zoom + or - from the Open Layers control panel

Loads Specimen immediately to the right or left of the current Specimen in the Workflow Queue

Pan

Select Hand from Open Layers control panel

Click and drag on the image

Use navigation arrows from the Open Layers control panel

Loads Specimen immediately to the right or left of the current Specimen in the Workflow Queue

Glide Mouse to left to reveal Workflow Queue

Click Specimen from List

Create Regions of Interest (ROIs)

Click Pencil Tool from Open Layers Panel

Click and drag on the Image to highlight a region

Assign Region an ROI Type

Primary Label

Annotation/Other

Barcode

Undefined

Transcribe ROI

Select an ROI to Transcibe

Click Transcibe from the right-hand side Specimen ROI List

Loads ROI immediately and switches to the Transcribe Text Tab

Glide Mouse to left to reveal Workflow Queue

Click Arrow in Specimen to Expand ROIs

Click Transcibe in any ROI

Loads ROI immediately and switches to the Transcribe Text Tab

OCR Results Tab

OCRAD Results

Reprocess OCRAD Results

Reprocesses the ROI through the OCRAD OCR Engine

Copy OCRAD Results to Transcription Textarea

Click the Icon that looks like a Document with a swooshing arrow on it.

OCRopus Results

Reprocess OCRopus Results

Reprocesses the ROI through the OCRopus OCR Engine

Copy OCRopus Results to Transcription Textarea

Click the Icon that looks like a Document with a swooshing arrow on it.

GOCR Results

Reprocess GOCR Results

Reprocesses the ROI through the GOCR OCR Engine

Copy GOCR Results to Transcription Textarea

Click the Icon that looks like a Document with a swooshing arrow on it.

Text Transcription Tab

Enter or correct Verbatim text as transcibed by OCR and human

Click Save text

Click the Save Text Button to save the data to the Text datastream of the ROI Object

Parse ROI

Select an ROI to Parse

Click Parse from the right-hand side Specimen ROI List

Loads ROI immediately and switches to the Parse Text Tab

Glide Mouse to left to reveal Workflow Queue

Click Arrow in Specimen to Expand ROIs

Click Parse in any ROI

Loads ROI immediately and switches to the Parse Text Tab

Image Tab

Displays the ROI Image that was Transcibed

Text Tab

Select a section of text

Assign Selection to Metadata Element

Right Click Selection and Follow the Menus

Use the Top Menu

Remove Selection of Assigned Metadata Element

Right Click Selection and Click Remove

Click Remove from the Top Menu

Find taxa via uBio [http://www.ubio.org/]

Retrieves and displays the uBio results in an overlay

Parse using HERBIS [http://www.herbis.org/]

Runs the Transcibed Text through a HERBIS Natural Language Processing (NLP) algorithms whose return is mapped to Apiary Project specimenMetadata elements.

Save Parse Text

Click the Save Parsed Text Button to save the data to the specimenMetadata datastream of the ROI Object

Note: The Text datastream is also updated to include assign Span tags.

Delete ROI

Click Delete ROI from the right-hand side Specimen ROI List

Remove ROI from Queue

Glide Mouse to left to reveal Workflow Queue

Click remove for the ROI

Remove Specimen from Queue

Glide Mouse to left to reveal Workflow Queue

Click the gear icon

Click Remove specimen

Chapter 5. Troubleshooting the Apiary Project Workflow

Preface: An attempt to cover a variety of known issues

Fedora Will Not Start

Check System Variables

From a command line, typing in $FEDORA_HOME should result in the location of the fedora directory

If it does not, then your variable needs to be set See Edit Environment Variables

Others that could be missing:

JAVA_HOME, CATALINA_HOME, LD_LIBRARY_PATH

Some of these directories use Symlinks that can become invalid after an update, especially JAVA_HOME

Check Tomcat/Catalina logs

```
vi $FEDORA_HOME/server/config/fedora.fcfg
```

Verify hostname is accurate

Verify mysql db and information is accurate

Chapter 6. Apiary Project Advanced Configuration

Other techniques used to improve the Apiary Project

Apache-Solr schema

Edit schema.xml file

```
vi $FEDORA_HOME/tomcat/webapps/apache-solr-1.4.1/example/solr/conf/schema.xml
```

Add Dynamically Indexed Item

Edit the dynamicField section of the XML document

Add Other Solr Features

Solr offers many other features that are beyond the scope of this project. They can be learned about here

Edit search.php file

```
vi /var/www/drupal/modules/apiary_project/workflow/include/search.php
```

Add new code to include added Dynamically Indexed Item

Restart Solr

```
/etc/init.d/solr restart
```

Re-Index Fedora Objects into Solr from http://hostname/drupal/apiary?ref=solr_index_all

