Open Application Development (OAD)

Towards OAD of microscopy & image analysis with ZEN (Software) and .czi (File Format)

Dr. Markus Neumann
Software and Camera Requirements

Dr. Olaf Selchow
Light Sheet Fluorescence Microscopy

Paris, 24.06.2013

Carl Zeiss Microscopy GmbH

Open Application Support
→ multifaceted concept

1.) Documented file format
- → CZI
  - → Bioformats reader
- → OME-TIFF

Olaf

2.) Customization of ZEN through Macros
- → Python macro language

Markus

3.) Connecting ZEN with other software solutions
- → COM / TCP-IP

4.) Image J plugin
- → interaction with Image J / Fiji directly within ZEN

5.) Experiment Feedback
- → modify experimental protocol through online image analysis

Markus

.czi File Format

1. Unified File Format for all Carl Zeiss Light Microscopy Systems
   - Consistently used by Carl Zeiss Microscope Systems (ZEN software) since 2010/2011
   - Replaces .lsm and .zvi
   - Optimized for acquisition performance and readability / transparency
   - .xml structure of Meta data developed with an eye on OME Meta data scheme (to facilitate compatibility)
   - Binary data and machine parameter optimized for image acquisition

2. Open up access to microscopy data acquired with Carl Zeiss Microscopes
   - Make proprietary .czi file format readable to everybody who needs it
   - Support implementation (e.g., Bioformats, but also 3rd party commercial)
   - Exporting to open file formats (OME-tiff)

Carl Zeiss Microscopy Software

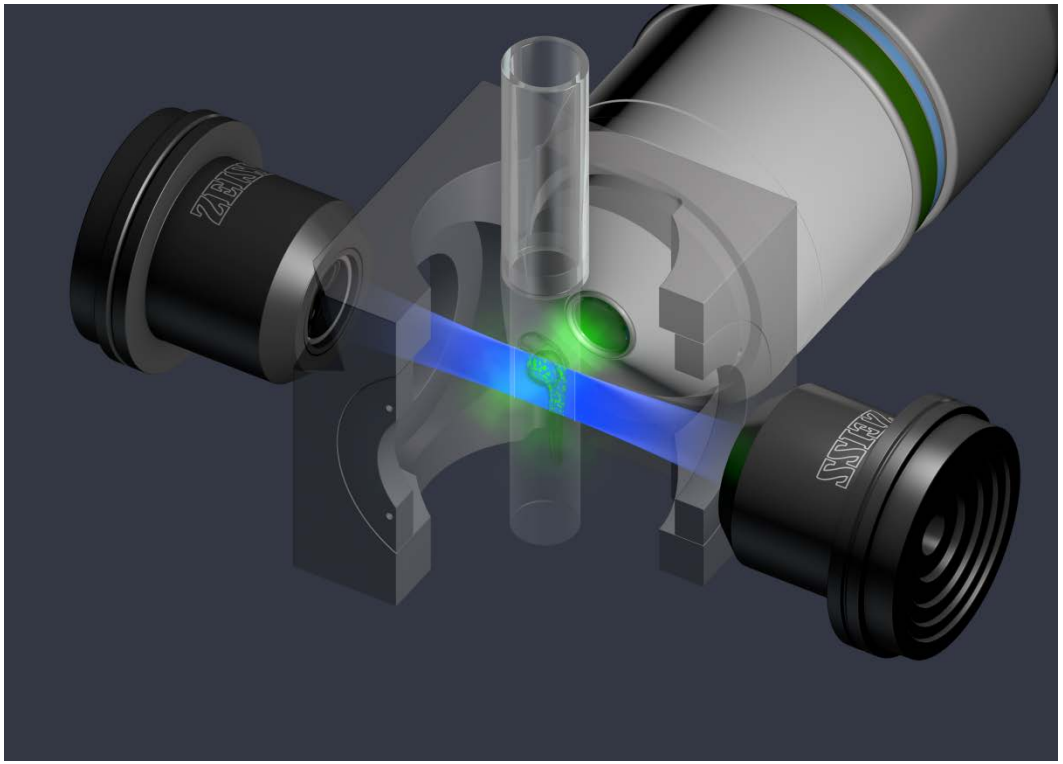
< 2007	2007 - 2011	2011	2012	> 2012	
Axiovision	Axio-Vision				Cellobserver (wf, SD) TIRF Vivatome Apotome Stereo & Macroscopes
		ZEN (blue edition)	New OAD (Python, etc) ZEN 2012 (blue)		
common structures		.czi GUI elements	.czi Processing libs More GUI	?	
		ZEN (black edition)	ZEN 2012 (black) VBA Support will stop		LSM 710 / 780 LSM 710/780 NLO Elyra Lightsheet Z.1
AIM	ZEN				

Lightsheet Z.1

Light Sheet Fluorescence Microscopy by Carl Zeiss

Light Sheet Fluorescence Microscopy (LSFM)

