INSCOPER

User Guide

Imaging Software

v9.4

© 2026 Copyright INSCOPER. All rights reserved.

Table of contents

1. Imaging Software 5

1.1 Overview 5

1.1.1 Prerequisites 5

1.1.2 User Journey 5

1.1.3 Getting Started 6

1.2 Display Settings 7

1.2.1 Well Plate Editor 7

1.2.2 Snap & Live 8

1.2.3 Change Password 9

1.2.4 Project Name 9

1.2.5 Tiff Name 10

1.2.6 Metadata File Name 11

1.2.7 Logs 12

1.2.8 Crash Dump 12

1.2.9 Camera 13

1.2.10 FRAP 13

1.2.11 Memory 13

1.2.12 Image Display 14

1.2.13 Objectives 14

1.2.14 Wavelength 15

1.2.15 Preferences 15

1.3 Configuration 18

1.3.1 Overview 18

1.3.2 Camera Settings 19

1.3.3 Live Image Interaction 21

1.3.4 Motorized Device Control 26

1.3.5 System Calibration 27

1.4 Acquisition 36

1.4.1 Acquisition 36

1.4.2 Time 38

1.4.3 Multi-Channels 39

1.4.4 Positions 40

1.4.5 Tiling 44

1.4.6 Preview Feature 51

1.4.7 Multiwell Plate Mode 57

1.4.8 Z-stack 68

1.4.9 Software Autofocus 72

All rights reserved.

1.4.10 Multi-Sequence Acquisition 74
1.4.11 Data Processor 76
1.4.12 Saving Images and Metadata 82
1.4.13 Acquisition Designer 83

1.5 Visualization & Analysis 95

1.5.1 Visualization During Acquisition 95
1.5.2 Visualization After Acquisition 96
1.5.3 Interacting with Graphics 97
1.5.4 Image Processing 99
1.5.5 Data Export 100
1.5.6 Semi-automated feedback microscopy feature 103

1.6 Add-ons 106

1.6.1 Custom Python Scripting 106
1.6.2 Photomanipulation (ScanFRAP) 120
1.6.3 TIRF 126
1.6.4 LiveSR 128
1.6.5 liveRATIO 131
1.6.6 Confocal MAICO 133
1.6.7 Time Gated Domain FLIM (fastFLIM) 137

1.7 How-to guides 142

1.7.1 Use the Nikon Ti2 Water Dispenser 142

1.8 Notes 148

1.8.1 Opening OME-TIFF Files 148
1.8.2 Stitching Images in napari 152
1.8.3 Software Autofocus 158
1.8.4 Data Processor Algorithms 160

1.9 Support 165

1.9.1 Inscoper Software Version 165
1.9.2 Error Windows 165
1.9.3 Find log folder 166

1.10 Legal 168

1.10.1 Legal Information 168

All rights reserved.

All rights reserved.

1. Imaging Software

1.1 Overview

The **Inscoper Imaging Software** (Inscoper I.S.), powered by a proprietary hardware controller, provides an integrated interface for managing optical imaging workflows.

Inscoper I.S. offers research microscopists a unified interface to manage their optical imaging systems:

- Orchestrate multi-dimensional acquisition sequences.
- Control multiple cameras and manage data streams.
- Monitor hardware status in real time.
- Manage experimental configurations via reusable projects.
- Perform basic image processing and volumetric visualization.

1.1.1 Prerequisites

An Inscoper-certified installer assembled a hardware configuration for your system using the Configurator software. Now, your system is ready to perform your acquisition workflow.

If you want to perform minor hardware modifications (switch or add objectives, cubes), you can proceed with caution following this tutorial (part 3). For any other changes, do not modify the hardware configuration on your own at the risk of breaking image acquisition functions. Please contact us if you need advice.

1.1.2 User Journey

The Inscoper I.S. interface follows a guided workflow that helps new users get started quickly, while remaining efficient and enjoyable for experienced users.

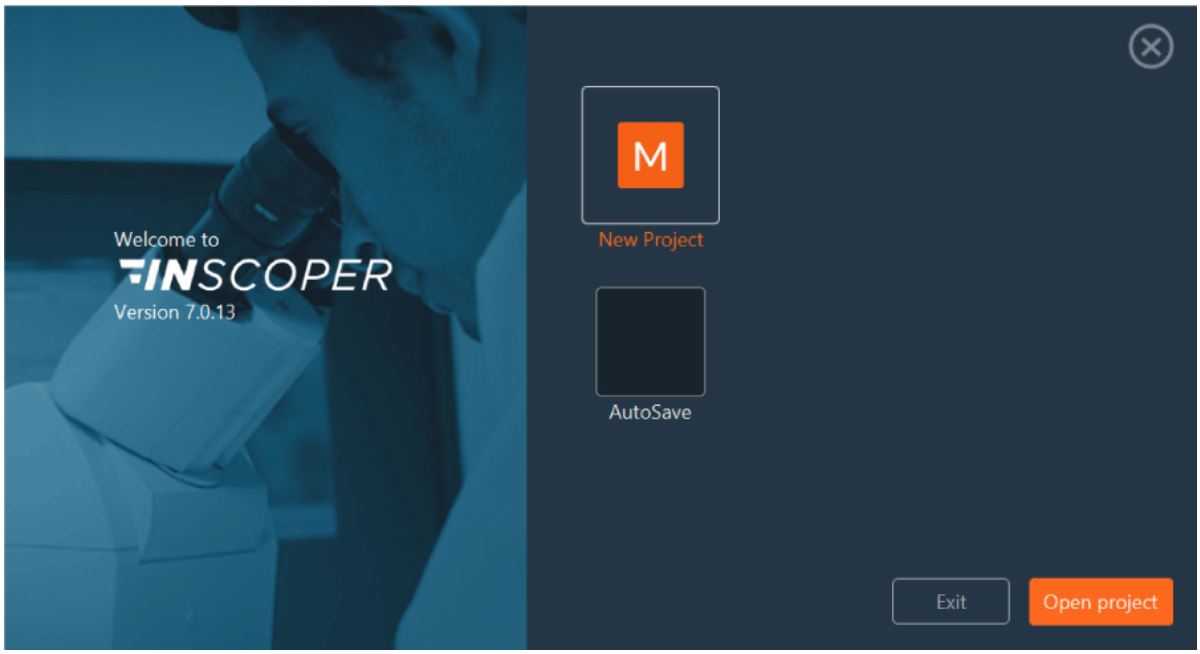


This journey follows three distinct phases:

1. Establish optical pathways, configure connected hardware devices, and define parameters (Configuration).
2. Construct an acquisition protocol by configuring programmable dimensions, or design a custom pattern (Multi-dimensional Acquisition / Designer).
3. Explore the resulting images using built-in visualization tools (Visualization).

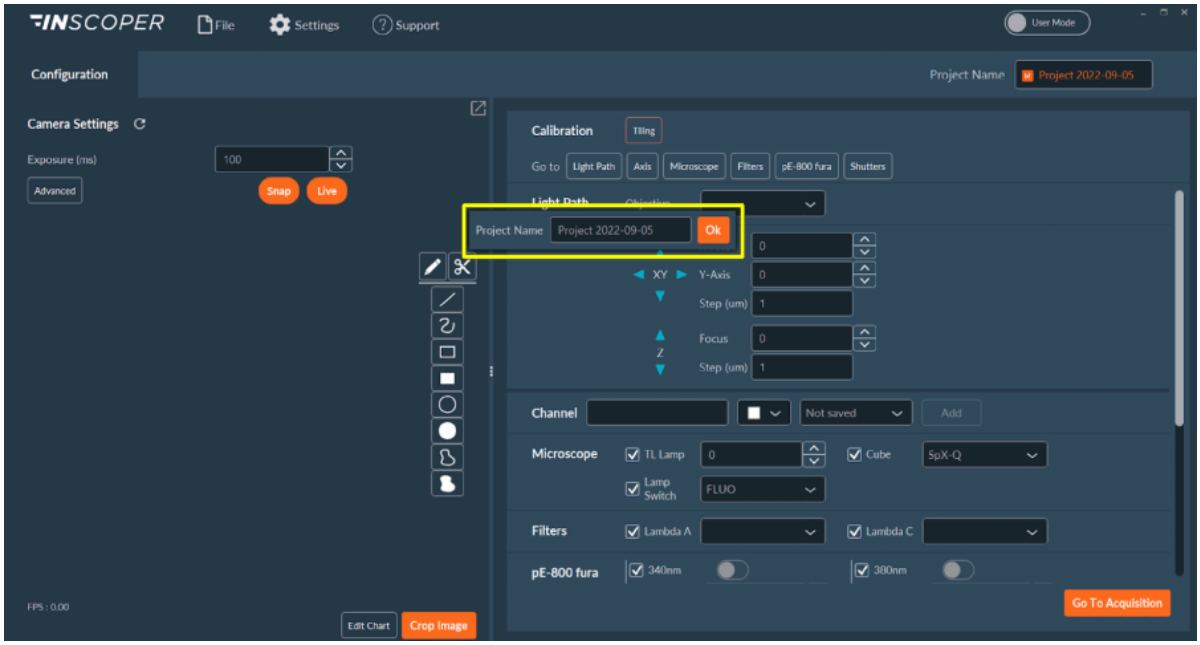
All rights reserved.

1.1.3 Getting Started



When you launch Inscoper I.S., the start window appears. The software version number is displayed, and you can choose from the following options to begin:

Command Button	Operational Description
New Project	Initializes a new working session.
Auto save	Recovers the most recent session state, including unsaved sequence parameters.
Open Project	Opens an existing project. Select your file (`.cbf`) from the directory tree, then click **Open**.



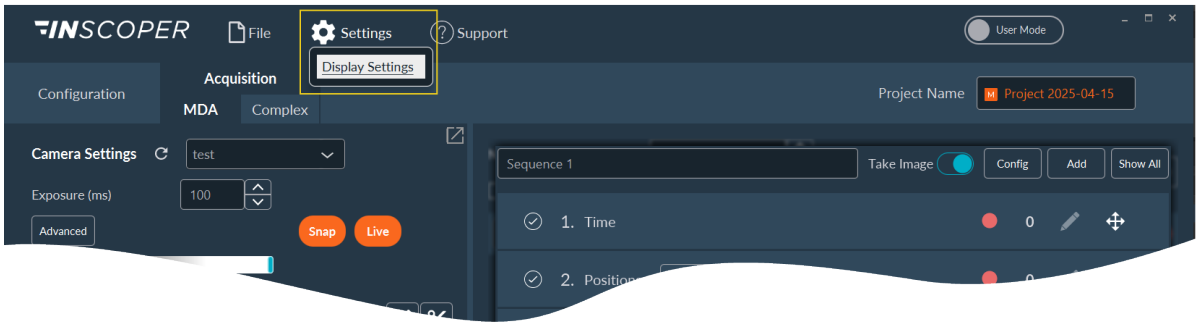
After you select the appropriate session setup, the software probes and initializes the connected hardware. You are then prompted to name the new project. If the name field is left blank, the system defaults to the current date, as shown in the image above.

2026-09-08

All rights reserved.

1.2 Display Settings

Access interface preferences by navigating to **Settings** > **Display Settings**.



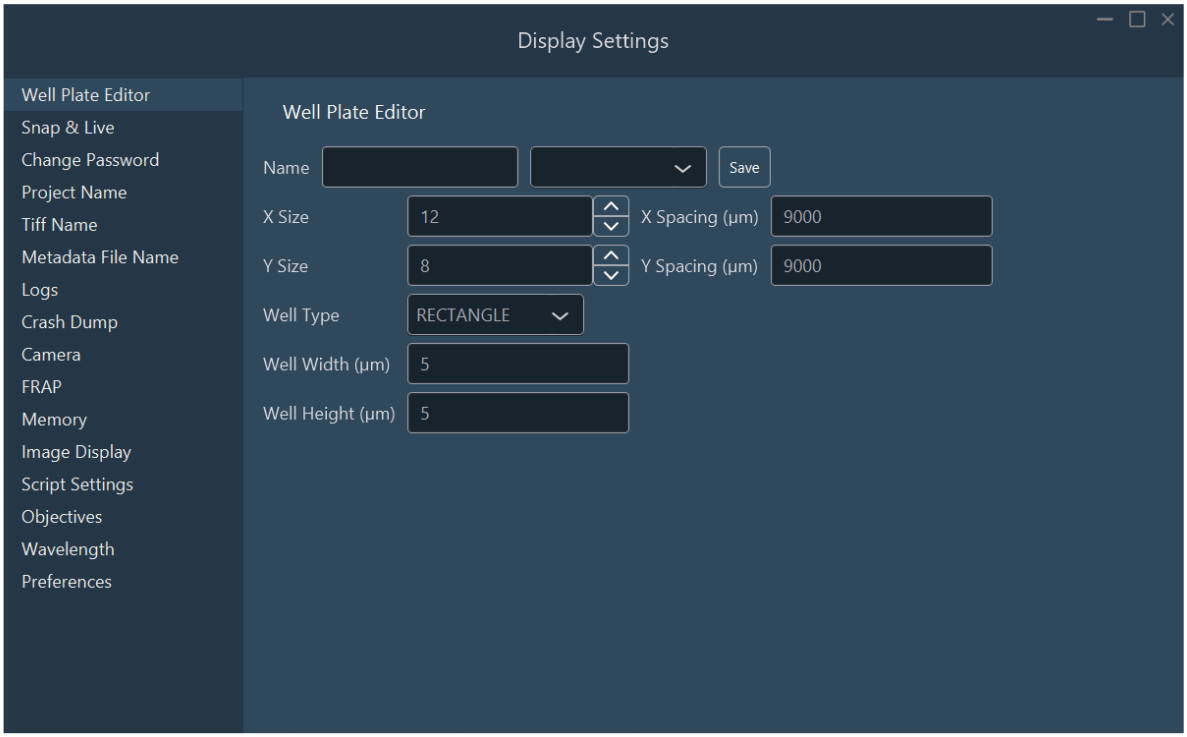
Use this window to configure software behavior, storage paths, and interface elements. Each tab in the left-hand list is documented below under the name shown in the application.

