Digitize – Organize – Manipulate

The digitization, preparation and storage of image, sound and movie files for digital archival hosting in general, and for the ODSAS platform in particular

Version 1.2

November 2016

Author:

Laurent Dousset

laurent.dousset@pacific-credo.fr

ODSAS

(Online Digital Sources and Annotation System)

(http://www.odsas.fr)

Table of Contents

1. Before you read this handbook, read this…

Important notice and disclaimer: the instructions, code or software mentioned and listed in this handbook are provided 'as is' and without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of fitness for a purpose, or the warranty of non-infringement. The use of the software or of the present documentation obtained through the ODSAS site or through other means is done at your own discretion and risk and with agreement that you will be solely responsible for any damage to your computer system or loss of data that results from such activities.

We therefore highly recommend you systematically backup your system and files before working with them or applying any of the recommendations produced in this document, and that you systematically test and implement the suggested procedures on copies of your files.

2. General instructions

This chapter summarizes the procedures necessary or useful to prepare and maintain the media files of the ODSAS platform system. While some recommendations and procedures are related to the particular structure of ODSAS, others are useful in and for other archival platforms in the digital humanities.

2.1. Image files

Step	Task	See
Step 1	**Digitize your material**	3.1 - Digitize images, page 7
Step 2	**Rename your files**	4 - Handling filenames and directories for ODSAS, page 12
Step 3	**Place your files in a directory**	4.3 - Organization of files into directories in ODSAS, page 13
Step 4	**Backup your directory with original file format**	3.4 - Backup your originals for long-term storage and preservation, page 9
Step 5	**Duplicate the directory for work and hosting**	3.5 - Create a work directory with your files to be hosted, page 10
Step 6	**Convert your files to JPEG**	5.2 - Convert TIFF (or any other format) images to JPEG images, page 17
Step 7	**Resize your JPEG files**	5.3 - Resize JPEG images, page 17
Step 8	**Proceed with OCR of image files if necessary**	7 - Optical Character Recognition, page 21

2.2. Audio files

Step	Task	See

Step 1	Digitize your material if it is not already in a digital forma	
Step 2	Rename your files	4 - Handling filenames and directories for ODSAS, page 12
Step 3	Place your files in a directory	4.3 - Organization of files into directories in ODSAS, page 13
Step 4	Backup your directory with original file format	3.4 - Backup your originals for long-term storage and preservation, page 9
Step 5	Duplicate the directory for work and hosting	3.5 - Create a work directory with your files to be hosted, page 10
Step 6	Convert your audio files to MP3	6.2 - Convert Audio files to mp3, page 19
Step 7	Split audio files	6.4 - Split large audio files into smaller files, page 20

2.3. Movie files

To come

3. Digitize and backup your files

3.1. Digitize images

Preliminary note: digitizing images is a procedure not only relevant for actual images (pictures, photographs, drawings etc.) but also includes the digitization of printed material or manuscripts. Through the digitization process, text becomes an image. If, additionally to the image itself, you would like to reconvert the latter into actual searchable text, you need to add another step called OCRization (see chapter 7 - Optical Character Recognition on page 21). OCR means Optical Character Recognition and is a procedure that attempts to interpret alphabetical characters in an image.

a) Think before you act. Before you actually start digitizing the material, you need to think how you want to organize your files later on. In the case of a book you want to digitize, the choice is quite straightforward: you will most probably create a set (and thus a directory) in which you will place all the files that result from scanning the book. In other cases, the choice might be more difficult. For example when you digitize photographs taken by a researcher: do you want to organize your material according to the date (year) of the photographs, the geographical location or a particular topic or theme. It is good practice to discuss this issue with the researcher him or herself, if he or she wants to be able to work on the collections later on and since he knows best what the collection is about. In ODSAS, you don't need to systematically choose the same principle of classification or organization for every collection. But a minimum of coherence is helpful later on when the amount of files and directories makes an overview difficult.

b) Organize your material into sets. Once you have decided for a particular collection what kinds of sets you want to create, I suggest you physically organize the original material and the work tasks according to these sets. Also, order the original material within your set in a logical way, either chronological or otherwise. Ordering is important since you will also scan the material in this order and since your files will be stored in this order on the ODSAS platform (note however that the ODSAS platform allows for reordering later on if necessary): the first page scanned will also be the first file displayed in ODSAS.