Principle

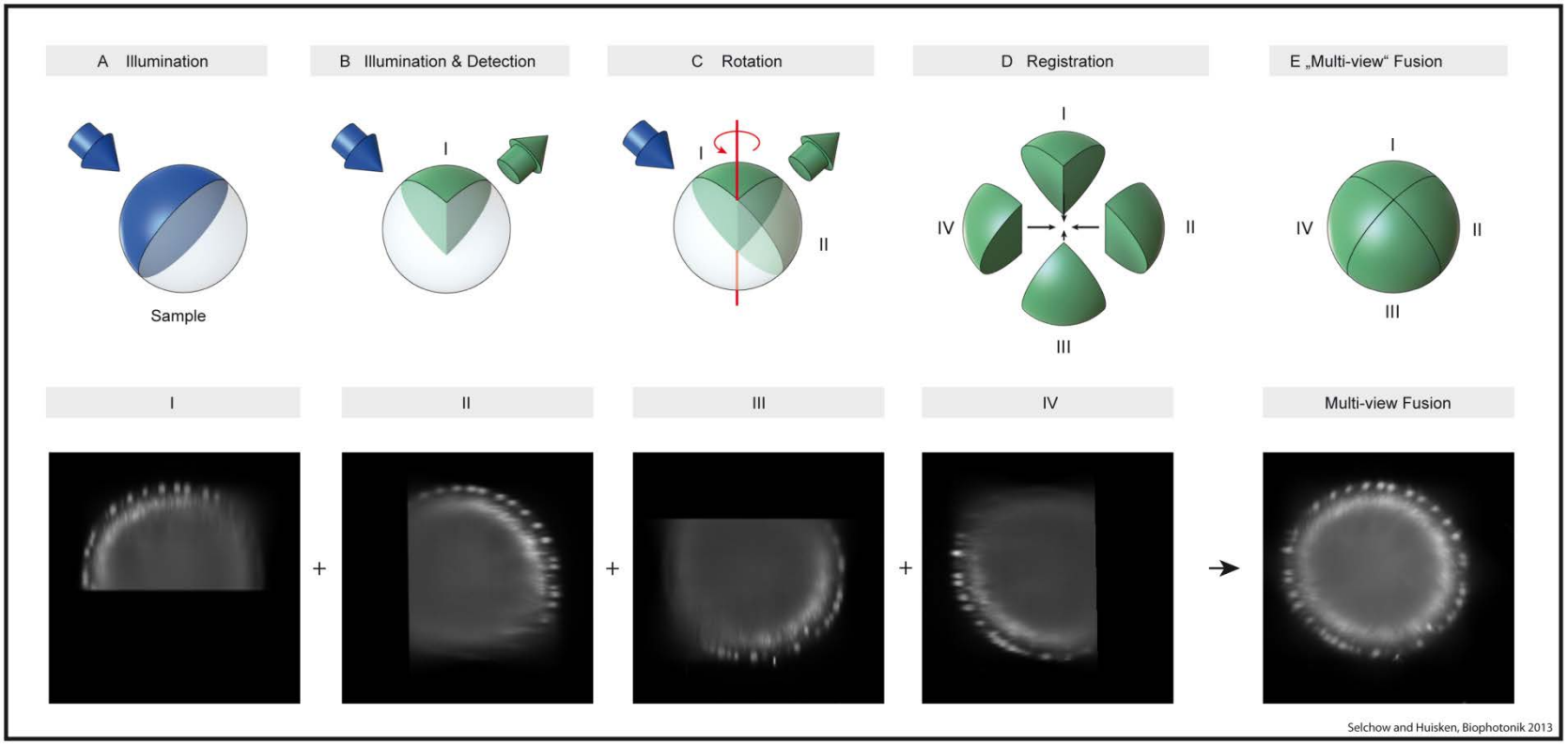


Sample mounted vertically in hydrogel

- Translation & rotation: easy positioning, z-stacks & Multiview
- Suspended in medium / buffer: ideal for live imaging

Multiview Imaging

Multiview Imaging: Sequential acquisition of multiple stacks of optical sections from different directions. In LSFM they are usually taken from different rotation angles.

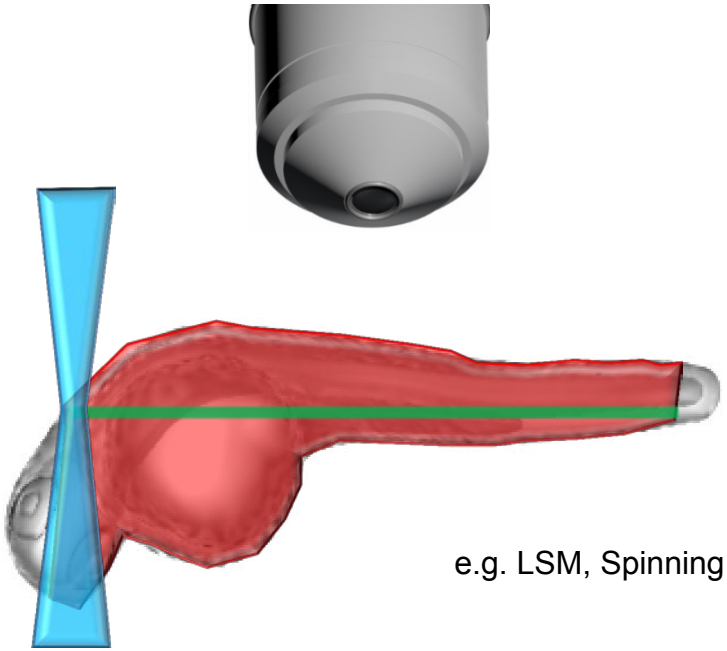


Benefit:

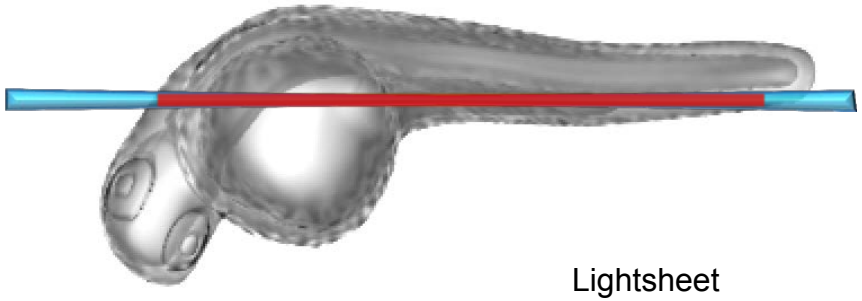
- Complementary information in different views (more info)
- Potentially improved resolution (depends on specimen)

The Principle of Light Sheet Fluorescence Microscopy

Eliminate photo-damage – longer and faster imaging



e.g. LSM, Spinning Disc



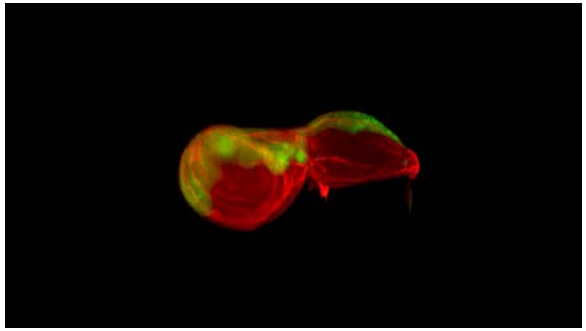
Lightsheet

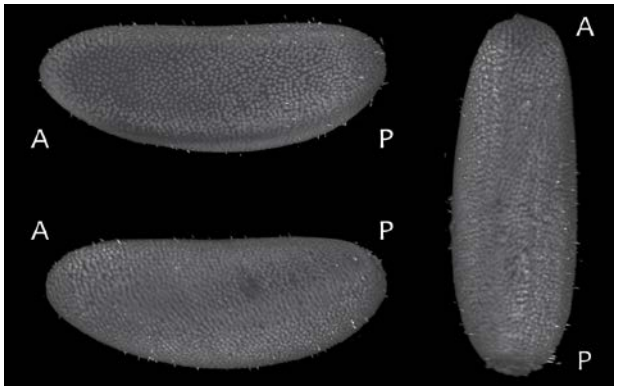
Up to 1000x less photo-damage

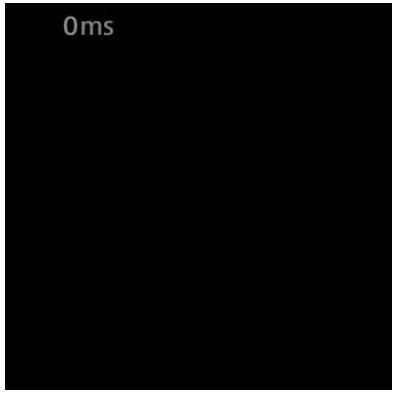
- Long periods of observation
- Fast imaging
- Multiview imaging

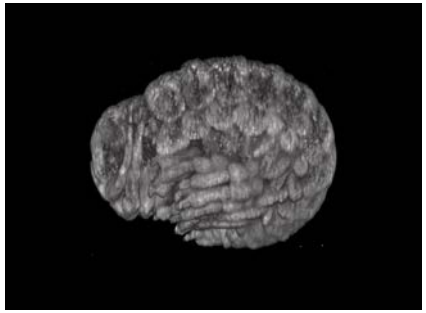
The benefit: Taking Live Imaging into New Dimensions

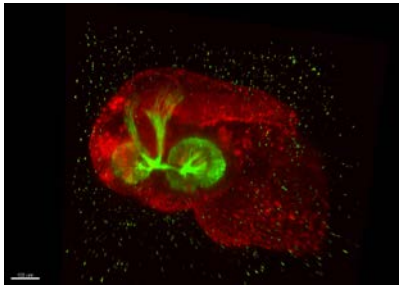
With Light Sheet Fluorescence Microscopy by Carl Zeiss: Lightsheet Z.1