- Well Plate Editor
- Snap & Live
- Change Password
- Project Name
- Tiff Name
- Metadata File Name
- Logs
- Crash Dump
- Camera
- FRAP
- Memory
- Image Display
- Objectives
- Wavelength
- Preferences

1.2.1 Well Plate Editor

Construct and edit multi-well plate configurations used during acquisition:

All rights reserved.



- **Name**: Assign a name to the template.
- **Edit**: Select an existing well plate configuration from the drop-down menu to modify and save its parameters.
- **Grid**: Define the number of horizontal (columns) and vertical (rows) wells, labeled X Size and Y Size respectively.
- **Spacing**: Define the distance (pitch) between the wells (in µm).
- **Type**: Select the well shape from the drop-down menu (e.g., circle, square, or rectangle).
- **Well Width/Height**: Define the internal dimensions of the individual wells (in µm).
- **Save**: Save this well plate configuration to the database.

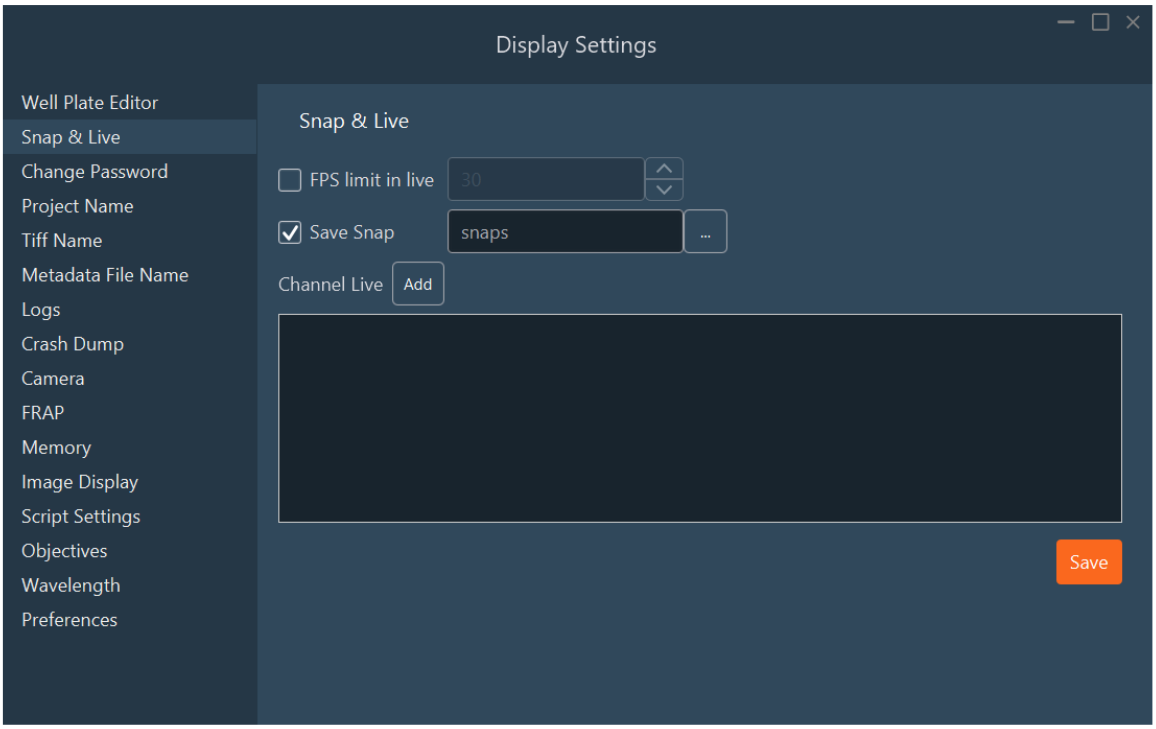
Multi-slide holders

Define a custom multi-slide holder by describing each glass slide as a large rectangular well in a 1×4 or 1×8 grid.

1.2.2 Snap & Live

Configure default parameters for the Snap and Live imaging modes:

All rights reserved.

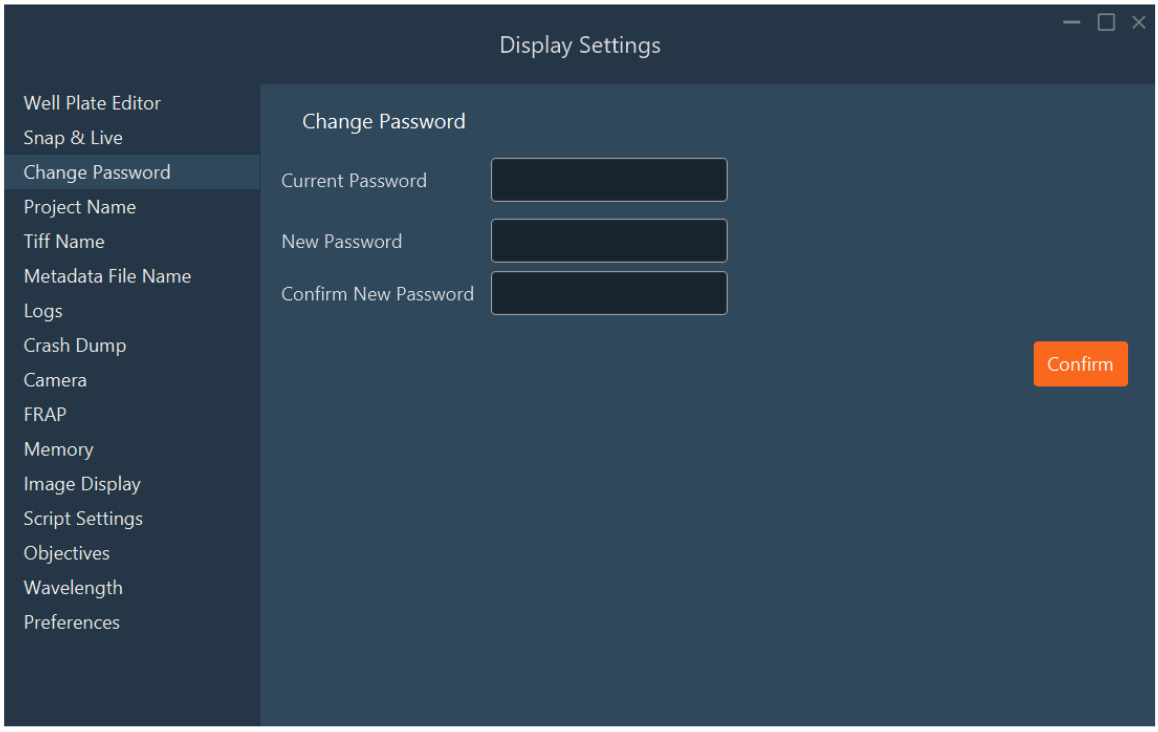


Available parameters:

- **Maximum frame rate (FPS)**: Sets an upper limit on the frame rate during Live streaming.
- **Default save directory**: Specifies the output directory for Snapshots.
- **Default startup channel**: Selects the channel that automatically engages when Live mode is toggled.

1.2.3 Change Password

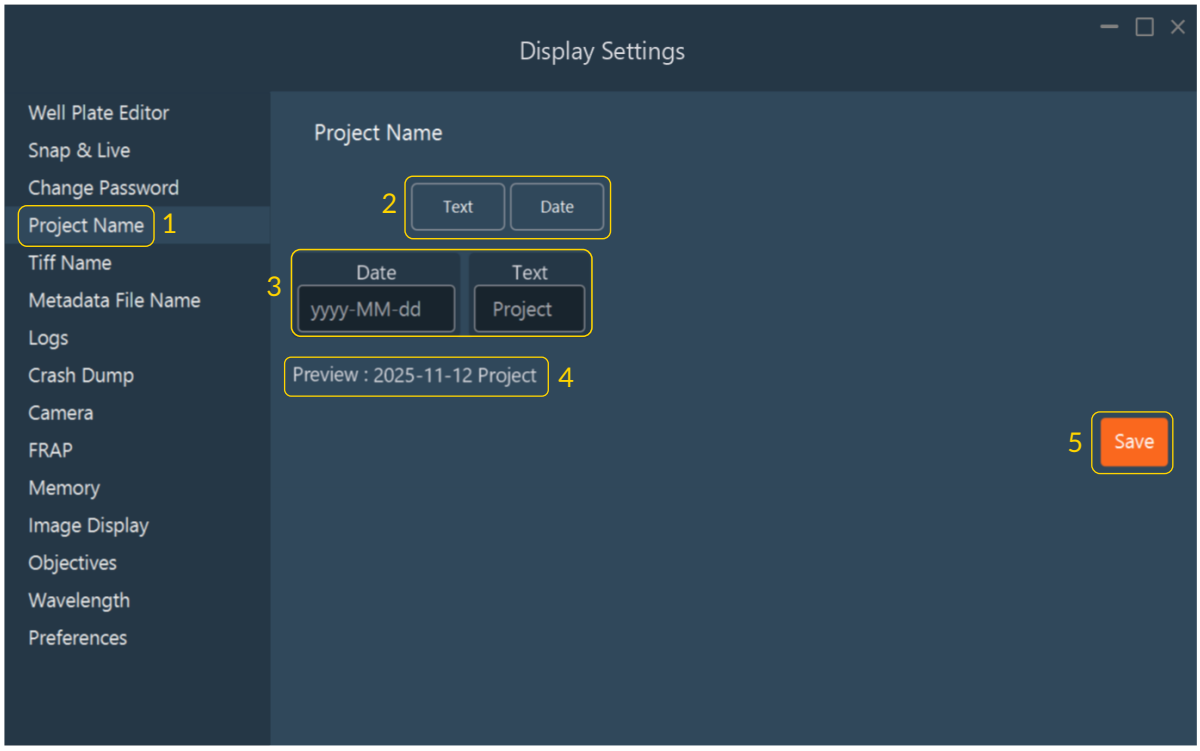
Define or update the password required to access **Expert Mode**.



1.2.4 Project Name

Establish a standardized naming pattern for new projects.

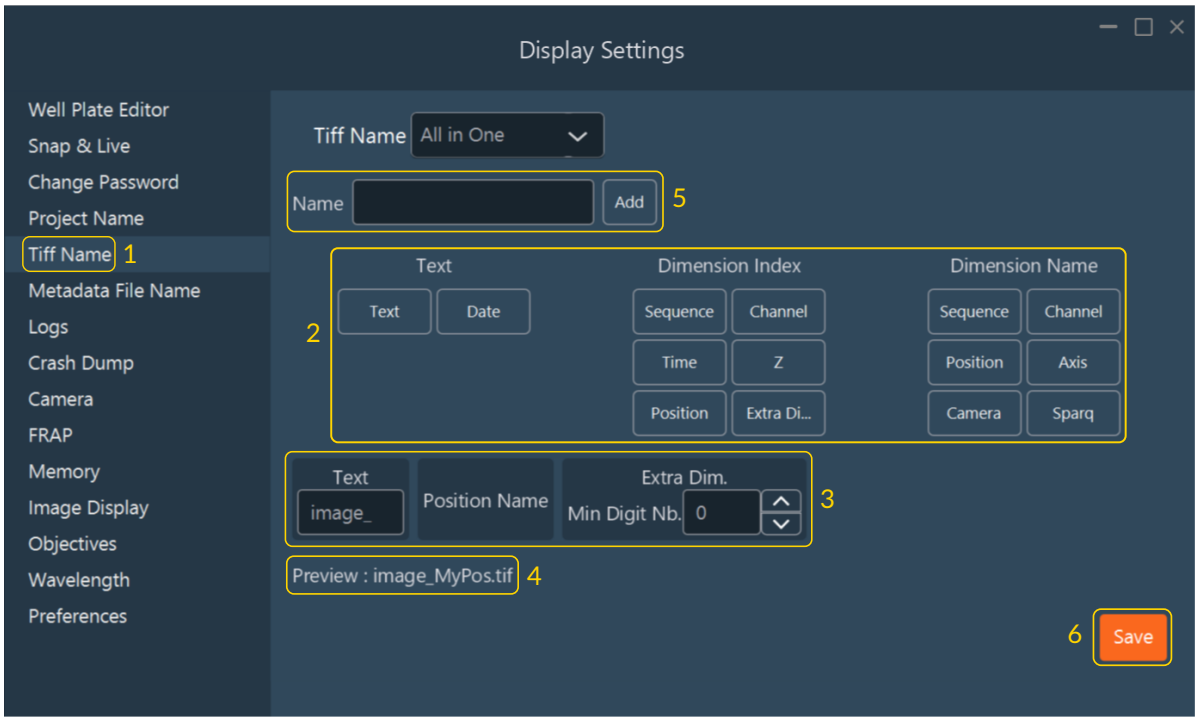
All rights reserved.



1. Select the **Project Name** tab.
2. Click the tags to add.
3. Modify the text and format of the selected tags in the line below. Organize the tags through drag and drop, or remove them with a right-click.
4. Verify in the **Preview** that the result matches the desired format.
5. Click **Save** to apply the settings.

1.2.5 Tiff Name

Customize how TIFF image files are named and exported.



By default, the software bundles all channels for a given position into a single multidimensional OME-TIFF file. Override this behavior to save each channel as a separate file.

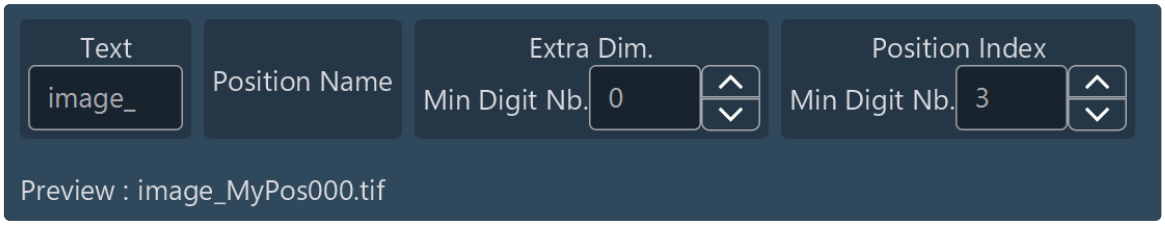
1. Select the **Tiff Name** tab.