We recommend creating sets that are neither too small nor too large. It eases navigation, overview and work on the material for the researcher. Of course, you wouldn't cut a book into several sets, but you could cut the two volumes of a

book series into two individual sets. A good number of files for each set is anything between 50 and 800.

c) Scan your material. Scanning itself can either be undertaken by a professional company, or through in-house work. In either case, we recommend the following:

- set your scanner to high resolution:

300dpi is the minimum for an A4 sized page. We recommend you aim at 600dpi and more.

The smaller the original, the higher the resolution needs to be.

Slides or negatives need, since they are small, be digitized at 2400dpi minimal. Check the technical possibility of the scanner. IT may well be that your scanner offers higher than 2400dpi resolution, but that the physical resolution is limited to 2400dpi. In this case, the software will effectuate an interpolation, which is in my opinion a bad choice (since interpolation software is likely to improve over time and since you will be able to apply this procedure later). I recommend in this case you limit resolution to the technical capabilities of your hardware.

If you use digital photography and you wear glasses, do NOT take them off when setting distance and sharpness of the image (an error many beginners make!).

- set your scanner's color depth

Even if the originals are grayscale or black and white, we recommend you scan in color.

Choose 24bit color depth, in millions of colors.

Choose a neutral color, hue, sharpness and contrast setting (make tests). If you scan in a sufficient high resolution, you will be able to change these characteristics at a later stage.

- set your scanner's file format

We recommend scanning-saving your files in uncompressed TIFF

In digital cameras, the format is sometimes limited to JPEG. If this is the case, choose the highest possible resolution available on the camera.

- set your scanner so that it creates sequential file numbering:

In most scanners and digital cameras, files can be numbered sequentially so

that the first image has the name **something_1**, the second **something_2** etc. Make sure this option is activated.

Make sure a number of zeros are added in front of the sequential number. This is necessary for appropriate ordering of your files. If you have the following filenames **1**, **2**, **13** etc., in some systems the ordering will be **1**, **13**, **2** because they are ordered alphabetically. Adding zeroes in front of the sequence helps to avoid this problem. Most scanner software allow for this feature. Digital cameras usually do it automatically. Your filenames should look something like:

something_0001 something_0002 something_0013 etc.

Obviously, if your material has 800 files, you need to have at least 2 zeros in front of the first sequential number. If you have 9000 files, you need 3 zeros in front, etc.

Sequential numbering is crucial to warrant organization and chronology in your set. Other parts of the filename can be changed later on (see 4 - Handling filenames and directories for ODSAS, page 12.

d) Store all the files of one and the same set into one directory before scanning a new set, name the directory using a comprehensive name (see 4.3 - Organization of files into directories in ODSAS page 13) and immediately back up this directory with its content (see 3.4 - Backup your originals for long-term storage and preservation, page 9).

3.2. Digitize sounds

To come

3.3. Digitize movies

To come

3.4. Backup your originals for long-term storage and preservation

Immediate backup of your directories and files after scanning is a compulsory step for at least three good reasons:

(1) you may want to be able to get back to the originals if something bad

happened with the files you are working on or hosting;

(2) you want to be able to get a high resolution version of you file for future paper publication, poster creation etc., and

(3) you need to distinguish what you want to store for preservation as archives, from what you want to store as material on which you actually work.

We recommend you store (backup) your files and directories in at least three places:

(1) One version on your **current hard drive**. This is your **work copy**, the one on which you will at a later stage operate all the modifications needed for online web hosting.

(2) One version of originals on an **external hard drive** which you disconnect while you are working on your work copy mentioned just above. Note that the average life span of hard drives seems to be around 5 years or even less.

(3) One version of originals on **DVD, Blu-Ray** or **tape** that is to be considered as the emergency backup and which you never touch unless there actually is an emergency. Note that the life span of DVD and Blu-Ray discs in not well known.

Additionally, you may want to create a DVD copy of these originals which you could/should hand to the researcher or institution of which you have digitized the material for his or her personal preservation and use.

IMPORTANT NOTES:

- We recommend you do not store your DVD/Blu-Ray copy and your hard drive copy in the same building. Imagine a fire breaks out and you loose both?