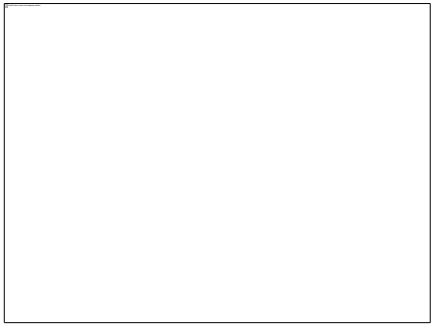


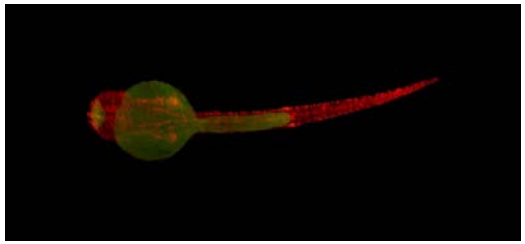








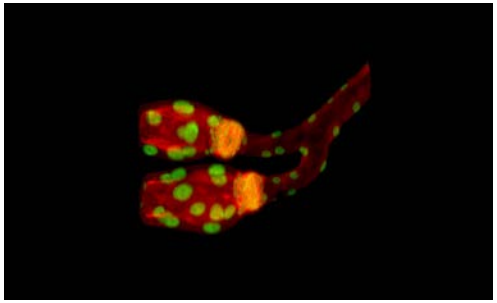




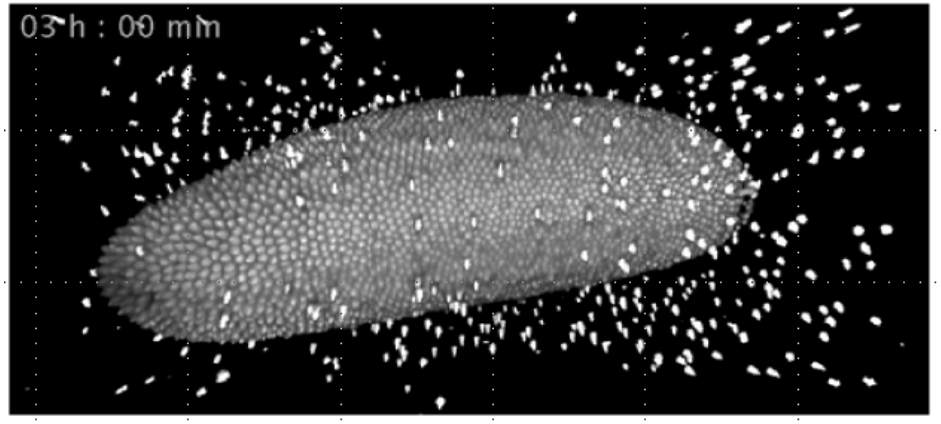
The whole volume

Every few minutes

For hours and days!

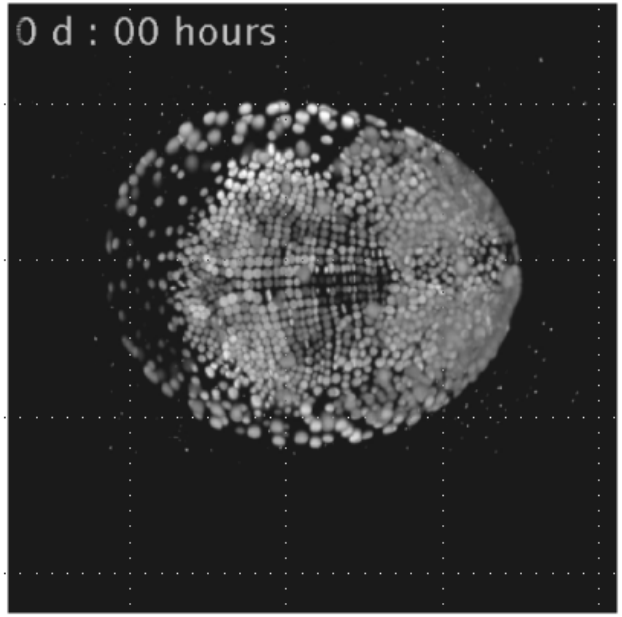


Opening up – the example of large data processing of Light Sheet Fluorescence Microscopy Data



Data by P.Tomancak and A. Pavlopoulos
MPI Dresden
Acquired on a Lightsheet Z.1
Processed on MPI storage & computing cluster

Parhyale hawaiensis
4.5 days, at 7.5 min intervals
HIS-YFP? / 3 views, 2 ill sides / view
Processed in Fiji, rendered using ImgLib2

