eeDAP: Evaluation Environment for Digital and Analog Pathology

2/14/14

Contents

0.1 Run eeDAP

How to run a sample eeDAP input file, assuming a successful software set up (Sect. 0.2):

- Click the button “Click to browse for .dapsi input file.”
- Navigate into the folder “sample_inputfiles”
- Select one of the input files: “sample_tissue-1.dapsi”, “sample_tissue-olympus.dapsi”, “ThorLabs.dapsi”, and “ThorLabs-olympus.dapsi”.
- Click “Extract ROIs”.
  - This will create folders in the “sample_inputfiles” directory (“Output_Files”, “Temporary_Registration_Images”, and “Temporary_Task_Images”)
  - This will also create .tif files in the “Temporary_Task_Images” directory. One for each task listed in the .dapsi input file.
- Click “Continue”. This will bring up the data collection GUI.
- Enter your name where you see, “readerID” and click “Next”. You are on your way. The text and labels for the questions are defaults and have no meaning. Hopefully you will figure out what to do. The current tasks are 1) pick a radio button, 2) enter an integer/count, 3) move the slider bar, and 4) click on the image.

0.2 Software set up

0.2.1 Prerequisites

Aperio’s ImageScope

eeDAP also requires the installation of third party software to extract ROIs from WSI images. WSI images are often extremely large (several GB) and are stored as large layered TIFF files embedded in proprietary image formats. eeDAP uses ImageScope, a product of Aperio (Vista, Ca) to read images scanned with WSI scanners from Aperio (.svs) and Hammamatsu (.ndpi). ImageScope contains an Active X controller named TIFFcomp that allows for the extraction and scaling of ROIs.

- TeX

  `www.aperio.com/appcenter`

Matlab compiler runtime libraries

The precompiled stand-alone eeDAP application requires that the Matlab compiler runtime (MCR) library be installed. The MCR library must be the same version as was used to create the stand-alone application. The current version of the MCR library that eeDAP is using is

- R2013a (8.1), Windows 32-bit

The installer can be found at

- **possible local folder**: TEX

  `C:\Program Files (x86)\MATLAB\R2013a\toolbox\compiler\deploy\win32`

- **internet**: TEX

  `http://www.mathworks.com/products/compiler/mcr/`

- **internet**: TEX

  `http://medviso.com/download/mcrinstaller/`

The installer will create a folder such as

- TEX

  `C:\Program Files (x86)\MATLAB\MATLAB Compiler Runtime\v81`

0.2.2 Using the stand-alone application

Unpack the stand-alone application

To unpack the stand-alone application, simply double-click the eeDAP package (eeDAP-1.0_pkg.exe). This will unpack and uncompress the executable (eeDAP.exe) and the required folders (gui_graphics, icc_profiles, sample_inputfiles).

0.3 Hardware set up

0.3.1 Stage

Requirements

eeDAP is written to communicate with a Ludl stage.

Test

When the user selects the mode to be "MicroRT" = Microscope Real Time, the program will force the user to set the communication port for the stage. The stage must be plugged into one of the communication ports, and the correct communications port must be selected.

Fix

0.3.2 Camera Set Up

Requirements:

The code is written to work with a DCAM compatible camera.
eeDAP requires camera images to be RGB24 (8 bit for each channel).
eeDAP requires camera images to have width > 640 and height > 480.
See below for descriptions of cameras at FDA.

Test

When the user selects the mode to be "MicroRT"= Microscope Real Time, there is a button that launches a Matlab utility called "imaqtool". This utility is an interactive GUI to allow you to explore, configure, and acquire data from your installed and supported image acquisition devices. If this doesn't work, Matlab can't find your camera.

Whenever you plug-and-play a camera, Windows enables the default Windows driver, which may actually be the camera manufacturer's driver. If the default Windows driver isn't DCAM compatible, Matlab can't find your camera.

Fix

After plugging in the camera, disable the default Windows driver and update the driver for the attached device to the industry standard DCAM compatible firewire (IEEE 1394 interface) driver. Carnegie Melon (CMU) has a general DCAM driver that may work (1394camera646.exe = version 6.46). It can be downloaded at TEX

`http://www.cs.cmu.edu/~iwan/1394/index.html`

We found out about the CMU driver at TEX

`http://www.mathworks.com/products/imaq/supportedio.html`

- Click on device manufacturer.
- Click on industry standard DCAM IEEE 1394

There is also a demo application at the CMU web site (1394CameraDemo32.exe). It behaves similarly to the Matlab "imaqtool" mentioned above. I think it gets installed when the CMU driver is installed, location TEX

```
C://Program Files (x86)\CMU\1394Camera\bin
-x64\1394CameraDemo32d.exe
```

. It is used to test the camera and different operating modes.