All rights reserved.

2. Drag and drop metadata tags into the central construction area to build your naming convention (e.g., **text block + dimension name + dimension index**).

Index Zero-Padding

The **Min Digit Nb.** field determines the zero-padding length for the dimension index value (e.g., a value of 3 yields `001`, `002`).



1. **Name Construction Tool**: Displays a visual schematic of the filename structure. Move tags with drag and drop, or remove them with a right-click.
2. **Name Preview**: Displays an example of the generated filename.
3. Enter a preset name and click **Add** to save the convention.
4. Click **Save** to apply the settings.

1.2.6 Metadata File Name

Customize how metadata files are named and exported.

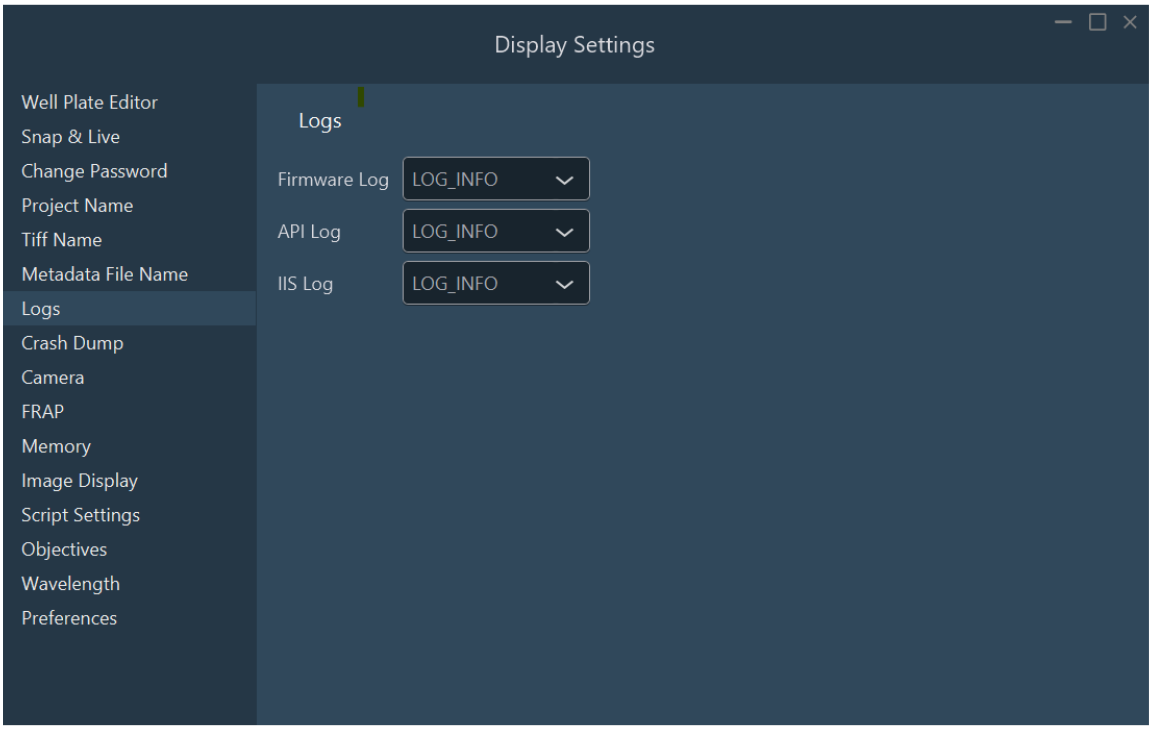


1. Select the **Metadata File Name** tab.
2. Click the tags to add.
3. Modify the text and format of the selected tags in the line below. Organize the tags through drag and drop, or remove them with a right-click.
4. Verify in the **Preview** that the result matches the desired format.
5. Enter a preset name and click **Add** to save the convention.
6. Click **Save** to apply the settings.

All rights reserved.

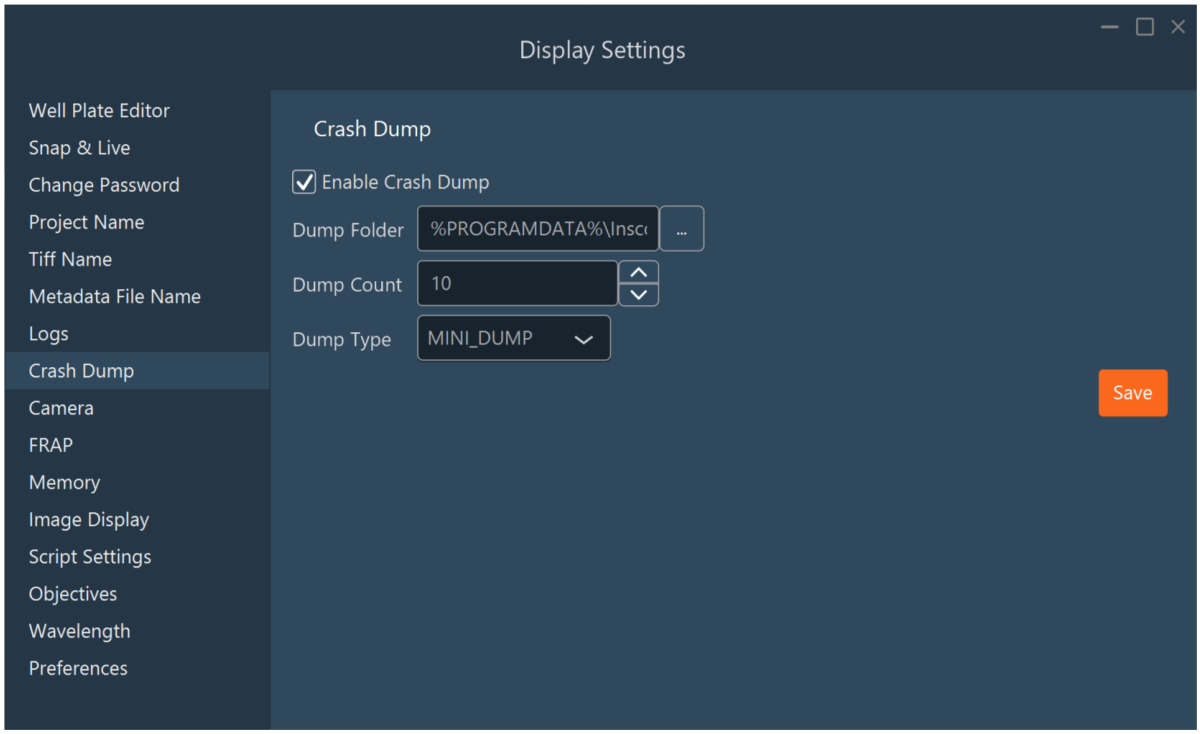
1.2.7 Logs

Configure the log verbosity level (e.g., Error, Warning, Info, or Debug).



1.2.8 Crash Dump

Configure crash reporting formats and retention policies.

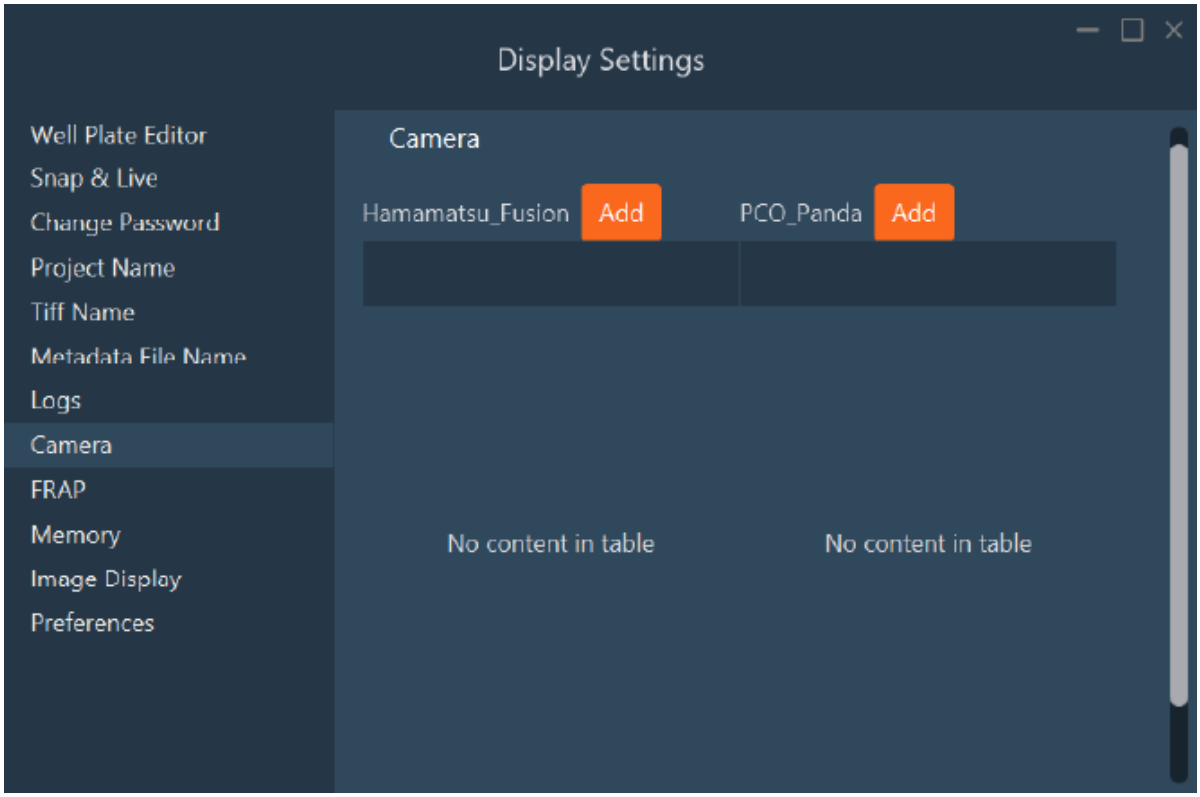


- **Dump folder**: Define the directory path where the system saves crash reports.
- **Dump count**: Set the maximum number of reports retained (defaults to 10; the system deletes older files automatically).
- **Dump Type**:
  - `MINI_DUMP` (Standard Default): Saves essential crash stack information, sufficient for most debugging.
  - `FULL_DUMP`: Saves the complete memory state. Takes longer and consumes significant disk space.

All rights reserved.

1.2.9 Camera

Manage connected cameras. For multi-camera setups that require field-of-view alignment, click **Add** to apply an image rotation to a camera.



1.2.10 FRAP

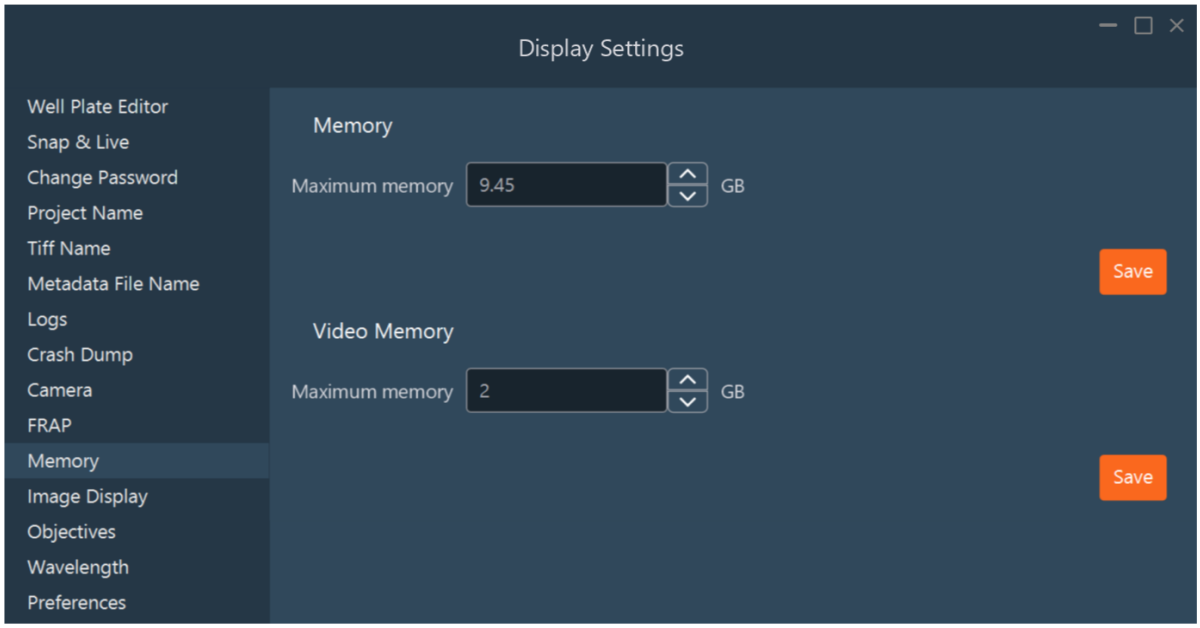
Review the FRAP hardware limits and set the virtual joystick sensitivity used during FRAP calibration.