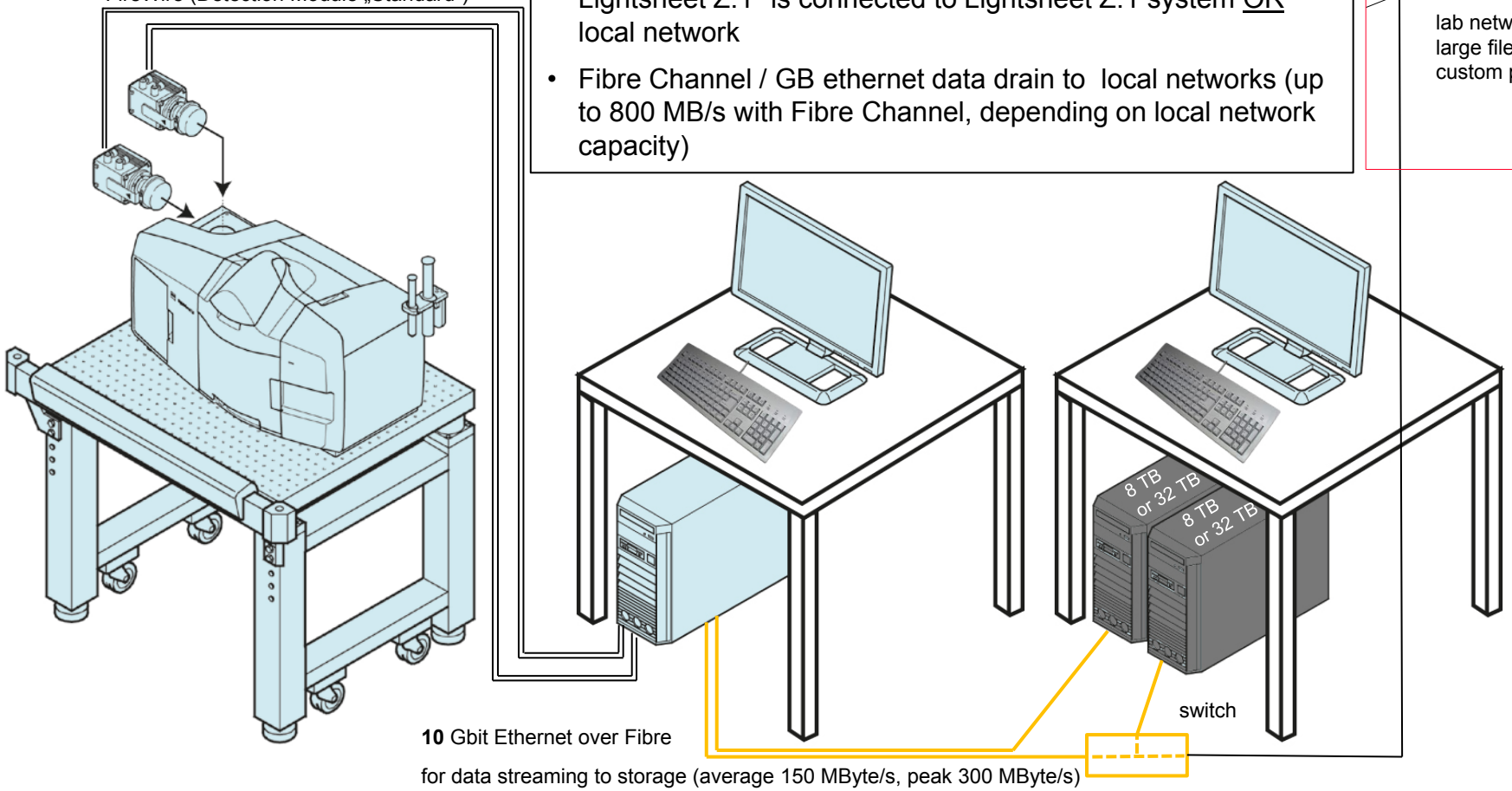


Drosophila Melanogaster
Ca 20 hr timelapse
HIS-Ruby
6 views, 2 ill sides / view
715 timepoints (90 sec intervals)
Processed in Fiji, rendered using ImgLib2

- **Volume data reconstruction required**
- **Deconvolution often desired / recommended**
- **Terabytes of data / day**
- **Experiment-specific p&a requirements**

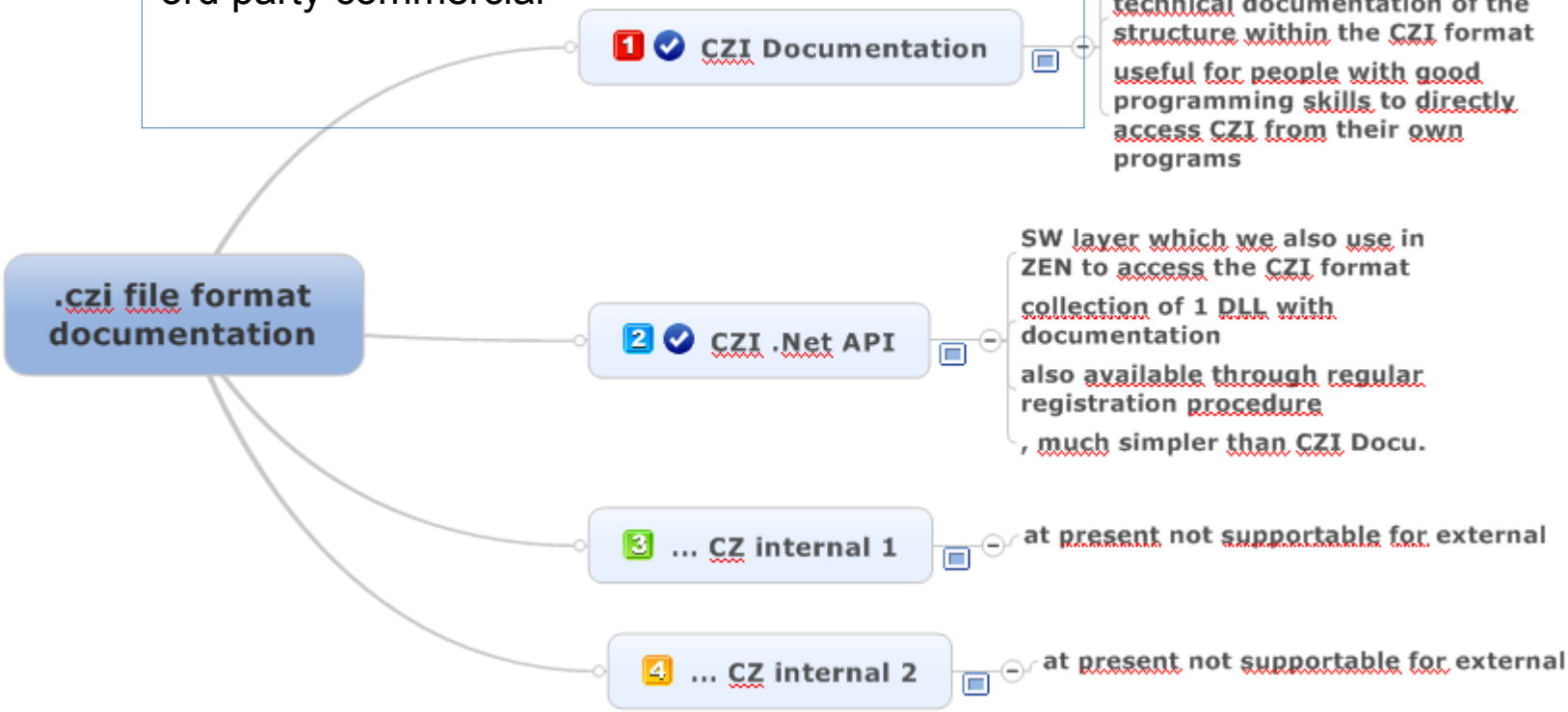
It‘s about interfaces

- Including 2 „Storage and Data Analysis PC for Lightsheet Z.1“
- Storage Volume 2x 32 TB
- Switch ensures that „Storage and Data Analysis PC for Lightsheet Z.1“ is connected to Lightsheet Z.1 system OR local network
- Fibre Channel / GB ethernet data drain to local networks (up to 800 MB/s with Fibre Channel, depending on local network capacity)



