AutoTx
(and more...)

Smart automatic background
data transfer for Windows
AutoTx

Smart automatic background data transfer for Windows

Windows

- Service based on .NET 4.5
- Targets an ActiveDirectory environment
AutoTx

Smart automatic background data transfer for Windows

Data Transfer

- Source: *local* disk drive
- Target: *UNC path* (SMB network share)
AutoTx

Smart automatic background data transfer for Windows

Background

- System service
- Autonomous
- Resource Monitoring
AutoTx

Smart automatic background data transfer for Windows

Automatic

- Queueing
- Suspend & resume
- Email notifications
AutoTx

Smart automatic background data transfer for Windows

Smart

- Constant monitoring
- Suspending transfers
AutoTx - General setting and motivation

Goal: take away the transfer task from the user & make it easy!
AutoTx - under the hood

- Service core
  - C# / .Net 4.5

- Transfer Task
  - RoboCopy

- Wrapper library for C#
  - RoboSharp (MIT license)
The microscope aspect

- How to tell the acquisition is done with a given file?
- With 100% accuracy...!?
Don’t mess with the photons!

Acquisition has top priority
AutoTx - User’s Perspective
User’s entry point is the corresponding Tray App
Usage - Tray App

- Current state
- What’s going on
Usage - Tray App

- Current state
- What’s going on
- Initiate new transfers
Usage - Tray App

- Current state
- What’s going on
- Initiate new transfers
- Real-time progress
**AutoTx**

**Notifications**

**Configuration**

**Cleanup**

**Updates**

---

**Configuration**

- Simple XML
- Minimal parameter set
- Two “layers”

```xml
<ServiceConfig>
    <!-- IncomingDirectory: directory on SourceDrive to watch for new files -->
    <IncomingDirectory>ProgramData\AUTOTRANSFER\INCOMING</IncomingDirectory>
    <!-- ManagedDirectory: dir on SourceDrive where folders are queued for transfer ('PROCESSING') and stored after the transfer ('DONE'). -->
    <ManagedDirectory>ProgramData\AUTOTRANSFER</ManagedDirectory>
    <!-- DestinationAlias: friendly name for the target to be used in mails -->
    <DestinationAlias>Core Facility Storage</DestinationAlias>
    <!-- DestinationDirectory: where files should be transferred to -->
    <DestinationDirectory>\fileserver.mydomain.xy\share</DestinationDirectory>
    <!-- TmpTransferDir: temporary directory relative to DestinationDirectory to be used for running transfers -->
    <TmpTransferDir>\AUTOTRANSFER-TMP</TmpTransferDir>
    <!-- MaxCpuUsage: pause transfer if CPU usage is above this value (in %) -->
    <MaxCpuUsage>25</MaxCpuUsage>
    <!-- MinAvailableMemory: pause transfer if free RAM is below (in MB) -->
    <MinAvailableMemory>512</MinAvailableMemory>
</ServiceConfig>
```
AutoTx

Notifications
Configuration
Cleanup
Updates

Cleanup

- Data moved to “grace” location
- No deletion!
- Notifications when expired
AutoTx

Notifications
Configuration
Cleanup

Updates
- Plenty of systems
- Several components
  - Service executables
  - Configurations
  - Message templates
  - Log files

→ Service Updater
AutoTx Next

- Entire code base is GPLv3
  - [github.com/imcf/auto-tx](https://github.com/imcf/auto-tx)

- Planned features
  - Multiple target locations
  - Network credentials
  - OMERO imports
What else?

The “more” part...
Bio-Formats via Python

- Bio-Formats
- scikit-image
- matplotlib

→ 3D stacks
Stacks with **bokeh**

```python
1 def update(zpos, np_slices, palette):
2     '''Update the figure with new settings.'''
3     zpos : int
4         The new z-position in the stack.
5     np_slices : int
6         The amount of slices to use for the maximum intensity projection.
7     palette : str
8         The palette to use for displaying the image.
9
dz = int(np_slices / 2)  # define dz
if dz == 0:
    i.data_source.data['image'] = [ch0[zpos, ...]]
else:
    lower = zpos - dz if zpos > dz else 0
    upper = zpos + dz if zpos + dz > dz else dz
    i.data_source.data['image'] = [ch0[lower:upper, ...]].max(axis=0)

p = figure(title='name', x_range=(0, dimx), y_range=(0, dimy))
1 p.image(image=[ch0[int(dims/2), ...]],
2         x0=0, y0=0, dwimx, dhimy,
3         palette='Viridis256')

show(p, notebook_handle=True)
```

Niko Ehrenfeuchter - Imaging Core Facility - University of Basel
OMERO-CellProfiler

→ See our Poster!
Plate-Data Quality Control

- Overview on plate wells
- Quick & Easy
  - Sharpness
  - Average Intensity
ImageJ ↔ OMERO

- Client side
- Sparse data
- ROIs
- Batching
Who we are...

Oliver Biehlmaier

Kai Schleicher

Laurent Guerard

Niko Ehrenfeuchter

Niko Ehrenfeuchter - Imaging Core Facility - University of Basel
Who we are - and where?