1.2.11 Memory

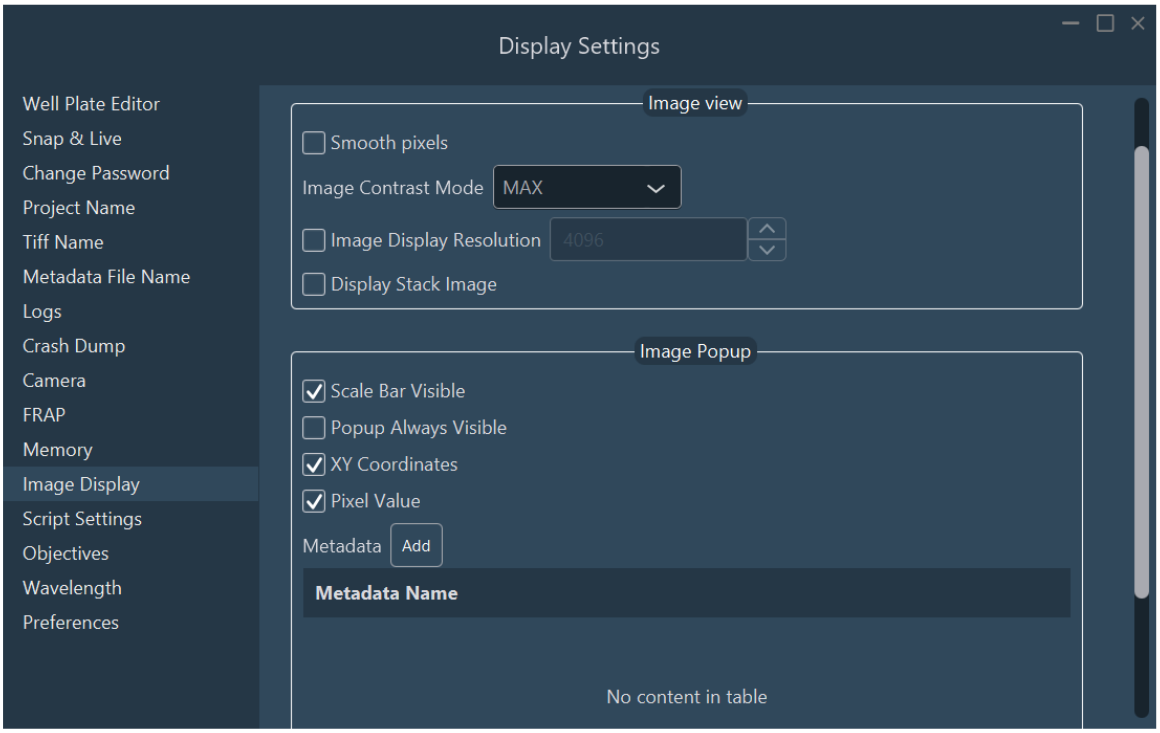
Set the memory limits for Inscoper I.S. **Memory** caps the system RAM available to the application, which defaults to 60% of the installed RAM. **Video Memory** caps the video buffer, which defaults to 2 GB.

All rights reserved.



1.2.12 Image Display

Configure the display preferences applied to the image viewers.



1.2.13 Objectives

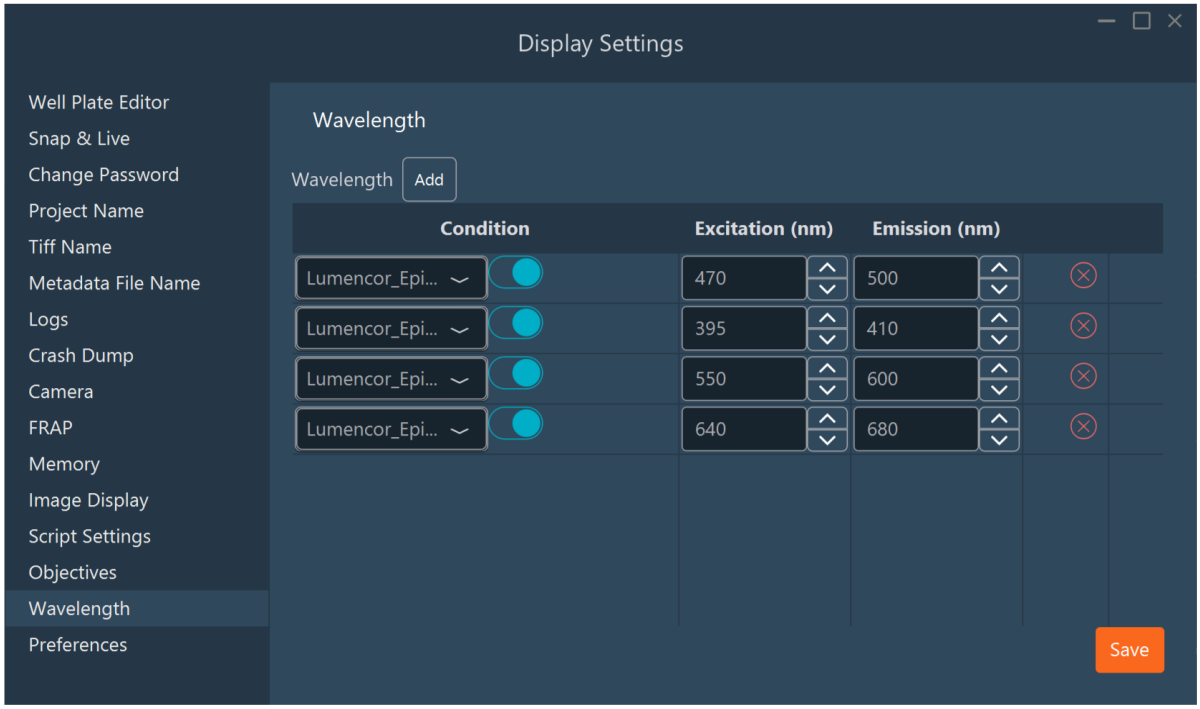
Define the **Numerical Aperture (NA)** and the immersion **Refractive Index** parameters for your objectives. The software uses these values for Nyquist sampling limit calculations and image processing.

All rights reserved.



1.2.14 Wavelength

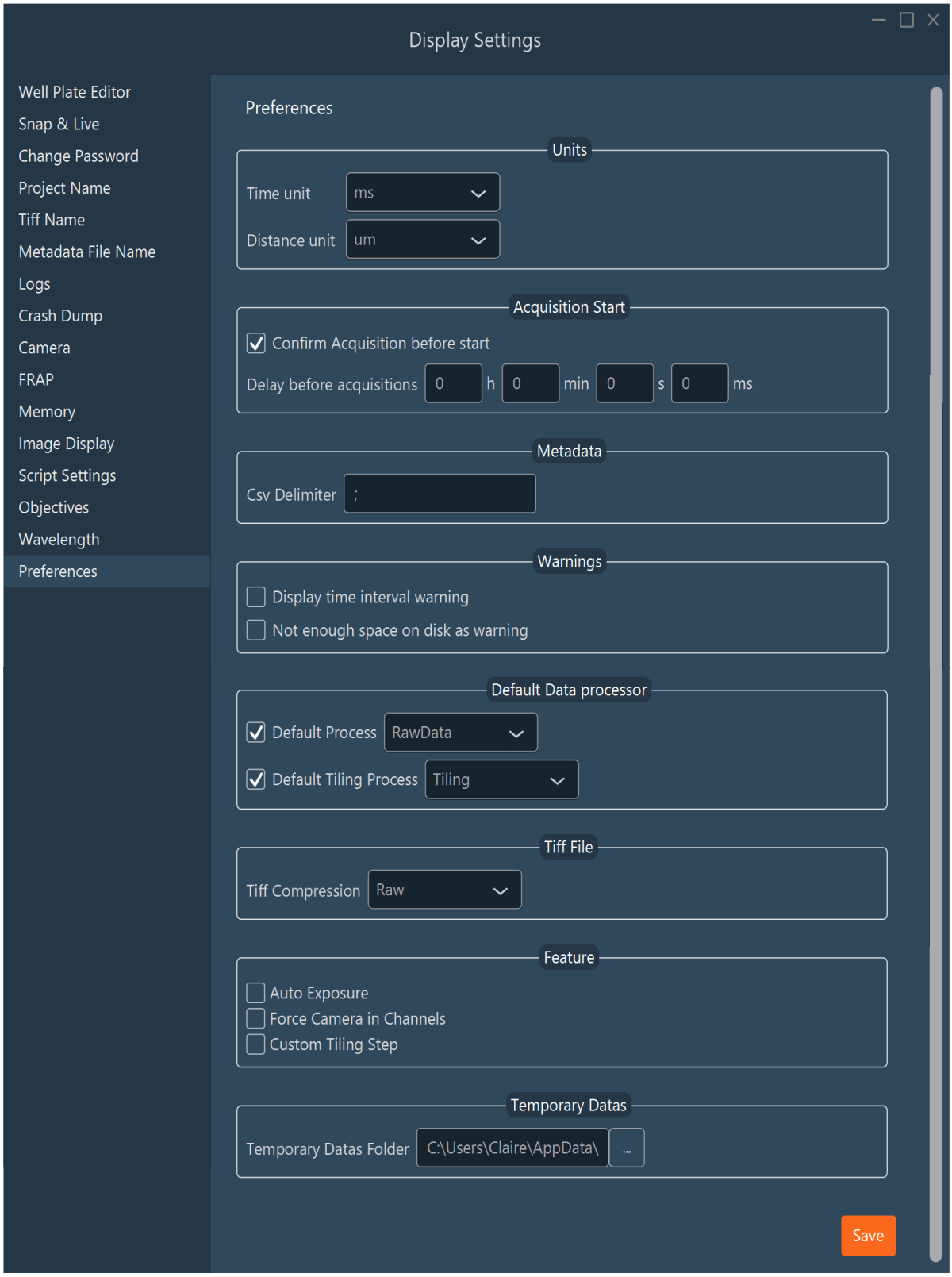
Define the **excitation and emission wavelengths** for your connected light sources, lasers, and filter cubes. The software uses this data for shading corrections, Nyquist sampling limits, and processing algorithms, and embeds it into the image OME-TIFF metadata.



1.2.15 Preferences

Configure general preferences, including measurement units, timing delays, warning-prompt behavior, and TIFF compression.

All rights reserved.



- **Auto exposure**: Enables the auto-exposure UI button on the primary dashboard.
- **Force camera in channels**: Associates specific cameras with specific channels, recommended for multi-camera or MAICO configurations.
- **Custom tiling step**: Define a translational overlap distance (in µm) rather than relying on a relative percentage for sequential tiling acquisitions.
- **Tiff Compression** (in the **Tiff File** group): Select the compression applied to the TIFF files written during acquisition.
    - **Raw** (default): Writes uncompressed pixel data. Provides the fastest saving and the widest compatibility.
    - **ZLib**: Applies lossless Deflate compression (TIFF compression code 8). Reduces file size at the cost of additional processing time during saving and opening. Compression efficiency depends on image content: low-noise images with large uniform areas compress well, whereas noisy images compress poorly.

All rights reserved.

Compatibility

ZLib-compressed files follow the standard TIFF Deflate specification and open in Fiji/ImageJ, Bio-Formats, and other TIFF readers that support Deflate compression.

2026-10-05

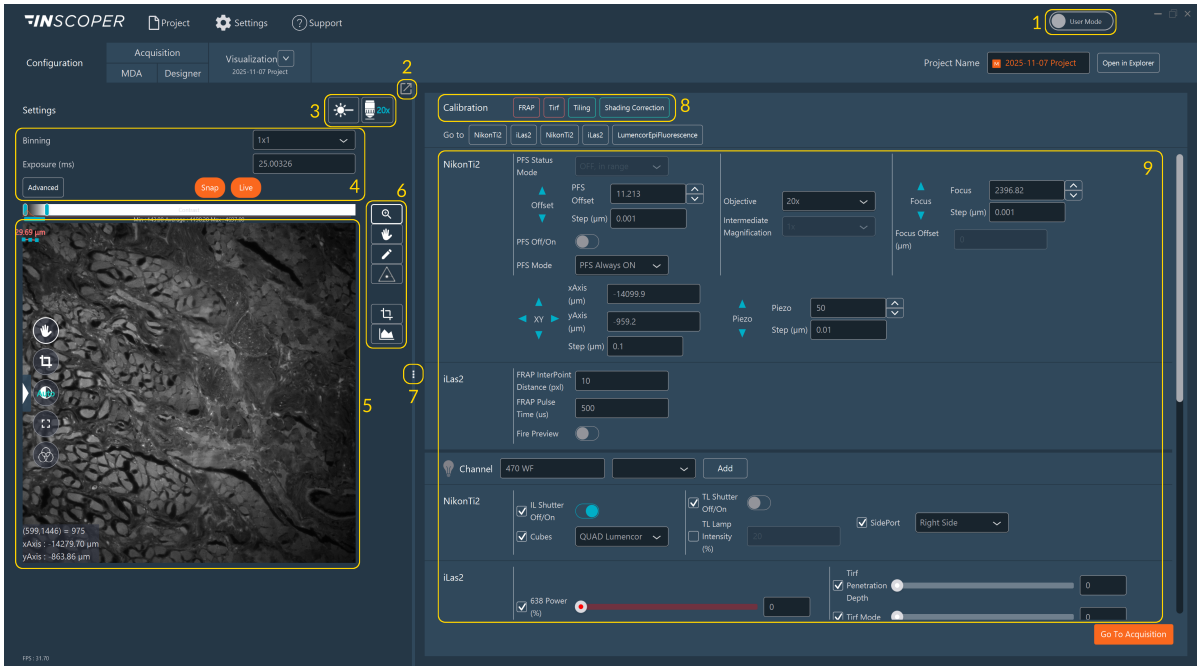
All rights reserved.

1.3 Configuration

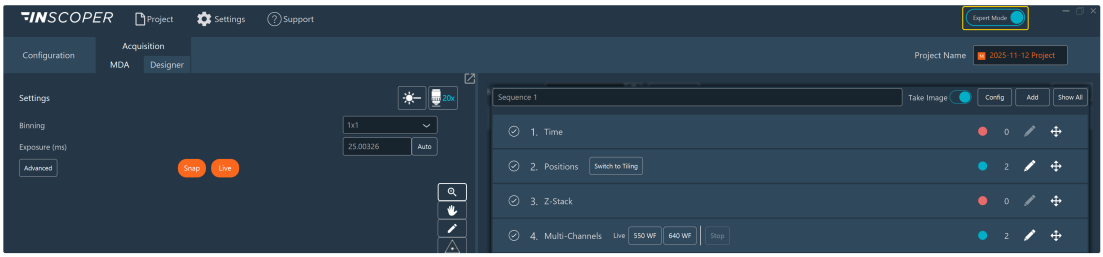
Use the **Configuration** tab to control each motorized device on the microscope, find the correct focus plane, select the optimal camera settings, and set up the channels.