Alignment

Depending on hardware, you may have the ability to completely align the camera with the eyepiece by rotating and shifting the camera. This process is made possible by loosening and tightening one-three screws on the camera mount. This can be done at any time during the registration process. The registration should persist over time and need little adjustments unless the screws are ineffective or the system is bumped. A calibration slide, reticle, and virtual reticle are very useful for this alignment. A good camera mount also has a focus screw that can be adjusted so that the camera and the eyepiece are nearly in focus at the same time.

0.4 FDA hardware specifications

0.4.1 Scanners

The scale factors of the scanner were measured by Neil O Flaherty

The scale factors can also be found embedded in the image files.

Hamamatsu Nanozoomer 2.0HT (at NIH ATC)

- scan_scale at 20x = .4558um/pixel
- scan_scale at 40x = .2279um/pixel

Aperio CS (at NIH ATC and FDA White Oak)

- scan_scale at 20x = 0.50um/pixel
- scan_scale at 40x = 0.25um/pixel

Aperio T2 (at NIH ATC)

We believe the specs are the same as Aperio CS.

Aperio Scanscope XT

We believe the specs are the same as Aperio CS.

0.4.2 Microscopes

The page that opened was Field of View Diameter at microscopyU TEX

`http://www.microscopyu.com/tutorials/java/fielddiameter/index.html`

. The first sentence is, “The diameter of the field in an optical microscope is termed the field number and represents the diameter of the field measured in millimeters at the intermediate image plane.”

Microscopy from the very beginning - Carl Zeiss, Inc. TEX

`http://www.microscopy-news.com/news/carl-zeiss-microscopy-from-the-very-beginning.html`

On page 36 it says, “Eyepieces (or oculars, from the Latin “oculus” = the eye) are the magnifiers with which you view the intermediate image in the microscope, produced by the objective and the tube lens. In the Axiolab microscope, the intermediate image has a useful diameter of 20 mm. Eyepieces are not just simple lenses, but are corrected optical systems consisting of several lenses. It would be a pity if the intermediate image produced with such sophisticated optics were to be impaired just before it reaches the eye.

Normally, the additional magnification provided by the eyepiece is 10x. The intermediate image in this example then has a diameter of 20 cm at a reading distance of 25 cm to the eye. A comparison: this diameter is about as large as the width of this page.”

Axioplan 2 Imaging microscope with an Axiophot 2 head

- Field Number (FN) of the oculars (eyepieces) = 23mm
- Magnification of eyepiece (mag_e) = 10x
- Magnification of objective (mag_o) = 2.5x, 5x, 10x, 20x, 40x
- FOV at Mag_o = FNmm/mag_o
- FOV at 2.5x = 23mm/2.25 = 9.200mm
- FOV at 5x = 23mm/5 = 4.600mm
- FOV at 10x = 23mm/10 = 2.300mm
- FOV at 20x = 23mm/20 = 1.150mm
- FOV at 40x = 23mm/40 = 0.575mm
- Apparent diameter of microscope image "at 25cm" = FN*mag_e = 23cm
- Diameter for ocular micrometer (reticle) = 26mm

Olympus BX43

- Field Number (FN) of the oculars (eyepieces) = 22mm
- Magnification of eyepiece (mag_e) = 10x
- Magnification of objective (mag_o) = 2x, 4x, 10x, 20x, 40x
- FOV at Mag_o = FNmm/mag_o
- FOV at 2x = 22mm/2 = 11.00mm
- FOV at 4x = 22mm/4 = 5.50mm
- FOV at 10x = 22mm/10 = 2.20mm
- FOV at 20x = 22mm/20 = 1.10mm
- FOV at 40x = 22mm/40 = 0.55mm
- Apparent diameter of microscope image "at 25cm" = FN*mag_e = 22cm
- Diameter for ocular micrometer (reticle) = 24mm

0.4.3 Cameras

Point Grey Grasshopper color (GRAS-03K2C-C)

The default Matlab format code is RGB24_640x480, which has an aspect ratio of 1.333. The pixel size and format of the default format equals that of the native sensor specs.