Configure Adore-Djatoka Database Resolver

Create and Setup djatoka mysql database

```
mysql -u root -p
```

Accesses mysql command line, password is apiary

Create Drupal user and database

```
create database djatoka;

grant all on djatoka.* to ‘djatokaAdmin’@’localhost’ identified by ‘apiary’;

grant all on djatoka.* to ‘djatokaAdmin’@’%’ identified by ‘apiary’;

use djatoka;

CREATE TABLE `resources` ( `identifier` varchar(150) NOT NULL, `imageFile` varchar(255) NOT NULL, `original_file_url` text NOT
```

```
NULL, `jp2_file_url` text NOT NULL, PRIMARY KEY (`identifier`) )
ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Reconfigure Djatoka properties file

```
vi  /usr/local/fedora/tomcat/webapps/adore-djatoka/WEB-INF/classes/
djatoka.properties
```

Comment out the current OpenURLJP2KService.referentResolverImpl

#OpenURLJP2KService.referentResolverImpl=gov.lanl.adore.djatoka.openurl.SimpleListResolverUncomment:

Uncomment the Database Resolver OpenURLJP2KService.referentResolverImpl

OpenURLJP2KService.referentResolverImpl=gov.lanl.adore.djatoka.openurl.plugin.r
ftdb.DatabaseResolver

Uncomment the Database Resolver properties

DatabaseResolver.url=jdbc:mysql://localhost/djatoka

DatabaseResolver.driver=com.mysql.jdbc.Driver

DatabaseResolver.login=djatokaAdmin

DatabaseResolver.pwd=apiary

DatabaseResolver.maxActive=500

DatabaseResolver.maxIdle=10

DatabaseResolver.query=SELECT identifier, imageFile FROM resources WHERE identifier='\\i';

Add Remote Caching property

DatabaseResolver.maxRemoteCacheSize=150000000

Note: This is in pixels so 150000000 is equivalent to a 15000x10000 pixel image.

Insert Images into Djatoka database

Get Image List

The images' identifiers and file names are originally stored in a text file at

```
$FEDORA_HOME/tomcat/webapps/adore-djatoka/WEB-INF/classes/
imgIndex.txt
```

Insert Images

For each image you wish to server, add a record modelling the following sql command used to serve the image MBB064.jp2

```
INSERT                                                          INTO
djatoka.resources(`identifier`,`imageFile`,`original_file_url`,`jp2_file_url`)
values('apiary:jpeg2000/mbb064',    '/var/www/jpeg2000/MBB064.jp2',
```

```
'http://research.apiaryproject.org/images/original/MBB064.tif',
'http://research.apiaryproject.org/images/jpeg2000/MBB064.jp2)
```

Install Varnish http accelerator

References:

http://fak3r.com/2009/01/27/howto-serve-jpeg2000-images-with-a-scalable-infrastructure/ [References: http://fak3r.com/2009/01/27/howto-serve-jpeg2000-images-with-a-scalable-infrastructure/ http://groups.drupal.org/node/25425/revisions/88248/view http://blog.gootum.com/linux-blog/installing-varnish-reverse-proxy-for-ubuntu http://comments.gmane.org/gmane.comp.web.varnish.misc/1840]

http://groups.drupal.org/node/25425/revisions/88248/view

http://blog.gootum.com/linux-blog/installing-varnish-reverse-proxy-for-ubuntu

http://comments.gmane.org/gmane.comp.web.varnish.misc/1840

Configure Apache

Enable Modules

```
a2enmod proxy

a2enmod proxy_ajp

a2enmod disk_cache

a2enmod file_cache

a2enmod mem_cache

a2enmod deflate
```

Set Apache to listen on port 8019 instead of standard 80

```
vi /etc/apache2/ports.conf
```

Change port 80 to 8019

Set Apache sites to use port 8019 instead of 80

```
vi /etc/apache2/sites-enabled/000-default
```

Change port 80 to 8019

Restart Apache

```
vi /etc/init.d/apache restart
```

Install Varnish

Two methods

Option 1 - via Synaptics Package Manager

Select the following to be installed:

varnish

libvarnish-dev

Option 2 - via command line

```
apt-get install varnish
```

Configure Varnish

Create a pressflow directory

```
mkdir -p /var/lib/varnish/pressflow
```

```
chown varnish.varnish /var/lib/varnish/pressflow
```

Set Varnish to run on port 80

```
vi /etc/default/varnish
```

Change INSTANCE=pressflow

Use Alternative 2

DAEMON_OPTS="-a :80 -T localhost:6082 -f /etc/varnish/default.vcl -s file,/var/lib/varnish/$INSTANCE/varnish_storage.bin,1G"

Redirect Varnish to Apache via default.vcl file

```
vi /etc/varnish/ default.vcl
```

change the backend default values:

backend default {

.host = "hostname";

.port = "8019";

protect against apache rewrite conflicts:

uncomment sub vcl_fetch

add after if statements:

if (obj.http.location ~ ":81"){

set obj.http.location = regsub(obj.http.location,"\:81","");

}

(Re)Starting Varnish