1.3.1 Overview

The configuration interface is divided into several sections:



1. **User Modes**: Two access levels are available in Inscoper I.S., based on the user's level of expertise in microscopy:



Mode	Description
Expert Mode	**Expert Mode** provides unrestricted access to all system settings and parameters. This mode is intended for microscope facility managers and researchers experienced in microscopy.
User Mode	**User Mode** provides restricted access to selected settings and parameters. The restrictions are fully customizable, ranging from basic channel configuration to the most advanced camera and device settings. This mode is intended for biologists who are not yet experienced with microscopy.

> **Note**
>
> Users can switch from User Mode to Expert Mode at any time. Facility managers can set a password to restrict access to Expert Mode.

2. **Split Screen**: For dual monitors, detach the live camera feed view to a separate screen to decouple it from the settings panel.
3. **Channel/Objective Access**: Switch between defined channels and objectives with a single click.
4. **Camera Settings**: Configure camera parameters.
5. **Live Viewing**: View the real-time image feed.
6. **Image Tools**: Interact with the image display.

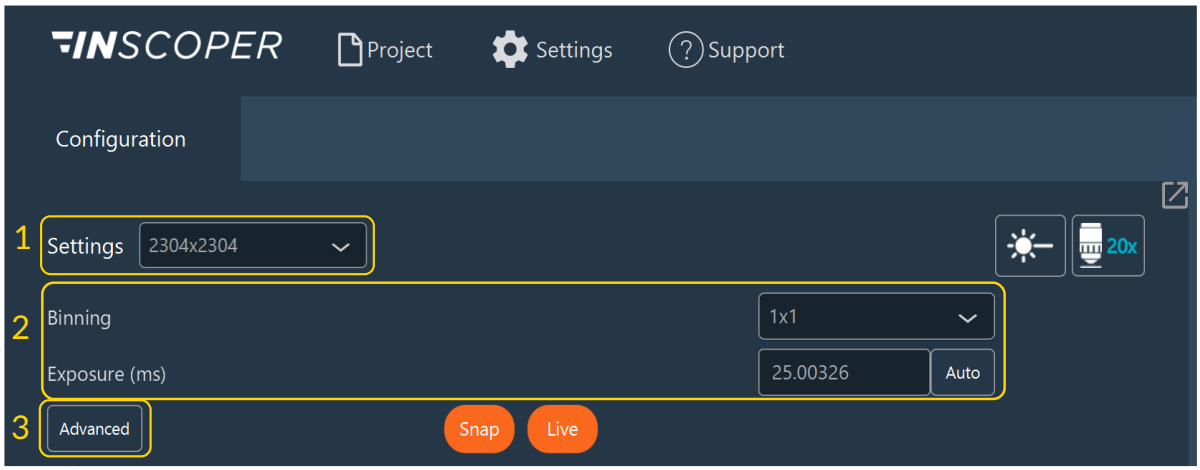
All rights reserved.

7. **Divider**: Adjust the screen size of the central camera view and the side settings panel.
8. **Calibration Protocols**: Access optical and stage calibration tools (see Tiling Alignment and Shading Correction).
9. **Motorized Devices**: Access connected motorized hardware, create new channels, and save them to the project.

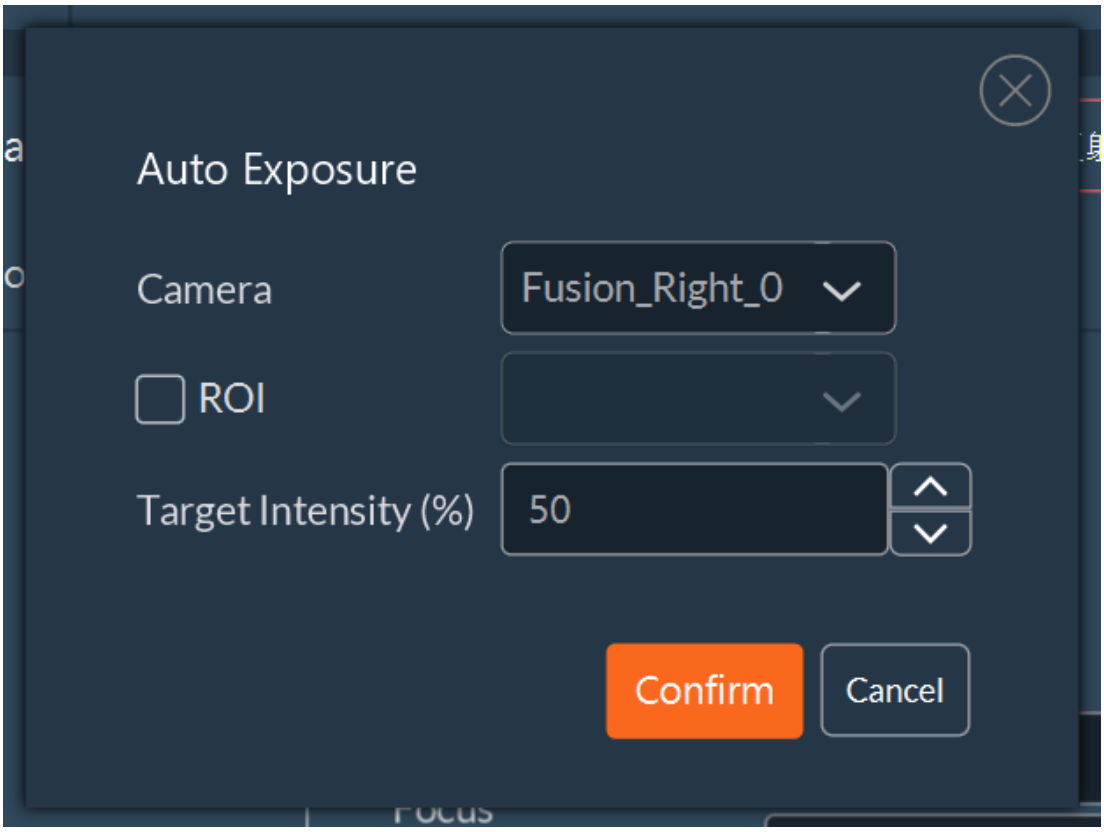
Rename the current project using the text field in the top-right corner.

1.3.2 Camera Settings

The software supports up to 4 cameras simultaneously. Create and save parameter presets for each camera using this section.

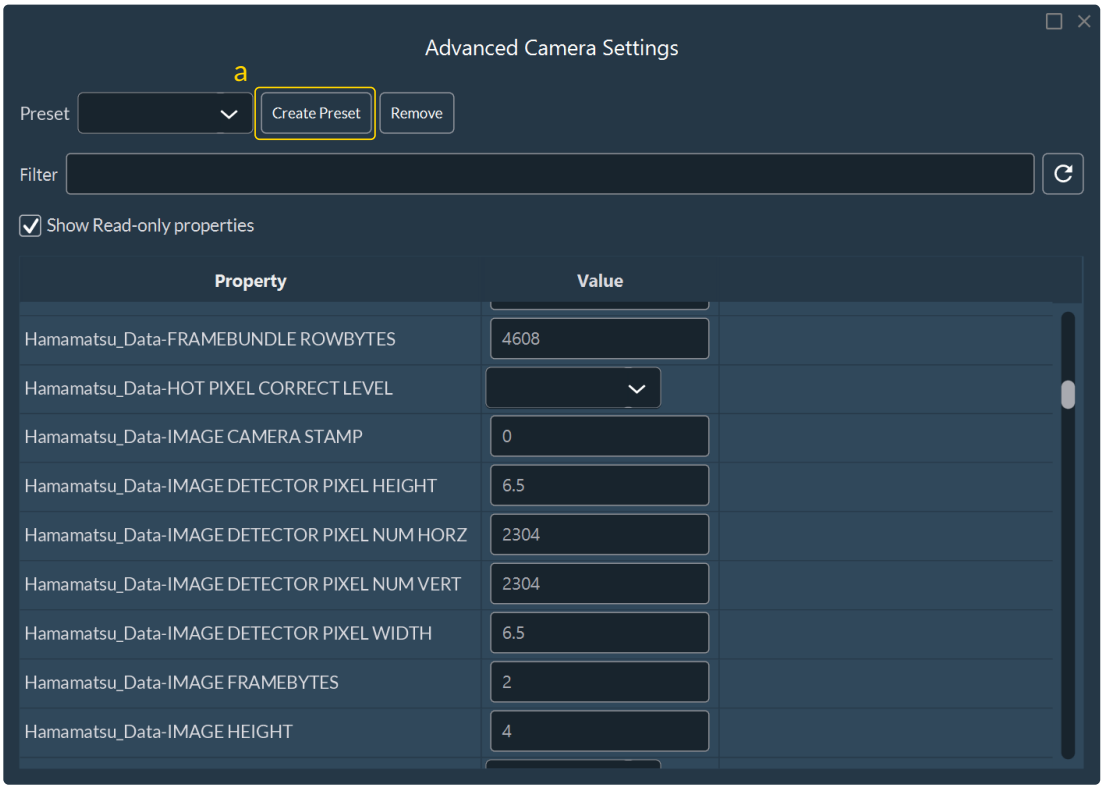


1. Select a camera preset from the drop-down list.
2. Adjust the exposure time and pixel binning. If **Auto Exposure** is enabled in the Display Settings, an **Auto** button is available. Auto Exposure adjusts the exposure time to prevent pixel saturation by analyzing the real-time pixel histogram.

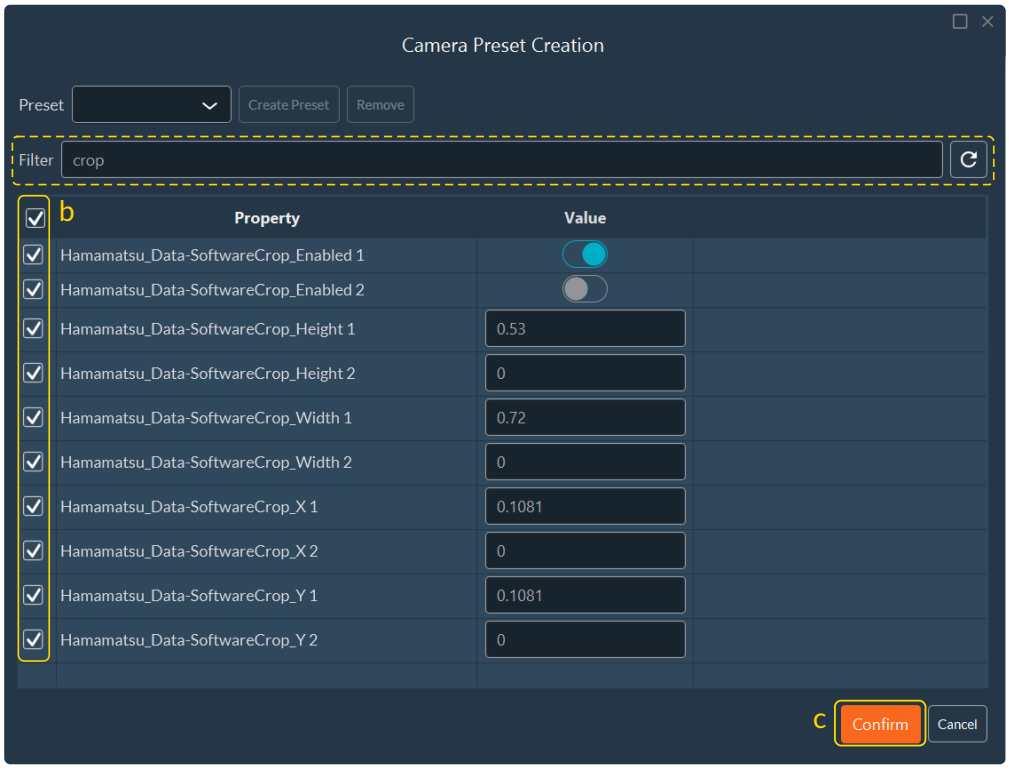


3. Access **advanced camera settings** to create custom presets:

All rights reserved.