- sensor size = 1/3"
- sensor size = 640x480 pixels (0.3MP)
- pixel size = 7.4um

When attached to the microscope, the **scale factors** are equal to the pixel size divided by the camera adapter magnification (if any) and the objective magnification. When there is no camera magnification:

- 0.185um/pixel at 40x
- 0.370um/pixel at 20x
- 0.740um/pixel at 10x
- 1.480um/pixel at 5x
- 1.850um/pixel at 4x
- 2.960um/pixel at 2.5x

Point Grey Flea2 color (FL2G-50S5C-C)

Native pixel format The Matlab format code that uses all of the native pixels is F7_RGB24_2448x2048. It is a non-standard format (Format_7) with aspect ratio 1.953 and the following specs:

- sensor size = 2/3"
- sensor size = 2448x2048 pixels (5.0MP)
- pixel size = 3.45um

Default format The **default** Matlab format code is RGB24_1024x768, which has a standard aspect ratio of 1.333. The default pixels are 2x2 bins of the native pixels. Consequently, the pixel size is 6.9um.

When attached to the microscope, the **scale factors** are equal to the pixel size divided by the camera adapter magnification (if any) and the objective magnification.

When there is no camera magnification, the pixel size of the default Matlab format (6.9um) is divided by the objective magnification.

- 0.1725um/pixel at 40x
- 0.3450um/pixel at 20x
- 0.6900um/pixel at 10x
- 1.3800um/pixel at 5x
- 1.7250um/pixel at 5x
- 2.7600um/pixel at 2.5x
- 3.4500um/pizel at 2x

Alternative format The Matlab format code F7_RGB24_1600x1200 uses the native pixels

but a smaller standard format.

0.4.4 Reticles

Placed in eyepiece at Intermediate Ocular Plane (IOP).

Diameter for Zeiss microscopes = 26mm

Diameter for Olympus microscopes = 24mm

Klarmann Rulings KR-429

width and length of 10x10 grid at IOP = 12.5mm
apparent width and length of 10x10 grid "at 25cm" = 12.5cm
grid spacing at IOP = 1.25mm
grid spacing at stage at mag_o
5x = 1.25mm/5 = 0.25000mm
10x = 1.25mm/10 = 0.12500mm
20x = 1.25mm/20 = 0.06250mm
40x = 1.25mm/40 = 0.03125mm
apparent grid spacing "at 25cm" = 12.5mm

Klarmann Rulings KR-32536

This is a custom reticle with cross-hair-like fiducials pointing to gaps: 2 are 1mm x 1mm and 3 are 0.5mm x 0.5mm.

0.4.5 Displays

HP L2335
Size: 49.6cm x 31.1cm
Size (pixels): 1920 x 1200
Pixel Pitch = 258um
Contrast: 500:1
Brightness: 250 cd/m^2

Dell 1908 FPt
Size: 37.6cm x 30.1cm
Size (pixels): 1280 x 1024
Pixel Pitch = 294 um
Contrast = 800:1
Brightness = 300 cd/m2