```
/etc/init.d/varnish restart
```

Install HERBIS

Reference Guide: http://projects.brit.org/projects/imls/repository/raw/trunk/documents/development/HERBIS%20Installation.docx

Chapter 7. Apiary Project Code Flow

The Apiary Project uses two powerful approaches to user interfacing and data handling. Both provide the ability to process everything on the server side using the full features of drupal. First, pages can be created within drupal itself. Second, pages external to drupal are supported using ajax and jQuery.

An internal PHP File as a Page

These are the primary type of pages the Apiary Admin pages use. They are called by the automatically created drupal node "apiary".

This file will be loaded from drupal, inside the cti-no-flex-title theme thanks to Themekey.

Example:

http://hostname/drupal/apiary?ref=groundtruth

This page is handled internally by drupal. The apiary page loads the apiary_project/workflow/include/groundtruth.php file and processes everything created therein.

An external PHP File as a Page

Currently index (the actual workspace page), comparer, search and workflow use this approach.

An externally loaded page that is popualted by ajax calls to the drupal site

Example:

http://hostname/drupal/modules/apiary_project/workflow/index.php

This page is completely outside the scope of drupal allowing full control of user interface customization

In our case, we needed to apply jQuery to the full body of the page and this could not be done with drupal adding its own headers, etc first

Ajax calls can still be made to drupal, since the VIEW_APIARY edit was made to the menu.inc file, allowing drupal to do the work and the page to then be populated

The Task Queue is loaded by the response of the ajax call made to http://hostname/drupal/apiary/workflow_ajax/queue_list/1/0/0

What actually takes place when a workflow is started?

Inside the external file, apiary_project/workflow/index.php, are variables that must be previously have been set

$drupal_url = "http://hostname/drupal";

$djatoka_url = "http://hostname:8080/";

These are stored using a SMARTY template

These tell the workflow where to make its ajax calls, mainly using jQuery

There are a number of loaded javascript and css files (apiary_project/workflow/assets/js and apiary_project/workflow/assets/css)

The return responses are then presented to the user with the javascript and css styling already loaded

When a jQuery ajax call is made, drupal checks for a current login and if the user has the VIEW_APIARY permission

Once verified, drupal processes the request on the server side

Often the request requires connection to the Fedora Repository, and a call using Islandora must be made (apiary_project/fedora_commons)

We also write to the Fedora Repository using CURL

Other times drupal executes a Solr query to process the response

All this information is used to return html and JSON data in an expected format

Chapter 8. Apiary Project Metadata Documentation

reference: http://projects.brit.org/projects/imls/repository/show/trunk/Metadata

Chapter 9. Apiary Project Fedora Object Model

reference: http://projects.brit.org/projects/imls/repository/raw/trunk/documents/ObjectModel/ApiaryDigitalObjects.pdf

Chapter 10. Apiary Project Virtual Box

In an attempt to completely avoid the time headache of completing Chapter 2, a Virtual Box instance of the Apiary Project is available for download

Download

Chapter 11. Documentation

Choices and Technical Hurdles

Working with Large Image Files

This was a main focus since the images we would be analyzing were high-resolution, 200+ MB tif files. The Adore-Djatoka Server is designed to handle large images but from a JPEG2000 file format. So the image conversion had to be explored. In the end, we found that a combination of ImageMagick and OpenJPEG would allow us to convert the large tif files into JPEG2000. A significant amount of RAM and or Swap space is definitely needed. System requirements can be viewed here.

Storing Digital Objects

The topic of storing the digital objects was a tough decision as the data we want to keep could be stored in a database, text files or a repository. The ability to track the versioning as the object progressed through the various stages of the workflow was a key necessity. Fedora's inherent use of Darwin Core and its multiple datastream support made it an ideal choice but its overall speed and lack of interfacing was a serious hurdle. Using a database like MySQL would be faster but lacked already developed tools we needed to have on hand to get the rest of the project moving forward.

Performance Issues and Concerns

Speed

The overall objective of the Apiary Project is to reduce the number of man hours it will take to process Specimens from start to finish. Plus, users in the world today want things fast! The faster the UI the better our objective will be met. Djatoka allowed us to handle large images effectively but sometimes had caching issues. Changing the caching properties was a must.

Varnish, an http accelerator, was also installed on the Apiary Project server. This receives http requests before they ever get to Apache2 to handle. So if an image is requested multiple times, the subsequent requests are returned by Varnish instead of being handled with Apache. Needing to see a new page can become an issue if Varnish never lets Apache process it. So the use of No-Cache headers is a good practice for some pages.

The API tools used to request a datastream and its content from Fedora, parse the data and then return results is anything but lightning fast. We chose to implement Apache-Solr indexing which is done when an indexed datastream for an ROI or Specimen Image is saved. For comparison, a request to get the Specimen, Image and ROI object information needed to fill the item browser with 40 Specimens using fedora only loaded in 10 seconds. The same 40 specimen request using Solr to fetch the information loaded in under 1.5 seconds.

Technical Documentation References

OxygenXML [http://www.oxygenxml.com]

I chose to create this documentation using OxygenXML given their relationship with BRIT and its ability to transform the xml into html and pdf formats.

Doxygen [http://www.stack.nl/~dimitri/doxygen/index.html]

I chose to create the code document this using Doxygen with its ability to layout the classes and function in an API-type format