- We do not recommend CD-ROMs, and we recommend duplicating DVD, Blu-Ray discs and hard drives every 4 to 5 years, since their life span is limited. Check well that every file on the Blu-Ray disc is actually readable. We have had some surprises with Blu-Ray discs.

- Since your originals are most likely saved in the uncompressed TIFF format, you need to follow the evolution of this and other formats. If the TIFF specifications change or if TIFF even disappears from mainstream, you need to update all your backups to comply with the new specifications or formats.

3.5. Create a work directory with your files to be hosted

Once you have created at least two backup copies, and disconnected your backup hard drive, you may consider the local copy you have on your computer hard drive as your work copy. Make sure you have beforehand named the files and directories

appropriately. See chapter 4 - Handling filenames and directories for ODSAS.

From here on, when this handbook discusses manipulation and modification of image files, we are talking about this work copy and not about the backup copies. Never ever do anything with the backup copies.

We recommend you organize your work copies in the same hierarchical structure as the directories and files will be hosted on the ODSAS platform.

4. Handling filenames and directories for ODSAS

4.1. Conventions for filenames in ODSAS (and beyond)

Names of files (images, sounds, movies etc.) are very important components in the ODSAS platform. The recommendations here need to be applied in a strict manner. Moreover, filenames should contain elementary meta information such as author, year, sequence number. In case of loss of meta information in a database or similar, the filenames should allow to reconstruct at least partially this loss. We suggest you think the issue before naming or renaming files. More on this later. Here a few principles:

1) Never used special characters in filenames. Don't use characters such as the following:

 # ‘ & $ £ “ % = + §

A good rule is to limit the use of non-alphanumerical characters to the underscore: _

2) Don't use language specific alphanumerical characters, such as

é ù à ï ß œ

3) Don't use blank spaces in filenames. If you need to separate entities in a filename, use the underscore

Bad example (don't do this): **my best car.jpg**

Good example (do this instead): **my_best_car.jpg**

4) Make sure the sequence of your files is visible in the filename and precede this sequential number with a few zeroes.

my_best_car_0001.jpg
my_best_car_0002.jpg
my_best_car_0003.jpg
...
my_best_car_0098.jpg
my_best_car_0099.jpg
my_best_car_0101.jpg
...
my_best_car_0999.jpg

etc.

Obviously, 0001 is the first image of a collection (first page or even the cover of a book for example), 0002 is the second page and so on.

Most scanning software and digital cameras provide the possibility to automatically create the sequence of your files.

Note: if you later want to delete one file for one reason or another, you don't need to renumber all other files.

4.2. Include minimal meta information in filenames

It is wise to include minimal information in the filename so to be able to recognize the belonging of the files and to be able to reconstruct in case of meta information loss. You need to think this with respect to your own projects and collections, but here are a few ideas.

I suggest integrating in the filename the author or owner, the geography and/or nature of the original document, the year and, of course, the sequence number.

For example, my (Laurent Dousset's) digital photographs taken in Vanuatu in 2008, I would name:

dousset_dig_vanuatu_2008_0001.jpg
dousset_dig_vanuatu_2008_0002.jpg
etc.

For example, images of the pages of Maurice Godelier's book published in 2002 I would name:

godelier_book_2002_0001.jpg
godelier_book_2002_0002.jpg
etc.

The slides of an unknown author taken Siberia in the 1930s and 1940s I could call:

unknown_slides_siberia_1930_40s_0001.jpg
unknown_slides_siberia_1930_40s_0002.jpg
etc.

4.3. Organization of files into directories in ODSAS

The repository of files needs to organized in directories and subdirectories. There

is in no logical or software problem with having all files of the platform on the same hierarchical level; ODSAS can handle this (with the exception of sound and movie files). But a hierarchical organization has substantial benefits:

1) You know which files belong together (all images of an expedition, all pages of a book etc.).

2) You don't run into problems with finding new and exotic filenames to avoid replacing existing files.

I thus recommend you organize your hierarchy as logically and coherently as possible, creating directories and subdirectories that mean something and in which navigation is eased.

First some general principles:

- all files and directories must be placed inside the folder called **/media**

- inside the **/media** folder, you need to identify, or if necessary create, two additional directories, called **movies** and **sounds**

```
/media/movies/
/media/sounds/
```

Again, strictly speaking it is not necessary, but the software will run better if you do this.