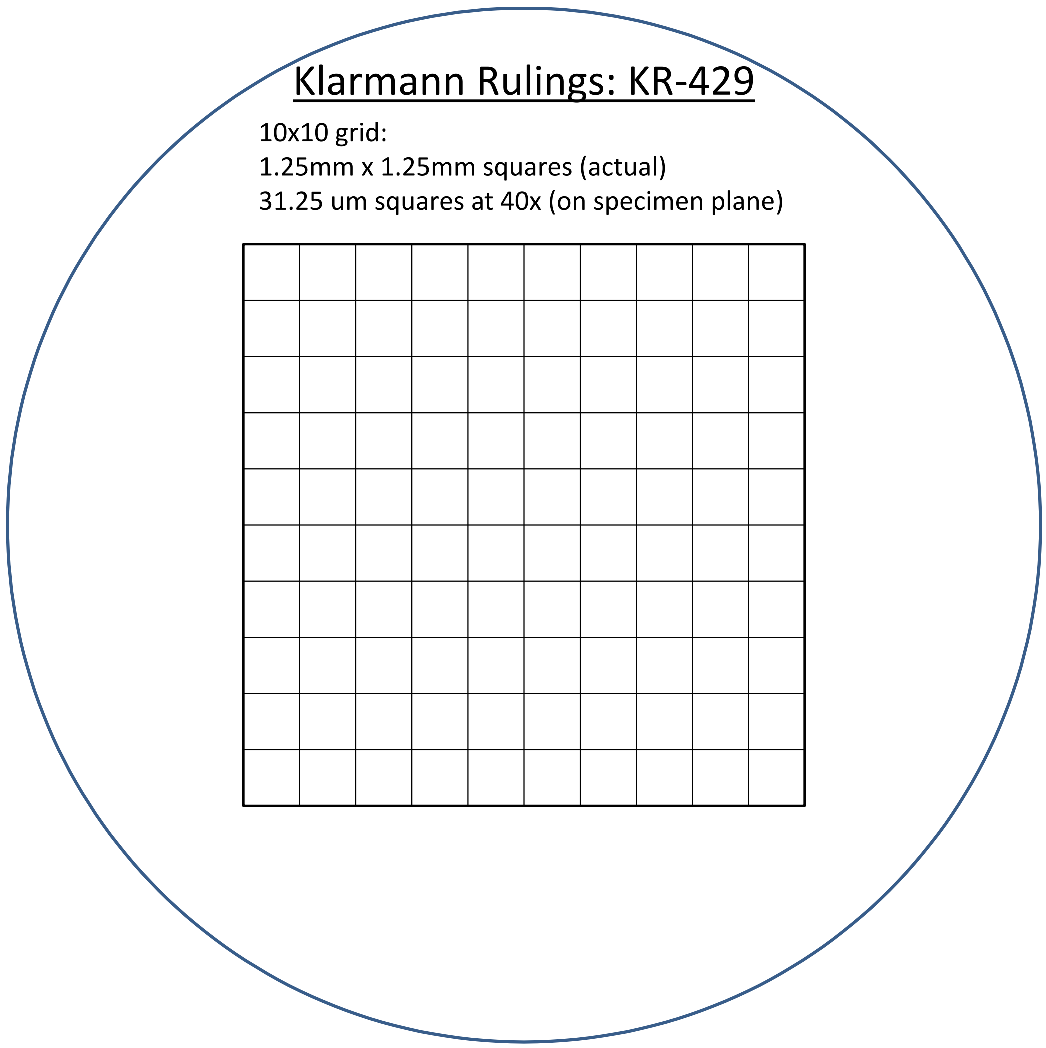
Lenovo Thinkvision L220XWC
Size: 47.4cm x 29.6cm
Size (pixels): 1920x1200
Pixel Pitch: 247 um
Contrast = 1200:1
Brightness = 325 cd/m2