.czi documentation

Bioformats 5.0.0-beta1
Some customers
3rd party commercial



ZEN blue Open Application Development
→ Key Messages

Microscope and the image acquisition are only part of a higher-ranking workflow

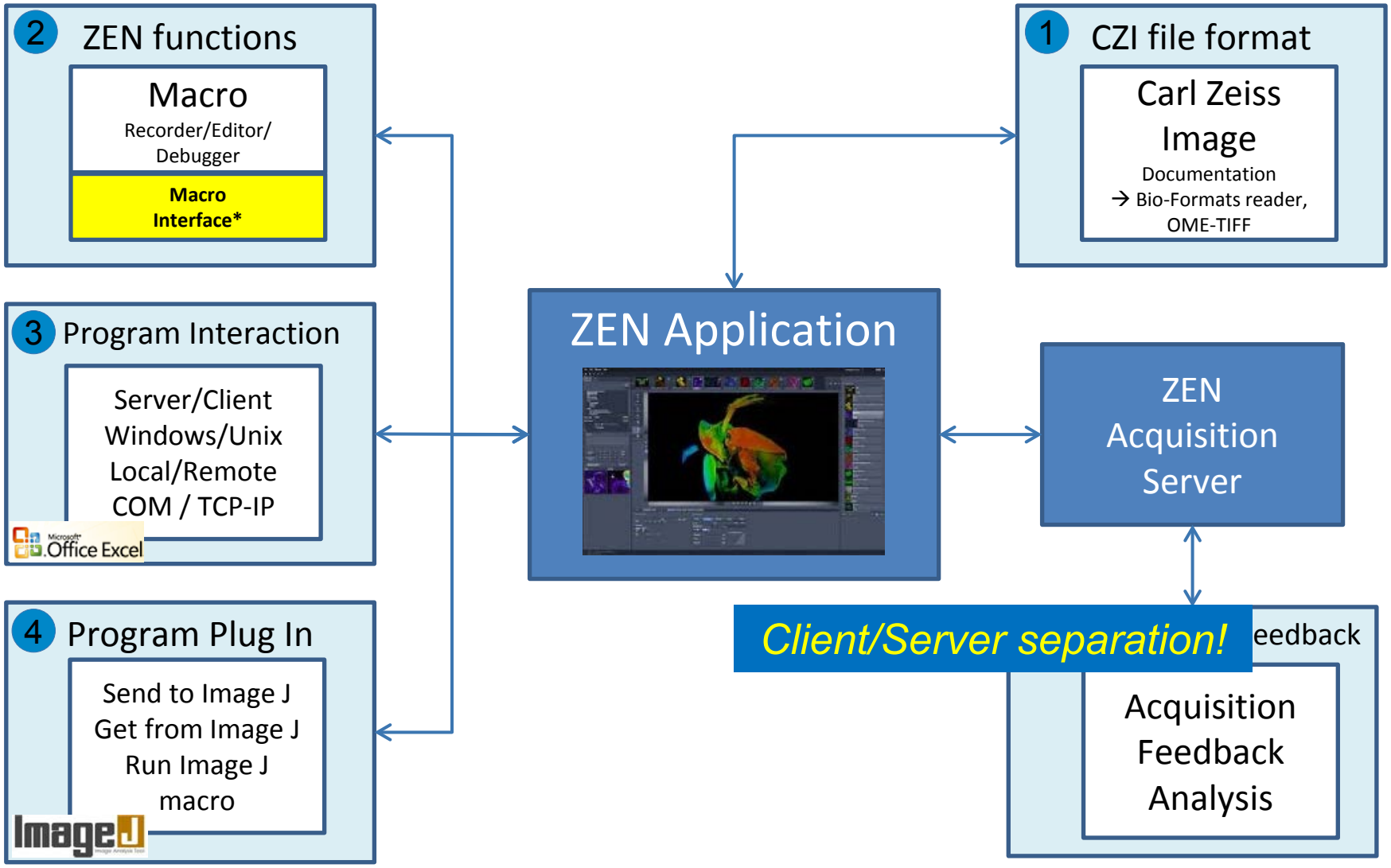
ZEN now supports **Open Application Development** (OAD) with **Macros** and open software interfaces

OAD provides generic **building blocks** allowing users to create own solutions for a **broad range of applications**.

Users are **not locked** into a proprietary ZEISS system but can **integrate** ZEN in their IT workflows

→ connect the imaging system and ZEN with the “outside world”

OAD-Concept (simplified)



Macro Interface
→ Reduced Class Library of ZEN functionality

- About **100** classes
- reduced nr. of available functions and attributes
- Stable! → only incremental changes

```
15 ## Import system library
16 import System.Byte as byte
17 ##
18 ## Get active image
19 image = Zen.Application.Documents.
20 ## Create RGB extractions
21 outputImages = Zen.Processing.Util
22 splitR = outputImages[0]
23 splitG = outputImages[1]
24 splitB = outputImages[2]
25 ## Create 3 channel image
26 imageAdd = Zen.Processing.Utilitie
27 imageAdd = Zen.Processing.Utilitie
28 ## Define RGB colours for channels
29 r = byte.Parse('255')
30 imageAdd.SetChannelColor(0,r,0,0)
31 g = byte.Parse('255')
```

```
[]
ActiveDocument
Add
Contains
Count
GetByFileName
GetByName
GetEnumerator
Remove
RemoveAll
SaveAll
```

Property read-write ZenDocument ActiveDocument

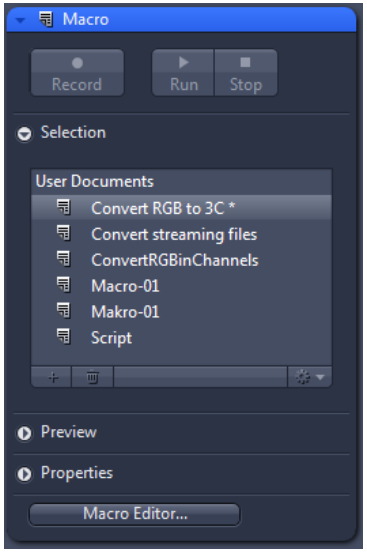
Watch | Message

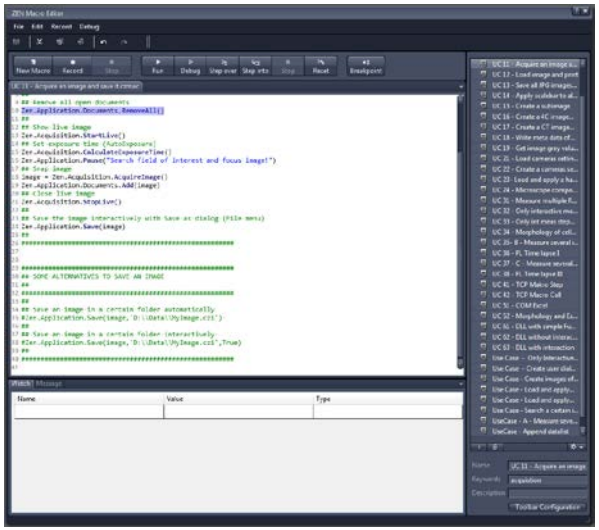
Macro IDE
(Integrated Development Environment)

Right Tool Area
- Run-Version (available for every ZEN blue version > ZEN Lite)

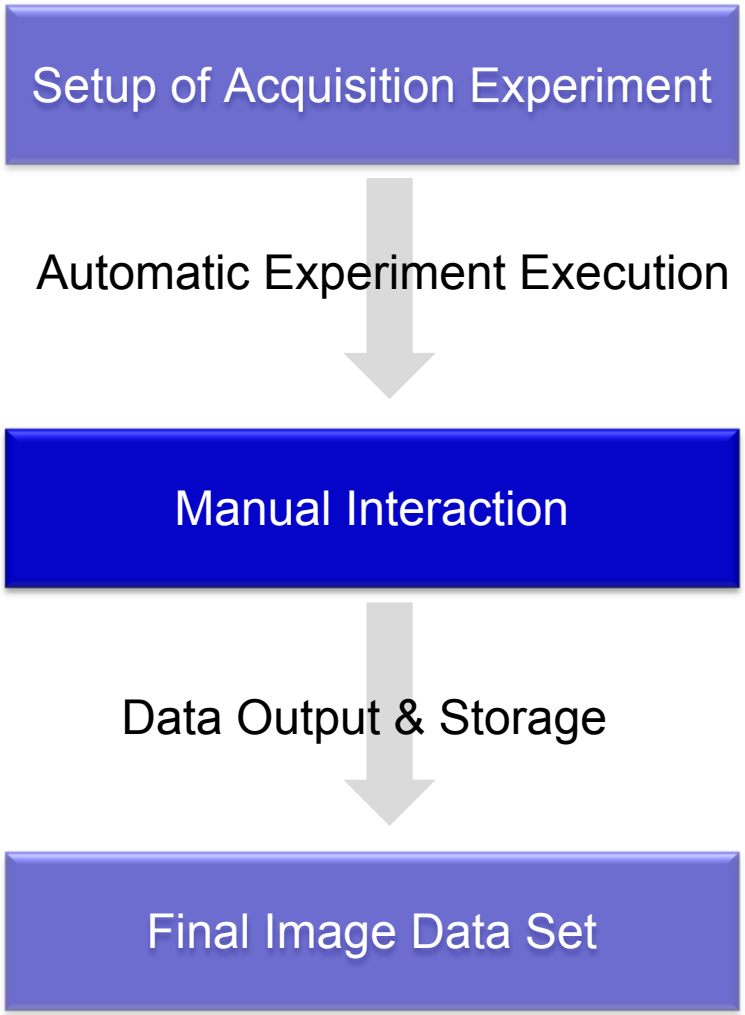
IDE
- Editor
- Debugger
- Makro Recorder (still limited)

Python macro language
- easy to use
- popular within the open source development community
- widespread within the academic life science community
- lots of external resources (WEB, books, libraries, tools, sample code, etc.)