All other files (JPEG mainly) may be organized in directories as you wish within the **/media** directory. For example you may create a directory for books, in which you distinguish authors and years:

```
/media/books/godelier/2002/
/media/books/godelier/2011/
etc.
```

Or you may also and additionally create directories for each researcher:

```
/media/dousset/australia/western_desert_1994/
/media/dousset/australia/western_desert_2001/
/media/dousset/vanuatu/malekula/2008/
etc.
```

Ideally, the filenames within each directory looks similar, for example, the directory

/media/dousset/vanuatu/malekula/2008/

could contain the files names as follows (we don't want to make the filenames too

long, so we use abbreviations):

dousset_dig_van_malek_2008_0001.jpg

dousset_dig_van_malek_2008_0002.jpg

etc.

4.4. Protecting your media directory and subdirectories

The files located in the media directory should not be directly accessible. This is particularly important if you have restricted files or files that should be accessed only by certain users. I therefore suggest you place in each directory an .htaccess and an index.html file which will (or should) prohibit search engines from accessing and indexing the media directories and files.

1) Create a new text file in which you copy-paste the following line. Save the file as **.htaccess** (the word htaccess must be preceded by a dot) and place it in each directory and subdirectory in your /media directory.

```
deny from all
```

2) Create a text file, copy-paste the below and save as **index.html**. Copy this index.html into each directory and subdirectory of your /media directory.

```
<html>
<head>
<title>ODSAS</title>
<META NAME="ROBOTS" CONTENT="NOINDEX, NOFOLLOW">
</head>
<body>
    <h1>Nothing here....Please go back to <a href="/index.php">ODSAS</a></h1>
</body>
</html>
```

Additionally, I suggest you change the permissions of all you media directories and files to 640. You should check this with your system administrator (importantly, your webserver needs to be able to access the files, so www or similar should be in the authorized group).

IMPORTANT EXCEPTIONS:

If you have difficulties accessing sound and movie files from ODSAS, then this may be related to the server's group ownership. A workaround is to do the following:

Don't change the above permissions of the directories called **movies** and **sounds**,

and don’t change the permission of the files that are within these two directories.

Don't copy the above **`.htaccess`** file into the directories called **`movies`** and **`sounds`** and their subdirectories.

But note that this is a temporary solution. You should in all cases check directory authorizations with you system administrator.

5. Preparing Image files for ODSAS webhosting

Currently, only image files of the JPEG format are handled by the ODSAS platform. This may change in the future. Please check back.

The choice of JPEG has been made because of PHP's easy capacity to handle dynamically this format.

5.1. Prerequisites

ImageMagick (download from http://www.imagemagick.org and install)

ImageMagick comes with the commands **`mogrify`** and **`convert`** and can be used in many more contexts and for many more purposes than those explained below, such as sharpening images or working with contrast and color. Check the ImageMagick documentation for more information.

5.2. Convert TIFF (or any other format) images to JPEG images

You need to prepare the image files: convert them into the JPEG format and diminish their size (see below) for appropriate network viewing. We recommend downgrading the files to 1MB each.

Open a terminal and **`cd`** into your image folder, and type

```
mogrify –format jpg *.tif
```

This will convert all TIFF images to JPEG; type something different than **`.tif`** if your images are in another format.

5.3. Resize JPEG images

We recommend downgrading the JPEG files to 1MB each for webhosting on ODSAS. Open a terminal and **`cd`** into your image folder, and type

```
mogrify –define jpeg:extent=1024kb *.jpg
```

Note that with some color images, the rendering is not that good.... You should first test on a sample of copies.

5.4. Quality improvements of JPEG images

You can use mogrify to do many things with the images to improve their quality or visual aspect. Here are those we are using most often:

Rotate all JPEG images of a directory (in the example below, 90 degrees to the left)

```
mogrify –rotate "–90" *.jpg
```

Augment brightness by 50%, diminish saturation by 20%, and leave hue as such:

```
mogrify –modulate 150,80,100 *.jpg
```

Sharpen all images of the directory by a factor of 3:

```
mogrify –sharpen 3 *.jpg
```

Apply twice the “increase contrast” filter to all images of the directory:

```
mogrify –contrast –contrast *.jpg
```

6. Preparing Audio files for ODSAS webhosting

Technically, ODSAS accepts various sound file formats as long as the user has the appropriate plug-in installed in his or her web browser. However, if you wish owners and users to allow for smooth sound time-code related sound annotation, we strongly suggest you convert all sound files to the MP3 format. This conversion can be done with

- Audacity [http://audacity.sourceforge.net/]), and the lame library, is the MUST for audio manipulation.