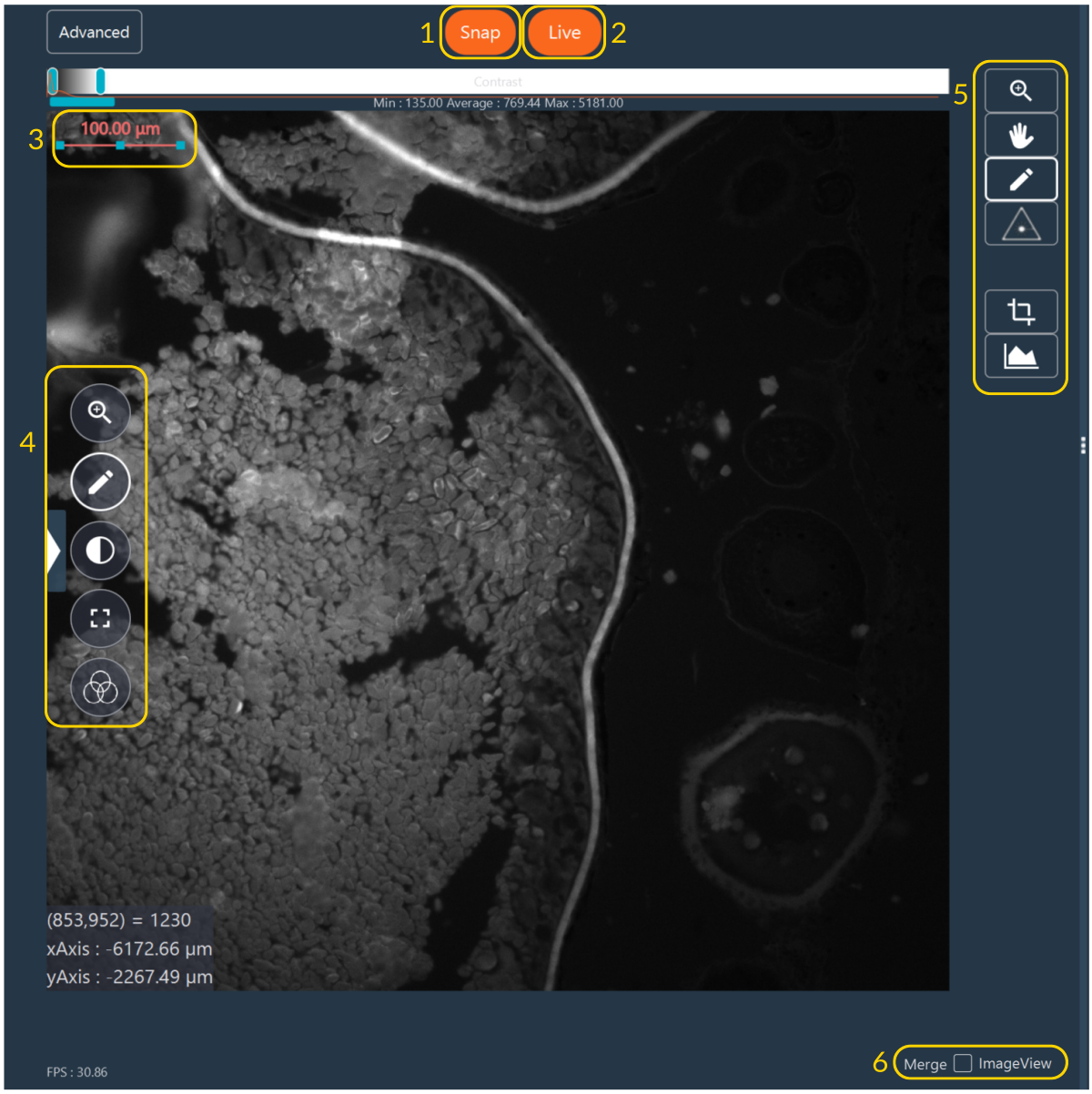
1. Click **Create Preset**:



2. Select the parameters to include in the new preset using the checkboxes. Use the search bar to filter the parameter list.
3. Click **Confirm**, then assign a name and save location for the preset.

All rights reserved.

1.3.3 Live Image Interaction



Use the toolbars to interact with the live image stream:

1. **Snap**: Capture a single image frame.
2. **Live**: Toggle the real-time camera stream.
3. **Scale Bar**: Double-click to modify orientation and physical length.

All rights reserved.

4. Tools to interact with the image:

Tool	Description
	Toggle automatic or manual contrast adjustments. Use the intensity sliders above the camera view to establish manual contrast boundaries.
	Toggle full-screen mode. Exit by pressing the button again or clicking the 'X'.
	Apply LookUp Tables (LUTs) to the image display in real time. Options include no LUT, single-color LUT, or a preset LUT.

LookUp Tables (LUTs)

Applying a LookUp Table maps raw grayscale intensity values to a specific colored gradient. This is critical for drawing ROIs accurately during quantitative assays, as pseudo-colored "heatmaps" (like the fire LUT) visually exaggerate subtle intensity differences that remain indistinguishable to the human eye in standard grayscale.

Available preset LUTs include:

- **Pixel indicator**: Highlights saturated (overloaded) pixels in red.
- **Inscoper ratiometric**: Designed for visualizing ratiometric images.
- **Multicolor LUTs**: Standard palettes such as "fire", "physics", etc.

5. **Advanced Image Tools** (a secondary toolbar with specialized functions):

Tool	Description
	Pan across the field of view by dragging and dropping. Scroll the mouse wheel to move along the Z-axis.
	Zoom digitally by scrolling the mouse wheel over the image.
	Draw Regions of Interest (ROIs) or crop the live image. Configure as a hardware sensor crop or software crop.



Create and manipulate ROIs to control processing, cropping, and photo-manipulation.

Creation Modes

Tool	Mode	Description
	**Pen**	Add one or multiple discrete ROIs to the view.
	**Scissors**	Remove a drawn sub-shape from an existing ROI, retaining the external boundary.

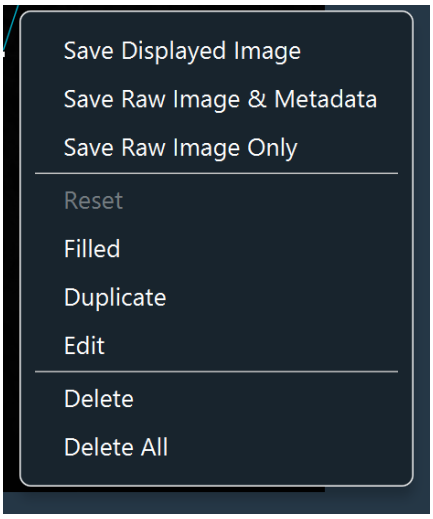
All rights reserved.

Drawing Geometries

Select the desired geometry tool to draw the ROI boundaries:

Tool	Shape	Description
	**Line**	Draw a 1D straight line.
	**Curve**	Draw a 1D freehand line.
	**Rectangle Edge**	Draw the 2D perimeter of a rectangle.
	**Filled Rectangle**	Draw a solid 2D rectangular area.
	**Circle Edge**	Draw the 2D perimeter of a circle.
	**Filled Circle**	Draw a solid 2D circular area.
	**Free Form Edge**	Draw a custom closed 2D perimeter.
	**Filled Free Form**	Draw a solid, custom closed 2D area.

ROI Management



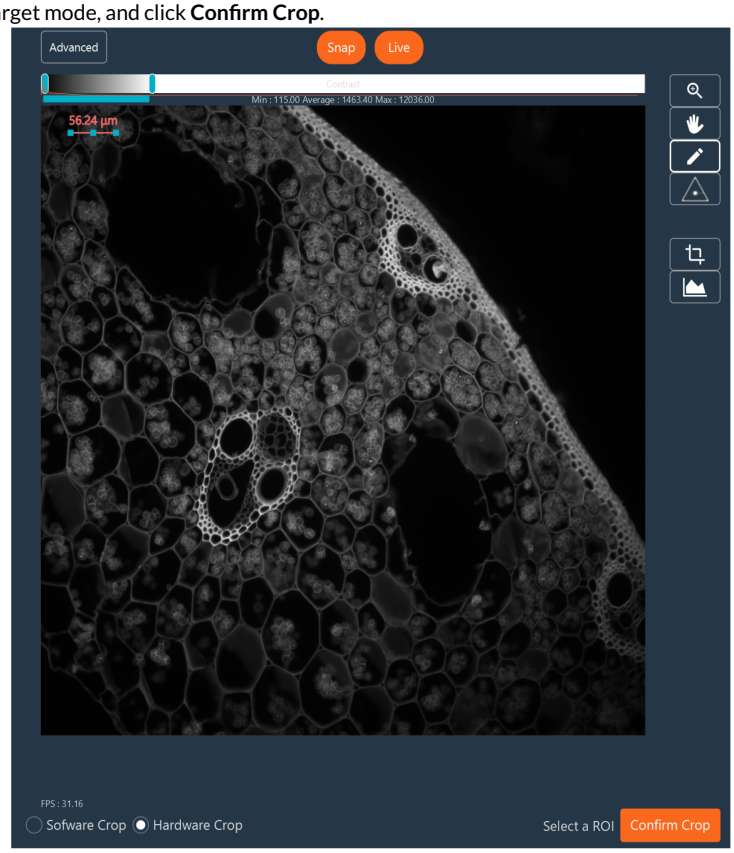
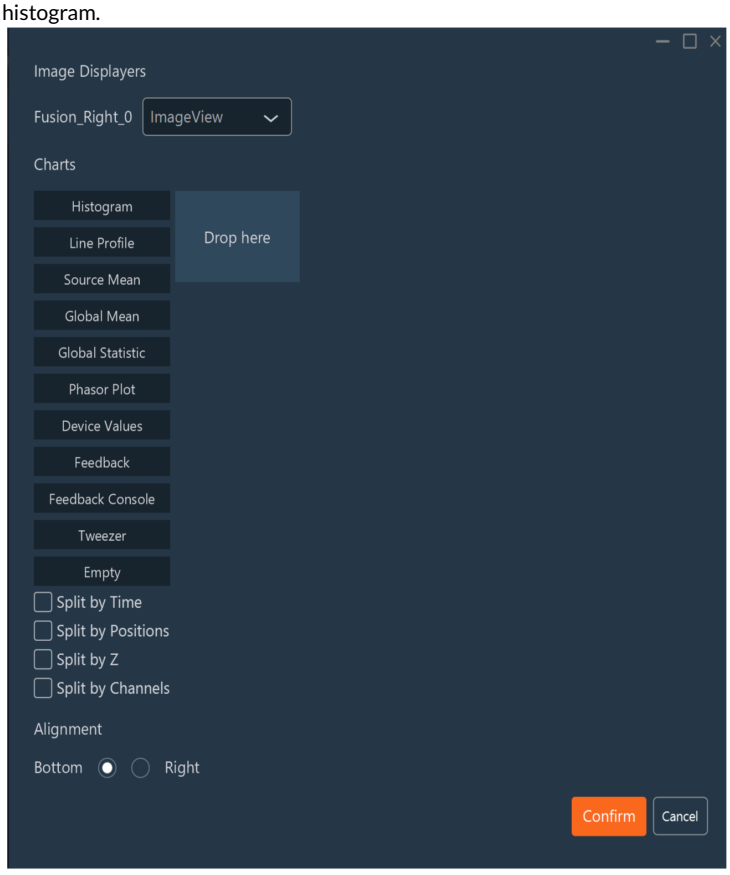
Right-click an existing ROI on the image canvas to access the spatial management window:

- Fill internal areas.
- Duplicate existing ROIs.
- **Edit ROI**: Enter the ROI dimensions in microns and the center coordinates within the camera field of view.

All rights reserved.

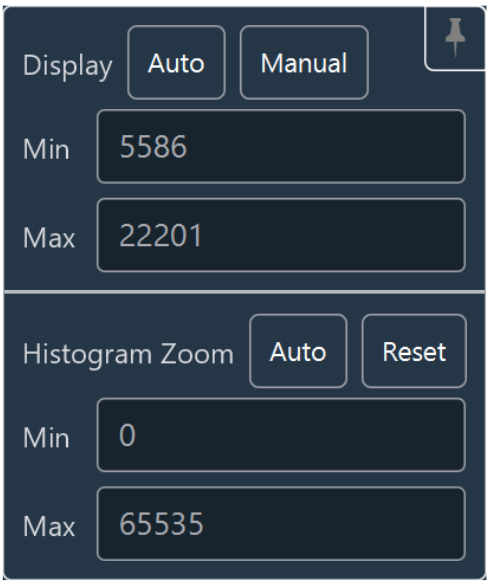
- Remove single or all drawn ROIs.

All rights reserved.

Tool	Plugin	Description
	**Fire on Click**	Trigger targeted photo-manipulation lasers. (Requires the FRAP module to be installed).
	**Crop Access**	Configure either **hardware** (sensor-level) or **software** (digital) cropping. Draw a ROI, select the target mode, and click **Confirm Crop**. 
	**Histogram Access**	Dock the quantitative histogram (or other 1D visualizations) below or to the right of the image panel. Adjust the X-axis scaling relative to camera bit-depth by hovering over the top of the histogram. 

All rights reserved.

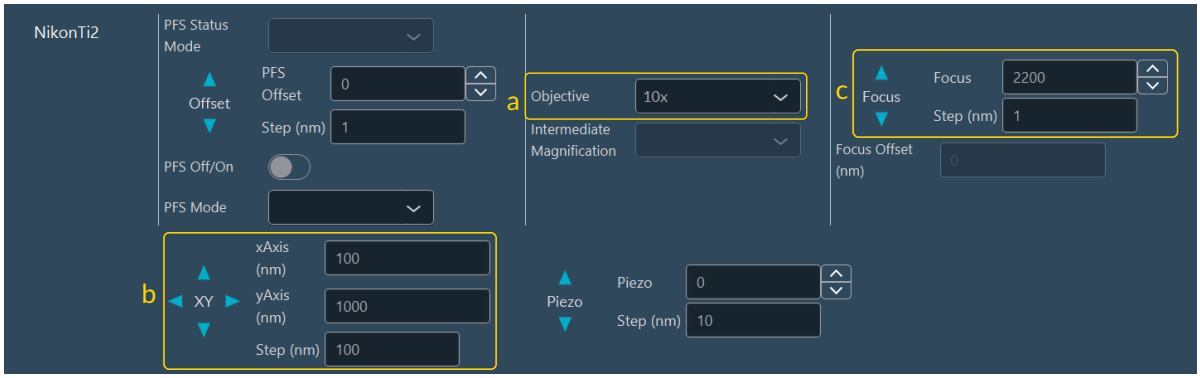
- **Contrast Adjustment**: Hover over the contrast bar located at the top of the image view. Use the high/low threshold sliders to modify the minimum and maximum displayed values, effectively redefining the black and white points to optimize the visual dynamic range. Zoom directly within this micro-histogram tool.



6. **Merged Images**: If the system is equipped with multiple cameras, use the **Merged images** toggle to display a combined live multiplexed superposition of all active camera streams.

1.3.4 Motorized Device Control

Interacting with Devices



This panel provides control over hardware connected to the Inscoper I.S., including the microscope stand, motorized XY stages, optical shutters, illumination sources, filter wheels, piezo Z-drives, and microfluidics pumps.

- Use panel **(a)** to move the objective turret.
- Panels **(b)** and **(c)** provide virtual joysticks to move the XY stage and Z-focus. Each arrow moves the stage or focus in the selected direction by the defined step.

All rights reserved.

Creating and Loading Channels



Expert users have direct control over optical path components (filter wheels, dichroic cubes, LEDs, laser lines). To configure a new fluorescent imaging channel:

1. Select the excitation light source.
2. Set its output intensity.
3. Select the optical filter cube.
4. Select the LightPath geometry.
5. (Optional) Select brightfield LED illumination to use instead of, or simultaneously with, the fluorescence path.
6. Assign a name to the channel configuration.
7. Save the channel:
    - **a. Temporarily** (for the current session) by clicking **Add** (**User Mode**).
    - **b. Permanently** (to the database) by clicking **Save** (**Expert Mode**).
8. The newly created channel populates the channel list.