Experiment Feedback → Imaging in the past



Steps of a Standard Acquisition Experiment:

- Configuration of an Imaging Experiment
- Start and automatic Execution until the Experiment is finished.
- Optionally, the user is able to ***manually*** interact with the system to influence the acquisition.

Experiment Feedback → Imaging today

ZEISS

Setup of Acquisition Experiment (including Feedback-Actions)

Automatic (but adaptable) Experiment Execution

Adaptive Acquisition Engine

Data Output & Storage

Final Image Data Set

Steps of an experiment with Experiment-Feedback:

- configuration of an imaging experiment, including definition of feedback actions.
- start and automatic execution
- acquisition automatically adapts to altered conditions
  - hardware (e.g. temperature)
  - analysis (e.g. nr. of rare events)

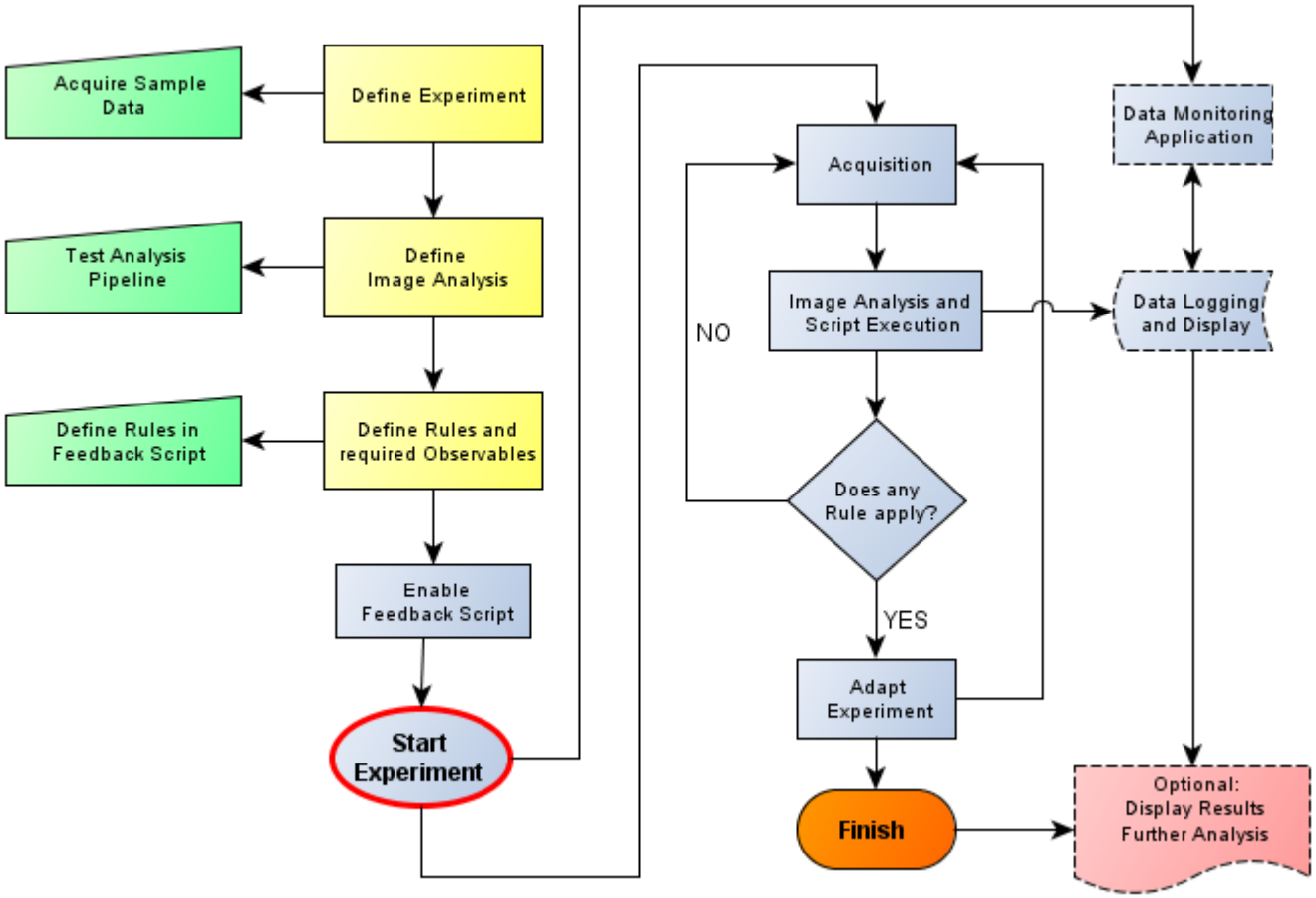
Experiment Feedback
→ Key Features

- Adaptive Acquisition Engine → experiments can modify itself according to the script or rules defined by the user

- Modification of a running experiment depending on the current system status and / or the nature of the acquired data (online analysis).

- Data logging or starting an external application (Python, Fiji, MATLAB,), directly wihin the imaging experiment.

Experiment Feedback → General Workflow



5 Experiment Feedback

Acquisition
Feedback
Analysis

Example: Acquire until a certain number of objects is detected

ZEISS

Script Editor for Experiment Feedback

Single Execution on Experiment Start

```
lastindex = 0
pos = 0 # just a counter
cells_total = 0 # total number of cells
imagename = ZenService.Experiment.ImageFileName
script = "display_results_tiles.py"
```

Repetitive Execution During Experiment Runtime

```
# this value is different for every acquired picture !!!
index = ZenService.Analysis.Cells.ImageAcquisitionTime

if (index != lastindex):
    # increase position counter
    pos = pos + 1
    # get cell number for the current tile
    cn = ZenService.Analysis.Cells.RegionsCount
    # sum up the number of cells
    cells_total = cells_total + cn

    # stop if the desired cell number was already reached
    if (cells_total > 1200):
        ZenService.Actions.StopExperiment()

    # read the xy position of the current image
    posx = ZenService.Analysis.Cells.ImageStageXPosition
    posy = ZenService.Analysis.Cells.ImageStageYPosition
    # write data into log file
    logfile = ZenService.Xtra.System.AppendLogLine(str(pos)+";"+str(cn)+";"+str(cells_total)+";"+str(posx)+";"+str(posy))
    # update lastindex
    lastindex = index
```

Single Execution on Experiment Stop

```
ZenService.Xtra.System.ExecuteExternalProgram(script, "-f " + logfile)
```