- You may also install SoundConverter [http://soundconverter.org/]available as an installable package in most Linux systems.

- On MacOSX and Windows you can use Switch [http://www.nch.com.au/switch/index.html] which has a free download for non-commercial purposes.

6.1. Prerequisites

Lame mp3 encoder library (download and install from http://lame1.buanzo.com.ar/)

Sound file splitter mp3splt (download at http://mp3splt.sourceforge.net)

6.2. Convert Audio files to mp3

On Mac OSX you have a pretty good free audio converter preinstalled. It is called *afconverter*. Open a terminal window and type the following to know more about the format of the file(s) you want to convert.

```
afinfo <filename>
```

Type the following to see all the formats into which you want to convert:

```
afconvert –hf
```

Type the following to actually convert the file to mp3

```
afconvert –f 'MPG3' <filename> <newfilename.mp3>
```

There are many free tools available for **Linux**. Most even have GUI (Graphical User Interfaces). Here are a few examples

- SoundConverter http://soundconverter.org/

- audio-convert http://www.archlinux.org/packages/community/any/audio-convert/

6.3. Increase the volume of sound files

In many cases, the sound volume of mp3 files is simply too low. The following procedures will increase the sound volume by 2,5 (see lame documentation). Open a terminal window, **cd** into the directory of your sound file and type

```
lame --scale 2.5 filename.mp3
```

You may also download and install **Audacity** [http://audacity.sourceforge.net/] (and the Audacity mp3 encoding library, which is, no surprise, *lame*), drag-drop or import all files into its window, select all sound tracks, then choose “Amplify” in the Effects menu and let it work. Don't forget to export as mp3 and not just “save” the files.

6.4. Split large audio files into smaller files

Sound files need to be of mp3 format. Moreover, we suggest you cut your files into smaller manageable tracks. More than 10 minutes of sound is often too long for comfortable work. We use the mp3splt tool to split files manually or automatically into smaller files. Mp3splt even has an automatic splitter function that looks for configurable silence to split the sound files.

For example, to find silences and split the file, open a terminal window, **cd** into the directory of your sound file and type the following command

```
mp3splt -s the_filename.mp3
```

Depending on the kind of sound you have in your files, the automatic split may not be always very useful. For example, in interviews there are many silences you want to keep. In this case you better use something like the following

```
mp3splt -a -t 3.0 the_filename.mp3
```

-t = is the time in minutes.seconds of each segment that has to be created

-a = add this option if you want the application to adjust the actual segment size to the next best silence

In the above example, the segments are 3 minutes and 0 seconds long in theory, but adapted to the next best silence.

7. Optical Character Recognition

You may want to ease transcription of image files that contain text. To do this, we suggest you test and use an OCR engine on good quality images that contain typed text. Accuracy of the resulting text heavily depends on the quality of the original print and on the quality of the images. In all cases and before you launch the procedure on hundreds of images, we suggest you test on a few image files first.

7.1. Prerequisites

The tesseract ocr engine available at Google Code (http://code.google.com/p/tesseract-ocr/).

Tesseract language support files.

7.2. OCR one JPEG file

Open a terminal and **cd** into your image folder, and type

```
tesseract -l lge filename1 filename2
```

Replace **lge** with the language code (such as **eng** for English, **fra** for French, etc.)

filename1 is the JPEG input file that needs to be OCRized

filename2 is the name of the text file that will be created with the output.