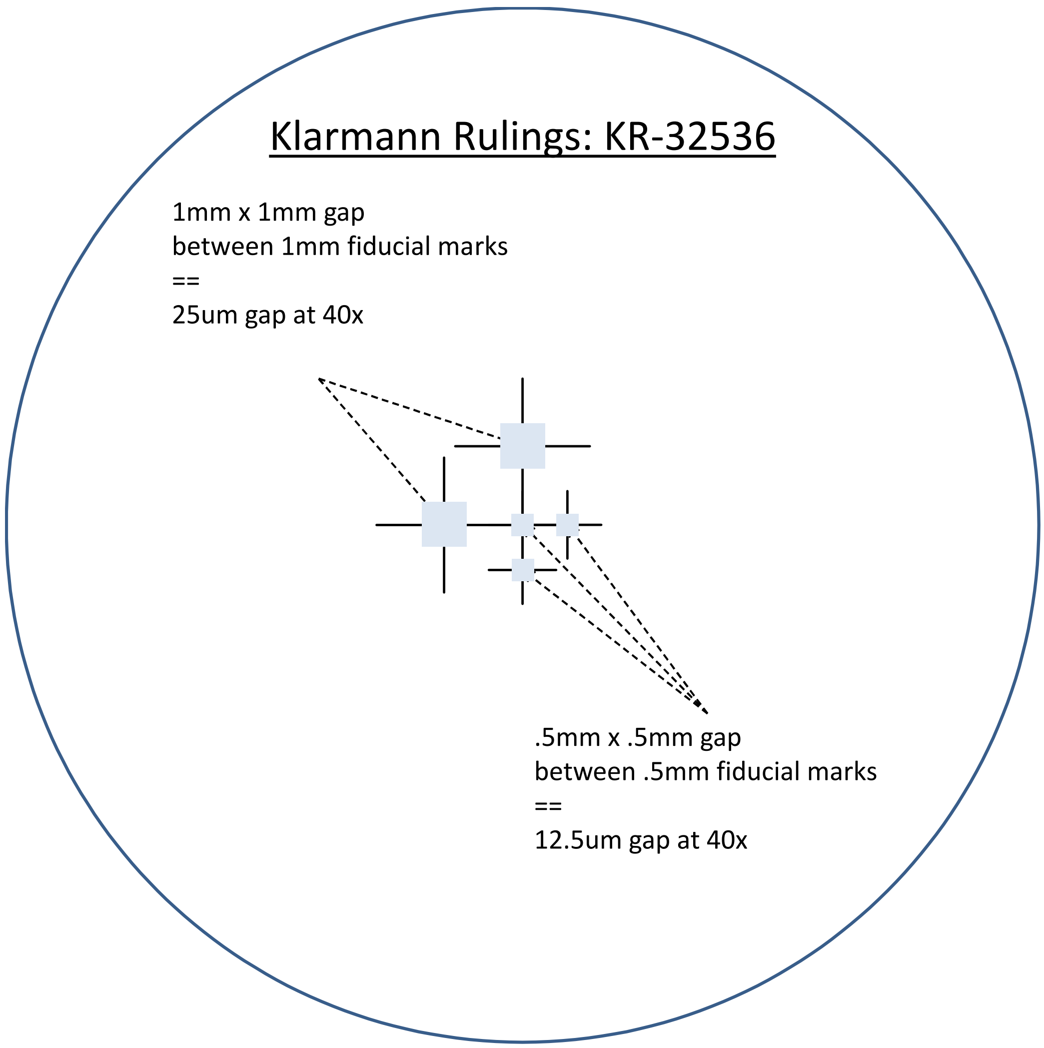
0.5 Program Variables

0.5.1 Assigned in the Input File

n_wsi (int)

The number of wsi used in the study.


Klarmann Rulings: KR-429
10x10 grid:
1.25mm x 1.25mm squares (actual)
31.25 um squares at 40x (on specimen plane)


Klarmann Rulings: KR-32536
1mm x 1mm gap
between 1mm fiducial marks
==
25um gap at 40x
.5mm x .5mm gap
between .5mm fiducial marks
==
12.5um gap at 40x

mag_obj (float)

Magnification of the objective to be used in the study.

reticle (string)

The model number of the reticle.

cam_format (string)

The Matlab format code for the camera, e.g. "RGB24_1024x768"

cam_pixel_size* (float)

The size of the camera pixels [um]. This depends on possible (de)magnification of the mounting adapter. The camera pixel size may not be equal to that given in the camera specifications. The camera format may lead to binning which changes the camera pixel size.

cam_lres_mag (float)

The low resolution magnification applided to the camera to be used for registration.

cam_hres_mag (float)

The high resolution magnification applied to the camrea to be used for registration and study.

scan_scale (float)

The width in specimen units that a scanner pixel represents [um/scan_pixel].

BackgroundColor_R (float)

The red value of the background color given as a fraction between zero and one.

BackgroundColor_G (float)

The green value of the background color given as a fraction between zero and one.

BackgroundColor_B (float)

The blue value of the background color given as a fraction between zero and one.

AxesBackgroundColor_* (RGB = float, float, float)

The red value of the background color of the axes area given as a fraction between zero and one.

ForegroundColor_* (RGB = float, float, float)

The red value of the foreground color given as a fraction between zero and one.

FontSize (int)

The font size.

taskorder (int)

The place in the order that each task will be executed. (See below).

0.5.2 myData.settings

List of Variables Assigned in the Input File

cam_w

int [determined from camera]
Width [pixels] of the camera image.

cam_h

int [determined from camera]
Height [pixels] of the camera image.

cam_roi_w

int [hardcoded]
used to determine width of camera patch for registration

cam_roi_h

int [hardcoded]
used to determine height of camera patch for registration

cam_scale_lres

float [derived]
The width in specimen units that a camera pixel represents at low magnification [um/cam_pixel].

cam_scale_hres

float [derived]
The width in specimen units that a camera pixel represents at high magnification [um/cam_pixel].

cam2scan_lres

float [derived]
The conversion factor to convert the width of a camera pixel at low mag to the width of a scanner pixel [scan_pixel/cam_pixel].

cam2scan_hres

float [derived]
The conversion factor to convert the width of a camera pixel at high mag to the width of a scanner pixel [scan_pixel/cam_pixel].

eye_cam_offset

(int, int) [determined]
This variable determines the offset caused by misalignment between the camera and the eyepiece. It is set in Administrator_Input_Function -> align_eye_cam.

scan2cam_lres

float [derived]
1.0/cam2scan_lres [cam_pixel/scan_pixel]

scan2cam_hres

float [derived]
1.0/cam2scan_hres [cam_pixel/scan_pixel]

Taskorder defines the place in the order that each task will be executed. After a study is executed, the place in the order that each task is executed will be saved in the field myData.tasks.order.