Available Observables

Analysis Count_Cells_DAPI

Environment

Experiment

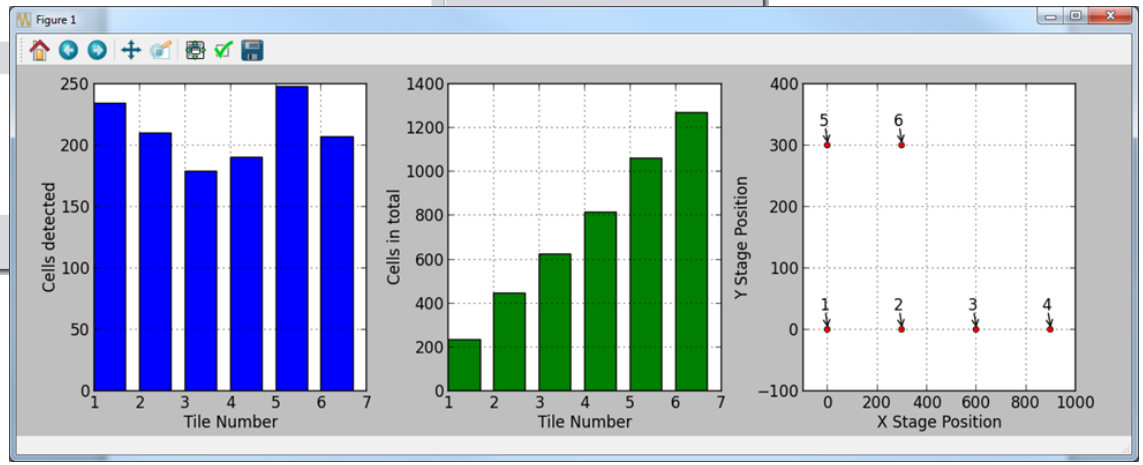
C:\Users\M1SRH\Documents\Testdata_Zeiss\Zen_Output\temp\Experiment-359_Log.txt - Notepad++

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

Experiment-359_Log.txt

```
1;234.0;234.0;0.0;0.0
2;210.0;444.0;300.0;0.0
3;179.0;623.0;600.0;0.0
4;190.0;813.0;900.0;0.0
5;248.0;1061.0;0.0;300.0
6;207.0;1268.0;300.0;300.0
```

length : 150 lines : 6 Ln : 6 Col : 27 Sel : 0 | 0 Dos\Windows ANSI INS



Internet Support

OAD Forum: http://www.zeiss.com/zen-oad

CZI file format: http://www.zeiss.com/czi

OAD Forum

ZEISS

User Name | Password | Log in | Help | Register

Remember Me?

Advanced Search

What's New? | Forum | ZEN Homepage | SmartSEM Homepage | Imaging Newsletter | Imprint

New Posts

Forum

If this is your first visit, be sure to check out the **FAQ** by clicking the link above. You may have to **register** before you can post: click the register link above to proceed. To start viewing messages, select the forum that you want to visit from the selection below.

Carl Zeiss Microscopy Community

Welcome to the Carl Zeiss Microscopy Community.

Open Application Development (OAD) for ZEN (blue edition)

Threads / Posts | Last Post

You have a special application that demands functionality beyond ZEN? Use the integrated OAD (Open Application Development) environment of ZEN (blue edition). With OAD you create your own macro solution based on the well-established Python language. Benefit from the simple access to a vital set of ZEN functions and the ability to include libraries such as the .Net Framework. Join the OAD community to discuss macros and help other users to develop their ultimate solution.

Instrument Control Discuss macros to control the hardware functions of your ZEISS microscope	Threads: 8 Posts: 15	Remote microscope control by minimax109 06-05-2013, 02:10 PM
Image Acquisition Post your acquisition-related questions and macros here	Threads: 7 Posts: 7	Show camera setting parameters by Carl Zeiss Microscopy 5 04-15-2013, 09:34 AM
Image Handling *(1 Viewing)* Your place to discuss general handling of images and the CZI file format	Threads: 15 Posts: 19	Tile - Subimage by Carl Zeiss Microscopy 3 Yesterday, 03:02 PM
Image Processing Discuss questions and projects related to processing of imaging data here	Threads: 7 Posts: 7	Insert scalebar in all images... by Carl Zeiss Microscopy 5 04-15-2013, 10:07 AM
Measurement and Analysis Share your programming ideas for various measurement and analysis tasks here	Threads: 9 Posts: 9	Save object stage positions... by Carl Zeiss Microscopy 5 04-15-2013, 10:15 AM
ZEN (blue edition) Application Model Find inspirations how to access the ZEN application model	Threads: 3 Posts: 3	Write special folders in a... by Carl Zeiss Microscopy 3 02-19-2013, 12:26 PM
Miscellaneous Everything OAD-related that won't fit in the other categories: share general aspects of macro programming and discuss the OAD environment	Threads: 15 Posts: 52	Import Python Modules from... by tbaum Today, 01:54 PM
Documentation (read-only) Your place to begin if you are looking for OAD-related documentation to start your project. If you need documentation not covered here, please ask in the specific forum.	Threads: 5 Posts: 5	CZI file format: general... by Carl Zeiss Microscopy 2 05-16-2013, 01:05 PM

→ Backup

Experiment Feedback
→ Define Rules via Python Scripting

ZEISS

ZEN system 2012

File Edit View Acquisition Graphics Macro Tools Window Help

Workspace Zoom Reset

Design Light Workspace

Locate | Acquisition | Processing | Analysis | Reporting

Experiment Manager

Count_Cells_Tiles_DAPI

Smart Setup | Show all Tools

AF Find Focus | Set Exposure | Live | Continuous | Snap

Z-Stack ---
Tiles 16 Tiles
Panorama ---
Time Series ---

7.19 MB

All Channels per Tile | Start Experiment

Setup Manager
- Light Path Settings — Show All
- Automation

Experiment Feedback
Enable | Script...

Acquisition Parameter
- Acquisition Mode — Show All
- Channels — Show All
- Focus Strategy — Show All
- Focus Devices — Show All

Multidimensional Acquisition
- Experiment Designer — Show All
- Tiles — Show All
- Information on Experiment
- Auto Save

Script Editor for Experiment Feedback

Single Execution on Experiment Start

```
lastindex = 0
pos = 0 # just a counter
cells_total = 0 # total number of cells
imagename = ZenService.Experiment.ImageFileName
script = "display_results_tiles.py"
```

Repetitive Execution During Experiment Runtime

```
# this value is different for every acquired picture !!!
index = ZenService.Analysis.Cells.ImageAcquisitionTime

if (index != lastindex):
    # increase position counter
    pos = pos + 1
    # get cell number for the current tile
    cn = ZenService.Analysis.Cells.RegionsCount
    # sum up the number of cells
    cells_total = cells_total + cn

    # stop if the desired cell number was already reached
    if (cells_total > 1200):
        ZenService.Actions.StopExperiment()

    # read the xy position of the current image
    posx = ZenService.Analysis.Cells.ImageStageXPosition
    posy = ZenService.Analysis.Cells.ImageStageYPosition
    # write data into log file
    logfile = ZenService.Xtra.System.AppendLogLine(str(pos)+";"+str(cn)+";"+str(cells_total)+";"+str(posx)+";"+str(posy))
    # update lastindex
    lastindex = index
```

Single Execution on Experiment Stop

```
ZenService.Xtra.System.ExecuteExternalProgram(script, "-f " + logfile)
```

Accept | Cancel

Available Observables
- Analysis Count_Cells_DAPI
- Environment
- Experiment
- Hardware

Available Actions
- Experiment Actions
- Xtra

Editor Tools
- Examples & Templates

Validate Script
The script is okay.