7.3. OCR several JPEG files at once

To OCRize many files in one step, such as all JPEG files of a directory, you may also create a bash script (and transform it into an executable) that will do the job. Copy paste the below code into a text file, and save as ocrjpg.sh (read the comments in the code to understand what is happening).

```
#!/bin/bash
######################################################################
# Laurent Dousset, March 2012
# This script launches the tesseract OCR engine available at Google
# Code and OCRizes all files of a directory
######################################################################

#STEP 1 #
#Asking user for language
# Language packs need to be installed, see tesseract documentation
echo -n "Language of original JPEG image content (eng, fra ): "
```

```
read –e LANGUAGE

# STEP 2 #
# OCR all jpg files of the current directory
# and write the text into a filename.jpg.txt file
for f in *.jpg;
    do
          tesseract –l $LANGUAGE $f $f;
done

# STEP 2 (optional, erease this section if not desired) #
# Reads all above created .txt files and saves the contents
# into a single text file. Each record is ended by [EOF]

for f in *.txt;
    do
          cat $f | while read line;
          do
                echo "$line" >> summary.txt ;
          done;
          echo "[EOF]" >> summary.txt
done
# END OF SCRIPT
```

To use this script, two solutions.

Solution 1:

1) Copy the ocrjpg.sh into the directory where your images are located

2) Open a terminal and **cd** into this directory

3) Type **sh ocrjpg.sh**

Solution 2:

1) Copy ocrjpg.sh into a binary directory such as /usr/local/bin with

sudo cp ocrjpg.sh /usr/local/bin/

2) Make ocrjpg.sh executable with

cd /usr/local/bin

sudo chmod +x ocrjpg.sh

Your script is now available in all directories without the need to copy ocrjpg.sh into the image directory. Once the above done, **cd** into the image directory and simply type **ocrjpg.sh**.

7.4. OCR JPEG files and create an SQL-formatted string for ODSAS database insertion

Note: this procedure is still exploratory but preliminary tests have given interesting results.

This procedure and script OCRizes the JPEG files given in a directory, it then

creates with the text an SQL query that can be read by MySQL or copy-pasted into PhpMyAdmin for example.

The corresponding ODSAS MySQL table is: **odsas_transcriptions**, of which the structure is as follows (in case the table is not already in your system):

```
CREATE TABLE IF NOT EXISTS `odsas_transcriptions` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `doc_id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `transcription_text` text COLLATE utf8_bin NOT NULL,
  `set_id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `id` (`id`),
  FULLTEXT KEY `transcription_text` (`transcription_text`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```

Procedure:

1) Create an executable script: create an empty text file, copy paste the below and save as **ocrTOsql.sh**. Read the comments in the script since it will tell you what happens and what to do with the script.

```
#!/bin/bash
#######################################################################
# Laurent Dousset, April 2012
# This script launches the tesseract OCR engine available at Google Code
# It loops all JPEG files of a directory and parses it to tesseract,
# then creates an SQL insert string compatible with ODSAS.

# The SQL is an insert into the odsas_transcriptions table
# inputs: language, id of the set, id of the first document from original_files

# The resulting summary.sql file can be directly copied into phpMyAdmin,
# for example, to add the transcriptions to each record

# PREREQUISITES
# - odsas
# - tesseract (+ language support)
# - sed

# TO USE THIS SCRIPT: one of the two following solutions
# 1 - copy file into the image directory and shell "sh ocrTOsql.sh"
# 2 - copy file into a bin directory, (i.e. /usr/local/bin) and make it executable

#######################################################################

#STEP 1 #
#Asking user for language of original image file
echo "Language of original (eng, fra,...): "
read -e LANGUAGE

#Asking user for the set id to which the files belong
echo "SET id: "
read -e SETID

#Asking user for the first id of the first object of the set
# subsequent files will be numbered sequentially
echo "ID of first file: "
read -e ID
#-----------------------------------------------

# STEP 2 #
# OCR all jpg files of the current directory
# and write the text into a filename.jpg.txt file
for f in *.jpg;
```

```
do
          tesseract -l $LANGUAGE $f $f;
done
#--------------------------------------------------

# STEP 3 #
# get rid of any special characters and rubbish in the textfiles
for f2 in *.txt;
do
    sed -i~ 's/[!@#\$%^&*()]//g' $f2;
done
#--------------------------------------------------

# STEP 4 #
# Reads all above created .txt files and saves the contents
# into a single temp file as SQL inserts
for TheFile in *.txt;
do
  echo "INSERT INTO odsas_transcriptions VALUES (\"\",\"$ID\",\"1\",\"" >> summary0.txt;
  echo "[Note: this transcription was produced by an automatic OCR engine]<br>" >>
       summary0.txt;
          cat $TheFile | while read line;
          do
                 echo $line | sed 's/\"/\\\"/g' >> summary0.txt ;
          done;
          echo "\",\"$SETID\");" >> summary0.txt ;
          let ID++
done
#--------------------------------------------------