Camera Port Alignment

Many microscopes use more than one camera port (e.g., left and right ports). It is critical to select the correct active port before initiating imaging. Failing to select the active port may result in blank images or hardware errors during acquisition.

FRAP channels

To create a channel dedicated to photomanipulation, see the FRAP Channel section. FRAP channels do not appear in the channel list shortcut above the image view on the left side.

1.3.5 System Calibration

Access critical optical and stage calibration tools required for various sequences.

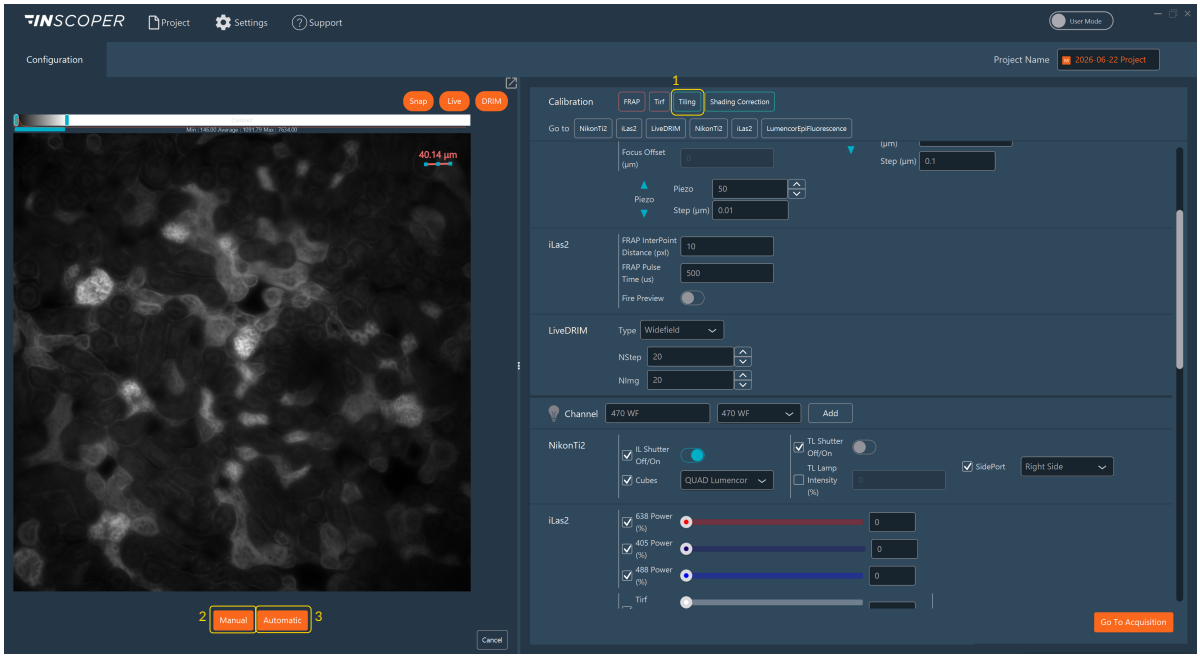
Tiling Alignment Calibration

Tiling calibration aligns the camera sensor's X/Y orientation with the motorized XY stage axes. This hardware alignment is strictly required for accurate image stitching.

Tiling Calibration

Each camera on the system requires this calibration.

All rights reserved.

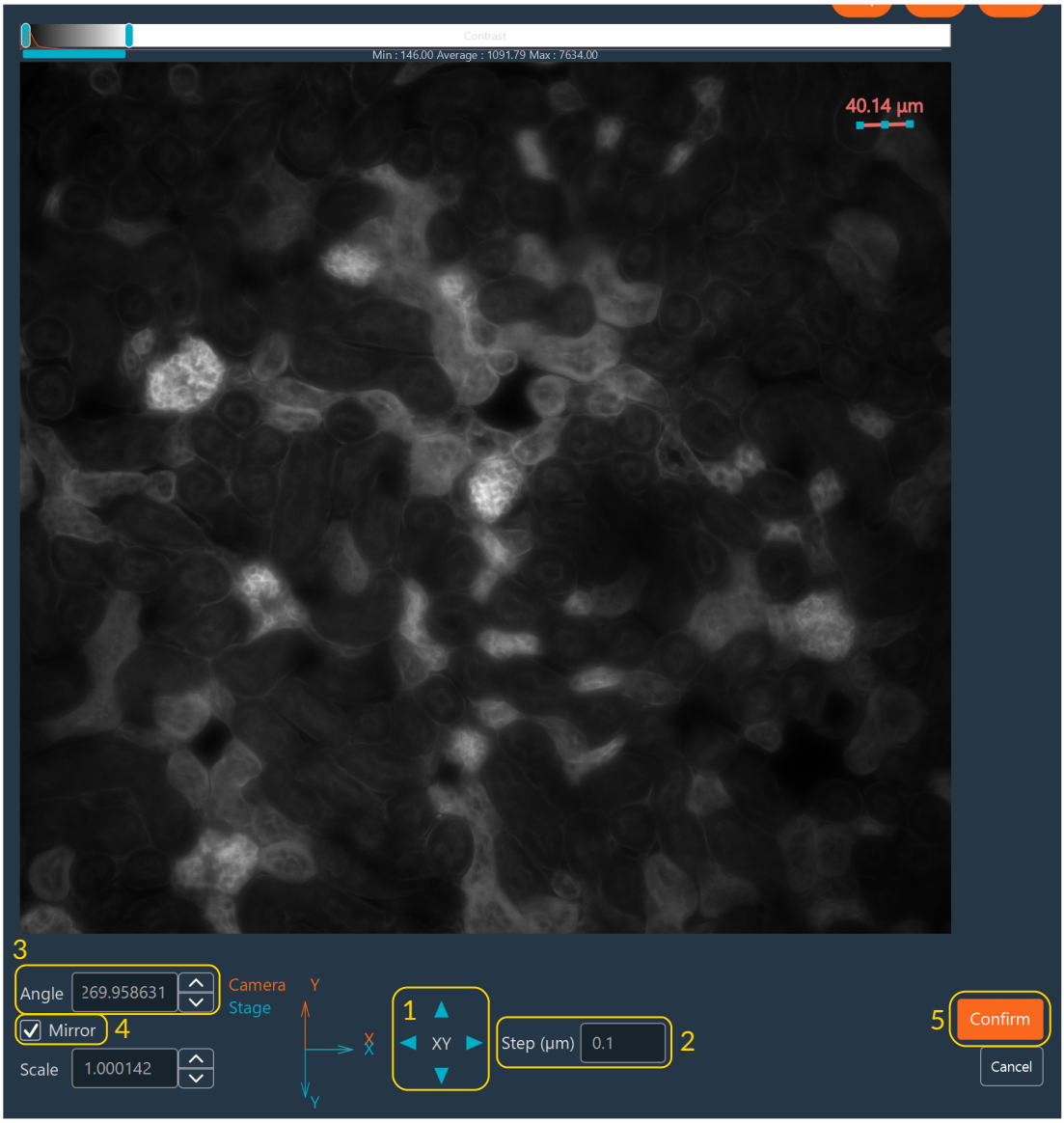


1. Click **Tiling** in the Calibration section. The indicator is orange if uncalibrated, and turns green when calibrated. Then **Select** a calibration mode.

> **Streaming Requirement**
>
> Stream camera data in **Live** mode and bring a sample into focus before initiating either calibration.

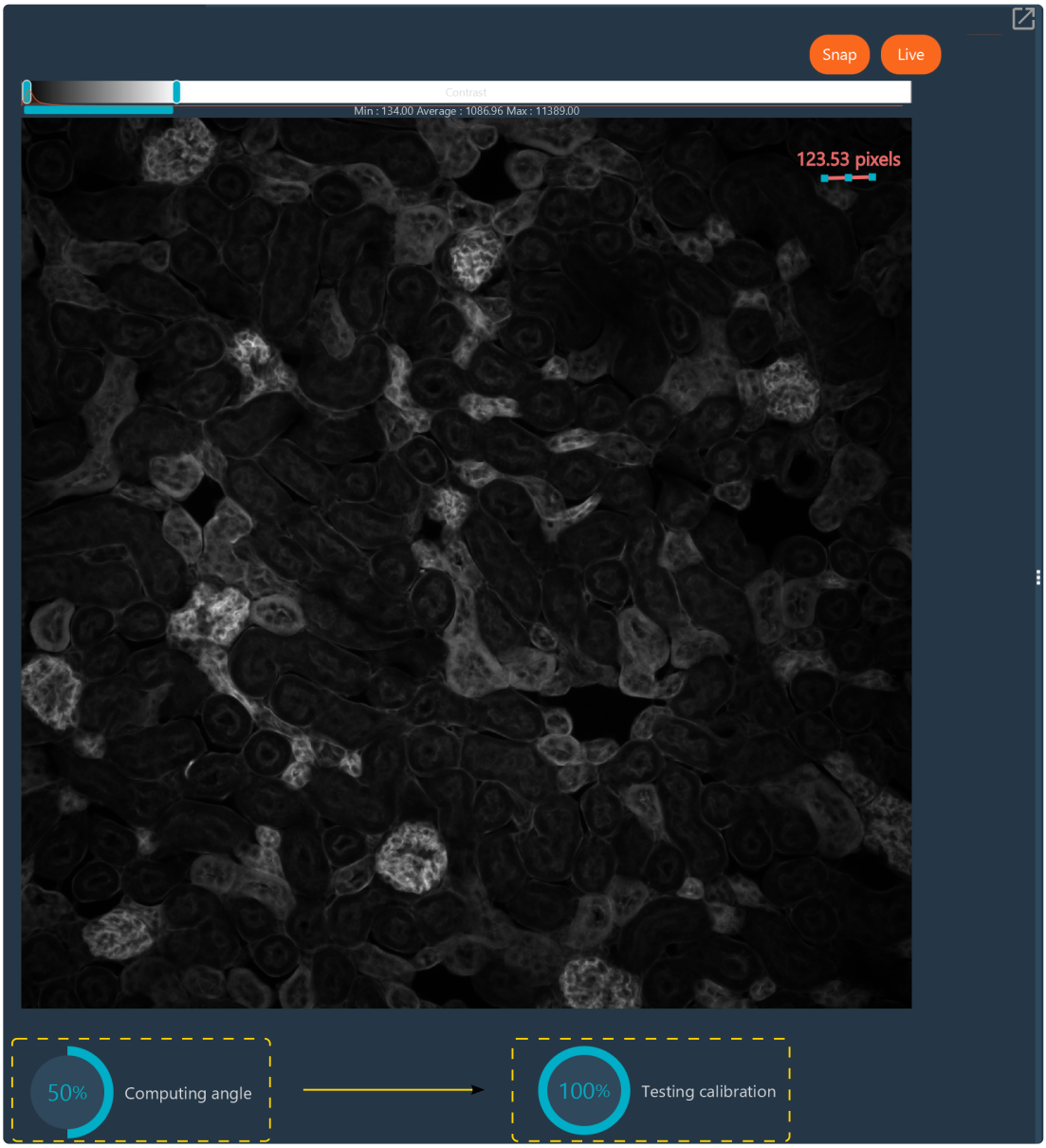
2. **Manual Calibration Protocol**:
    1. Move the XY stage using the virtual joystick. Confirm that the stage movement matches the image's orientation on the screen.
    2. Adjust the movement step size for finer control.
    3. Apply rotation to the live camera feed until it aligns with the stage movement axes.
    4. Apply mirroring to the live camera feed until it aligns with the stage movement axes.
    5. Click **Confirm** to save the calibration.

All rights reserved.



3. **Automatic Calibration Protocol**:

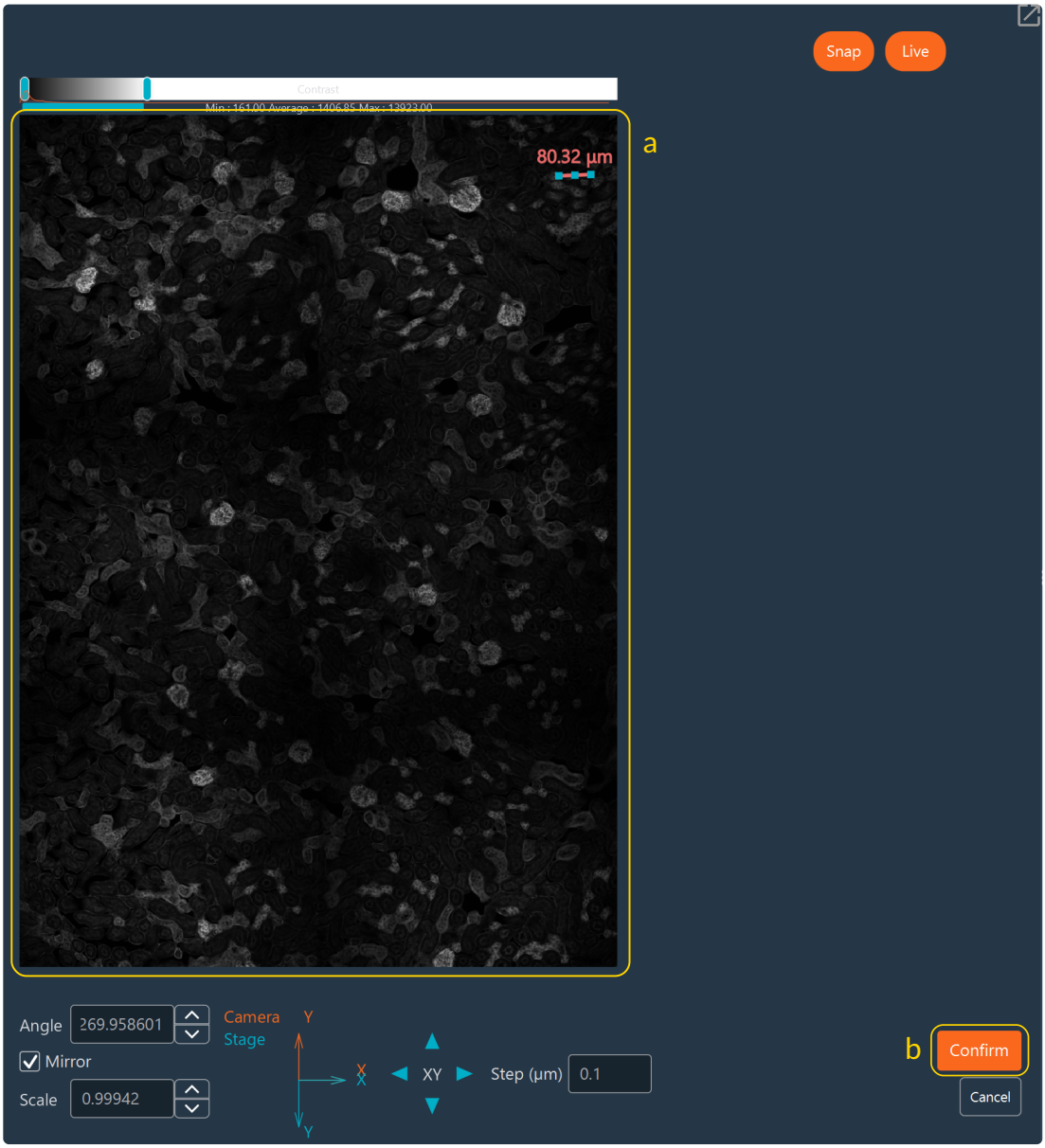
All rights reserved.