Scaling 0,2 µm/px (Theoretic) Automatic | System Information Idle 0 % | Free RAM 5.38 GB Free HD 70.94 GB | CPU 4 % | Frame Rate: - fps | Pixel Value: - | Position: X: - Y: - | Storage Folder: C:\Users\M1SRH\Docume | User: m1srh | 09:51

Experiment Feedback
→ Integrate Online Analysis

Image Analysis Wizard - ZEN system 2012

Image Analysis Wizard

1 Classes

2 Frame

3 Automatic Segmentation

Back

Execute | Interactive

Root
Cells
SingleCell DAPI 2

Smooth: None

Sharpen: None

Minimum Area: 130

Threshold: Reset | Undo | Redo

Low 55 | High 243 | Invert | Full Range

Histogram

Threshold definition: Click | Automatic

Method: Otsu

Fill

Separate: None

Dimensions | Display

Zoom 100% 100 % Auto Fit

Tools Navigator Interpolation

Channels DAPI

Single Channel | Range Indicator | Quick Color Setup

Reuse

Analysis

Show Objects | Fill | Opacity

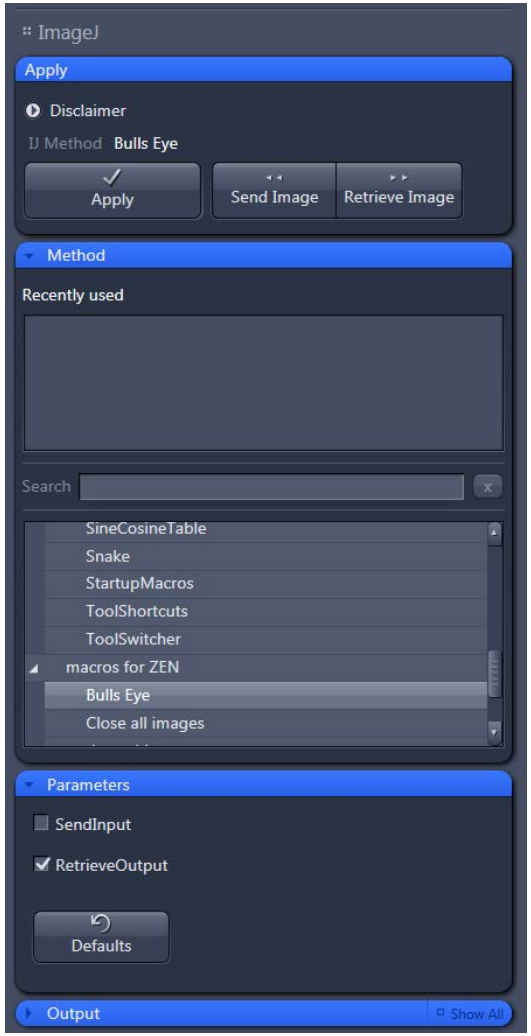
Show All Classes | Create Tables

Root 0
Cells 1
SingleCell DAPI 2

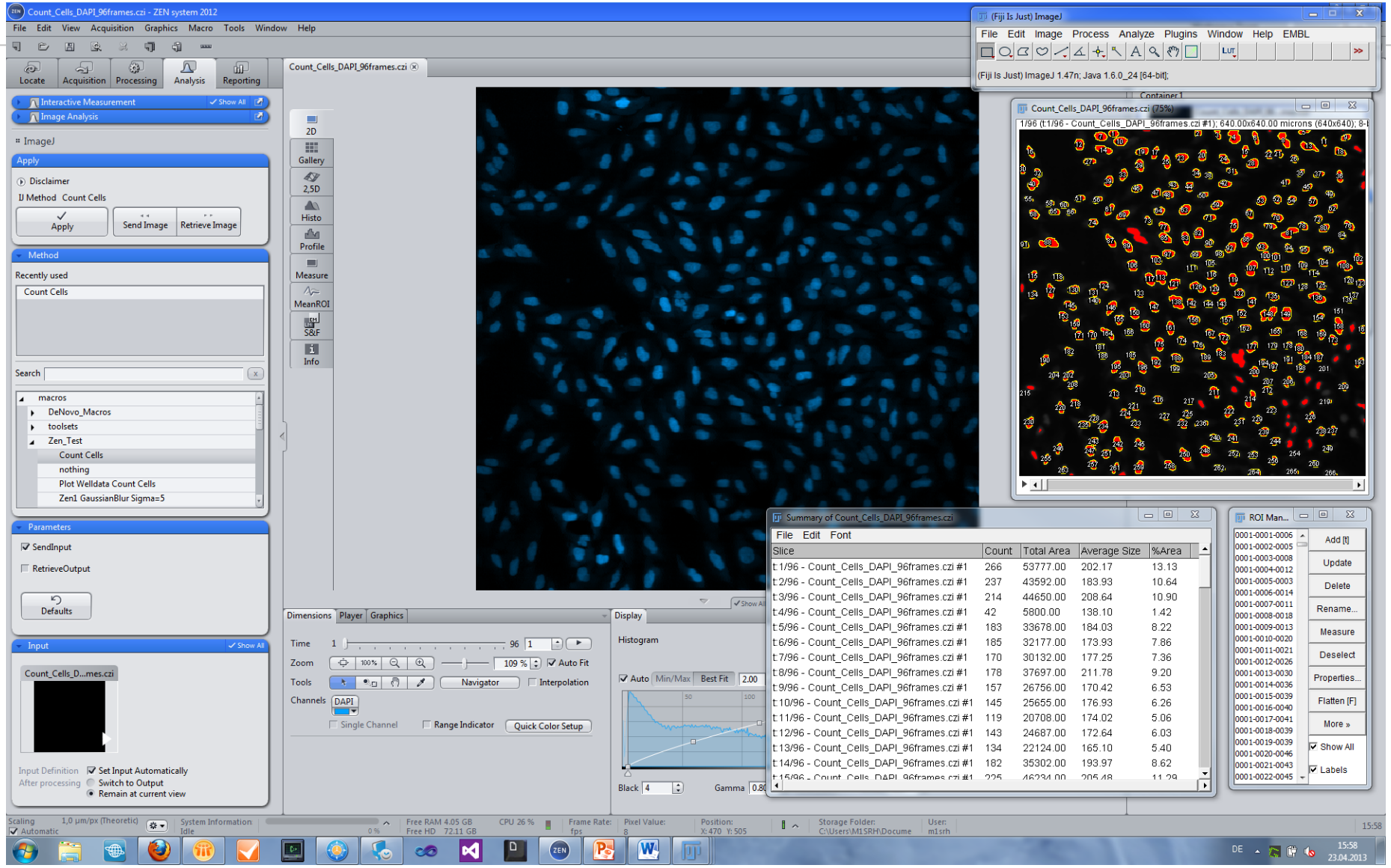
< Back | Next > | Finish | Cancel

Program Interaction

- ImageJ-Extension



OAD – Zen & Fiji as a „perfect combination“ at work



Open Application Development
→ multifaceted concept

1.) Documented file format

→ CZI

→ Bioformats reader

→ OME-TIFF

2.) Customization of ZEN through Macros

→ Python macro language

3.) Connecting ZEN with other software solutions

→ COM / TCP-IP

4.) Image J plugin

→ interaction with Image J / Fiji directly within ZEN

5.) Experiment Feedback

→ modify experimental protocol through online image analysis