# STEP 5 #
# escape single quotes and creates the .sql file
sed -e "s/\'/\\\'/g" summary0.txt > summary.sql
#--------------------------------------------------

# CLEAN UP - destroy all intermediary and temporary files
rm *.txt
rm *.txt~
rm summary0.txt

# END OF SCRIPT
```

2) Prepare JPEG files: Create your directory, scan your files, adapt the image settings to ODSAS (see chapter 5 - Preparing Image files for ODSAS webhosting, page 17)

3) Update database: create the set in the ODSAS database and upload the files on the platform (this is not the object of this handbook. It will be discussed when the “ODSAS Admin Handbook” is written, which will be done once an installable ODSAS package is available).

4) Create SQL query: CD into your image directory and launch the ocrTOsql.sh script or binary.

The script will ask for

- language: make sure you have the tesseract language packs installed.

- the Set id: this is the unique identification number given by MySQL once you have created the set (see step 3 above). The information is available in the table called **sets**.

- ID of first file: this is the unique identification number given by MySQL once you have inserted all the files into the database. This information is available in the

table called **original_files.**

5) Read the file created (it is called summary.sql and is placed in the same directory as your image files) into MySQL either by copy-pasting its content into phpMyAdmin or by reading directly into MySQL if you do have access:

```
mysql -u <username> -p
use odsas
source <filepath to summary.sql>
```

Summary.sql has the following structure:

INSERT INTO odsas_transcriptions VALUES ("","id of first object**","1","[Note: this transcription was produced by an automatic OCR engine]** OCR text of first object **","**set id**");**

INSERT INTO odsas_transcriptions VALUES ("","id of second object**","1","[Note: this transcription was produced by an automatic OCR engine]** OCR text of second object **","**set id**");**

etc.

Note: since transcriptions are in ODSAS tied to a particular user, the script will attribute to each transcription the user with id 1 (which is the superuser of ODSAS).

8. Preparing Movie files for ODSAS webhosting

8.1. Movie format and size

ODSAS accepts various movie formats, as long as the user's browser has the appropriate plug-in. However, for smooth time-code-related movie annotations we suggest you save your movies into the .mov or .mp4 format (MPEG4).

The movie size should be 352 x 288 (you can for example chose the H.264 format and set the proper size).

```
Video: MPEG-4 Video 352x288, 25 fps or more
Audio: MPEG-4, Stereo, 44100 Hz
```

8.2. Create thumbnails for movies

If you don't define thumbnails for movies, ODSAS will simply place a generic movie icon to represent them in the website. However, you may add thumbnails if you wish the movies to be represented in a more personalized way. Some principles:

- These thumbnails can be of any size you wish, since PHP scripts will resample them, but we recommend not making them too big.

- There is nothing to do really to make ODSAS take these thumbnails into account, other than simply placing the thumbnail in the same directory as the movie itself. To make sure the right thumbnail goes to the right movie, you need to name the thumbnail with the same basename. Example:

```
my_movie_1_name.mp4
my_movie_1_name.jpg

my_movie_2_name.mp4
my_movie_3_name.jpg
```

As you can see, only the extensions (mov and jpg) are different.

- Currently, your thumbnails need to be in the JPEG format.

To easily create thumbnails of movies, we recommend using ffmpeg [http://ffmpeg.org/] and ffmpegthumbnailer [http://code.google.com/p/ffmpegthumbnailer/]. You may then use the following simple script to create thumbnails with the correct filename of all movies of a directory.

1) Open a terminal and **cd** into the movie directory

2) Copy-paste or type the following into the terminal

```
for f in *.mov;
do
    filename=$(basename $f);
    filename=${filename%.*};
    ffmpegthumbnailer -i $f -t 3 -o $filename.jpg;
done
```

8.3. Convert movies

Movies must be of the mp4 format. While this format is very similar to the Apple .mov format, it does have some difference and the latter is not well read by other browsers than Safari. You therefore need to convert your .mov files to .mp4 files. To do this, as above, write a simple script, copy it into your directory and run it:

```
for f in *.mov;
do
    filename=$(basename $f);
    filename=${filename%.*};
    ffmpeg -i $f -vcodec copy -acodec copy $filename.mp4;
done
```