- 0: The order of tasks is randomized. The field myData.tasks.order is ignored on input.
- 1: The order of tasks follows the listed order. The field myData.tasks.order is ignored on input.
- 2: The order of tasks follows the order given by the field myData.tasks.order.

The field of view of the microscope (diameter of the field at the specimen) equals the field number divided by the magnification of the objective.

The apparent field of view of the microscope (diameter of the intermediate image assumed to be viewed at 25cm) equals the field number times the magnification of the eyepiece.

0.5.3 myData.tasks_in, myData.tasks_out

myData.tasks_in holds the tasks in the same order as they appear in the inputfile.

myData.tasks_out holds the tasks in the order determined by myData.settings.taskorder

0.5.4 myData.wsi_files

0.5.5 Qtype: Question types

0.6 eeDAP Developers

0.6.1 Creating the stand-alone application

Deployment project

Start a new "Deployment Project" named eeDAP. Matlab will create a project file and folder. The location for this project is not important. This content is not archived because it duplicates content already in the archive.

- 2013: Go to the "APPS" tab on the Matlab GUI.
- -OR-
- 2012: Go to File->New -> Deployment Project.

Build the stand-alone application

Move the main file into the build (Administrator_Input_Screen.m). Also add the "tasks" folder to the build. Click on the "Build" button. The build takes as much as 5 minutes! The build creates the "src" folder underneath the project folder.

Presumably, the build doesn't recognize the task functions as called from the main program. This might be because they are called by a function handle and not by name.

Package the stand-alone application

Add related files and folders to the package and click on the "Package" button. This creates the "distrib" folder underneath the project folder. Packaging also creates the package to be unwrapped by the client (eeDAP-1.0_pkg.exe). This package is what you share with the client. The related folders (and their contents) we are packing in the stand-alone package are:

myData.tasks_in, myData.tasks_out		
Field	**Type**	**Desc**
ID	string	ID given to the task.
order	int	If taskorder==2, then this field specifies the place in the order this task will be executed. Otherwise this field is ignored.
slot	int	The slot number identifies the wsi file and image for the task.
roi_x	int	The horizontal center of the ROI to be extracted from the wsi file.
roi_y	int	The vertical center of the ROI to be extracted from the wsi file.
roi_w	int	The width of the ROI to be extracted from the wsi file.
roi_h	int	The height of the ROI to be extracted from the wsi file.
img_w	int	The initial width of the displayed ROI. (Rotated with respect to wsi file s.t. it corresponds to the extracted height).
img_h	int	The initial height of the displayed ROI. (Rotated with respect to the wsi file s.t. it corresponds to the extracted width).
Qtype	string	A label identifying the task (See **??**).
Qtext	string	The text displayed to the user providing instructions to complete the task.
MoveFlag	int	Flag indicating whether or not moving is allowed (MoveFlag=1) or not (MoveFlag=0).
ZoomFlag	int	Flag indicating whether or not zooming is allowed (ZoomFlag=1) or not (ZoomFlag=0).
Op1	string	Label for task option #1 (See **??**).
Op2	string	Label for task option #2 (See **??**).
Op3	string	Label for task option #3 (See **??**).
Op4	string	Label for task option #4 (See **??**).

Table 1: Table Caption

myData.wsi_files		
Field	**Type**	**Desc**
fullname	string	name of wsi file including full path
wsi_w	int	width in pixels of wsi file
wsi_h	int	height in pixels of wsi file

Table 2: Table Caption

- gui_graphics

- icc_profiles

- sample_inputfiles (first deleting backup files and temporary files and temporary folders)

- Maybe docs some day and this user manual.

0.7 Modifying eeDAP code

0.7.1 Tasks

To be written.

0.7.2 Virtual Reticles

To be written.