The automatic tiling calibration runs independently. Track its progress using the **circular indicator**, which fills completely with blue upon completion.

Review the resulting tiling image **(a)** to verify the calibration, then click **Confirm (b)** to validate.

All rights reserved.

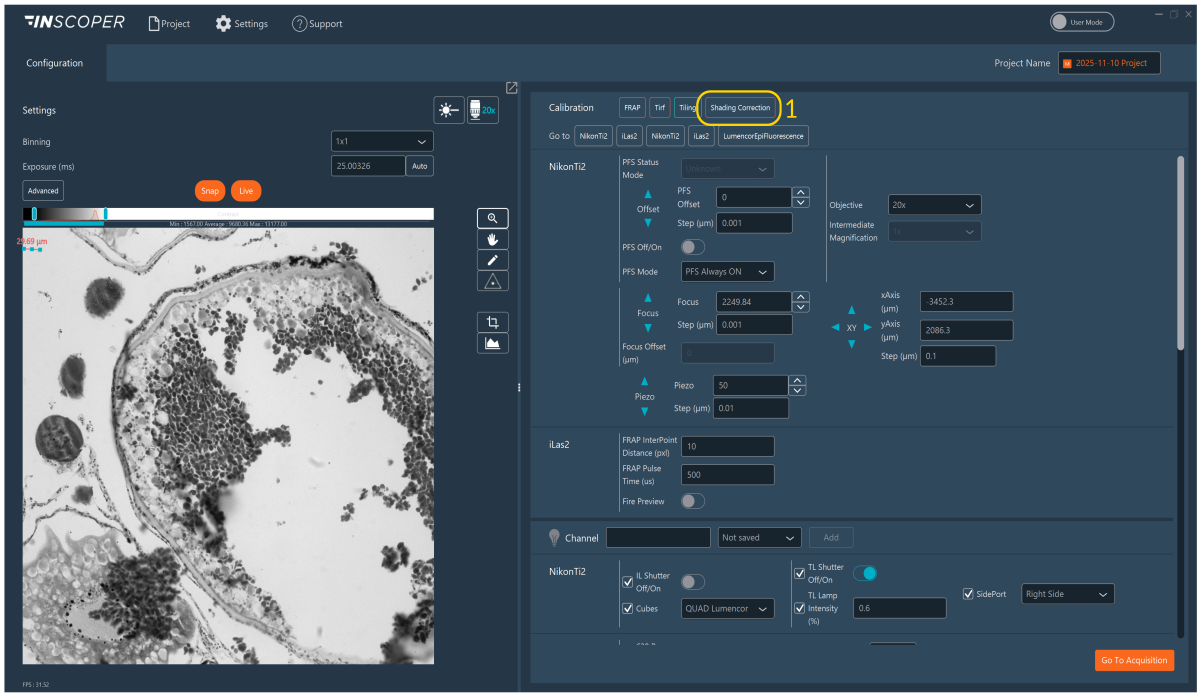


Shading Correction Calibration

Shading Correction (flat-field correction) ensures uniform illumination intensity across the camera field of view, compensating for light source vignetting or unevenness. Perform this calibration using a uniform reference sample, such as an auto-fluorescent slide (e.g., Chroma).

All rights reserved.

EXECUTING THE SHADING CALIBRATION SEQUENCE



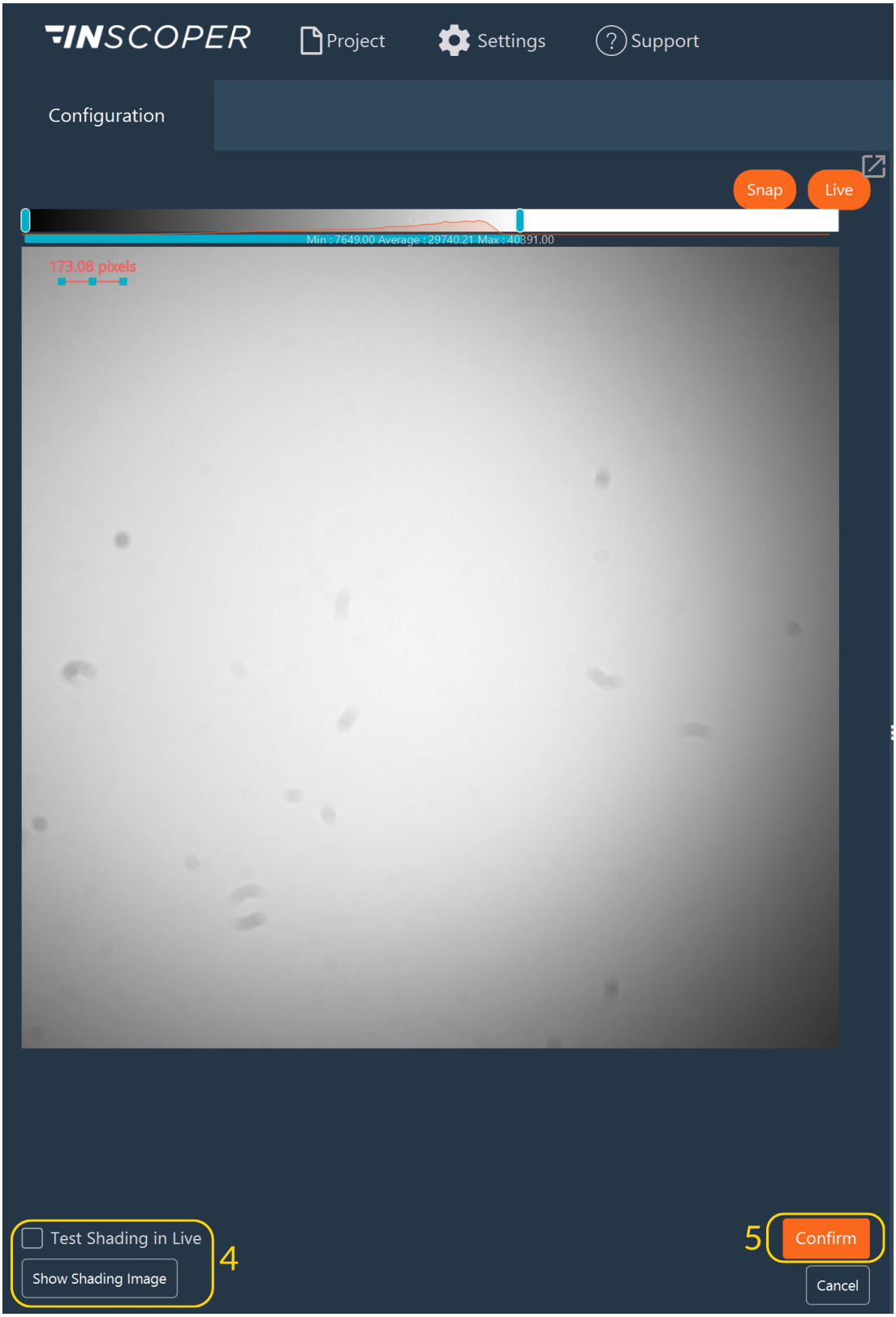
1. Click **Shading Correction**.

All rights reserved.



2. Specify the number of acquisitions. The software averages the captured frames to reduce noise.
3. Click **Confirm** to start the calibration routine.

All rights reserved.



4. Once the calibration is complete, validate it:
    - Verify the result by clicking **Test Shading** in **Live** mode.
    - View the generated shading image map by clicking **Show Shading Image**.
5. Click **Confirm** to save the calibration.

Once calibrated, the **Shading Correction** indicator turns **green**.

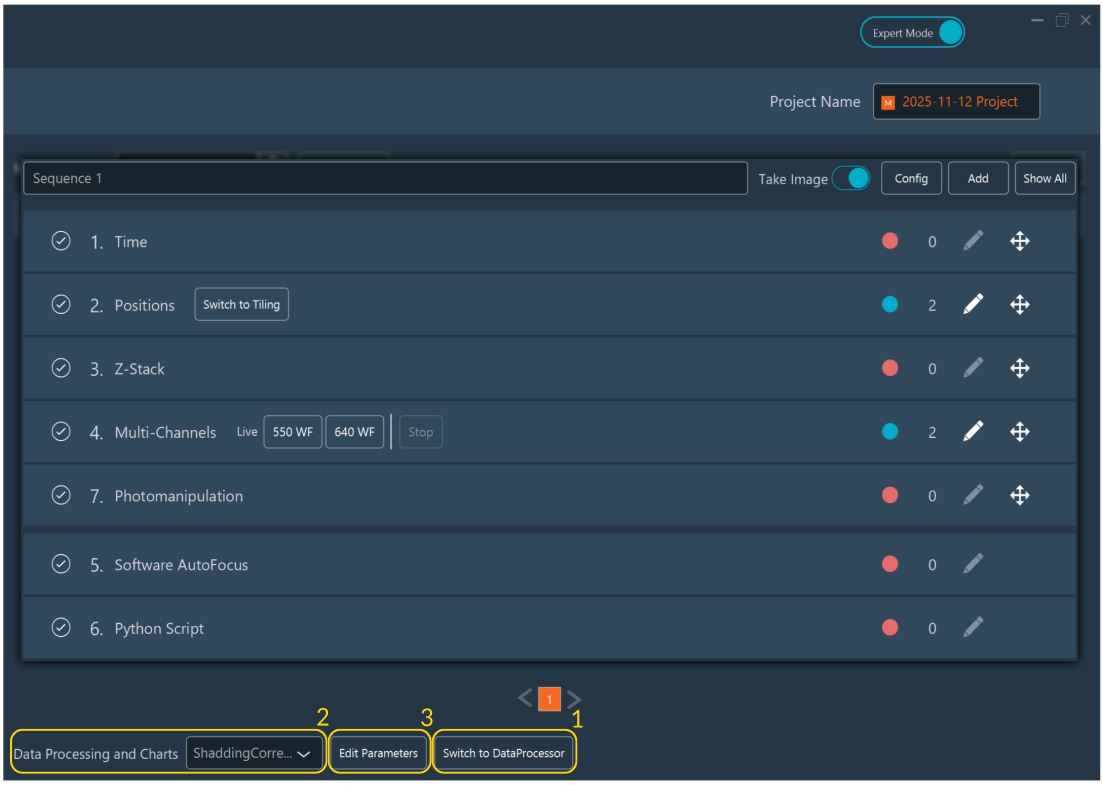
APPLYING SHADING CORRECTION VIA THE DATA PROCESSOR

Once calibrated, shading correction is not applied automatically; it must be explicitly invoked within the data processing pipeline.

1. In **Expert Mode**, construct a data processing pipeline containing the **Raw Data**, Shading Correction, and **Tiling** nodes.
2. Assign this pipeline to your sequence during the Acquisition setup.

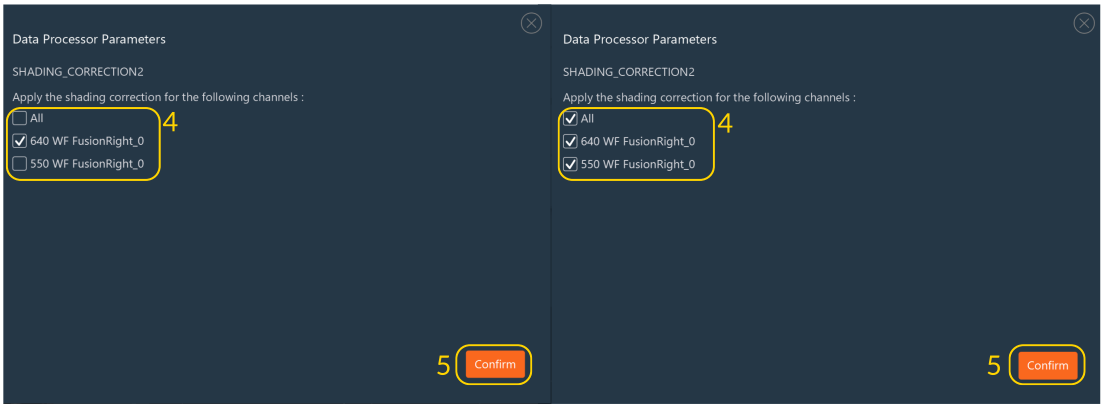
All rights reserved.

3. Click **Edit Parameters** on the **Shading** node.



4. Select the channel(s) that require shading correction.

5. Click **Confirm** to validate your selection.



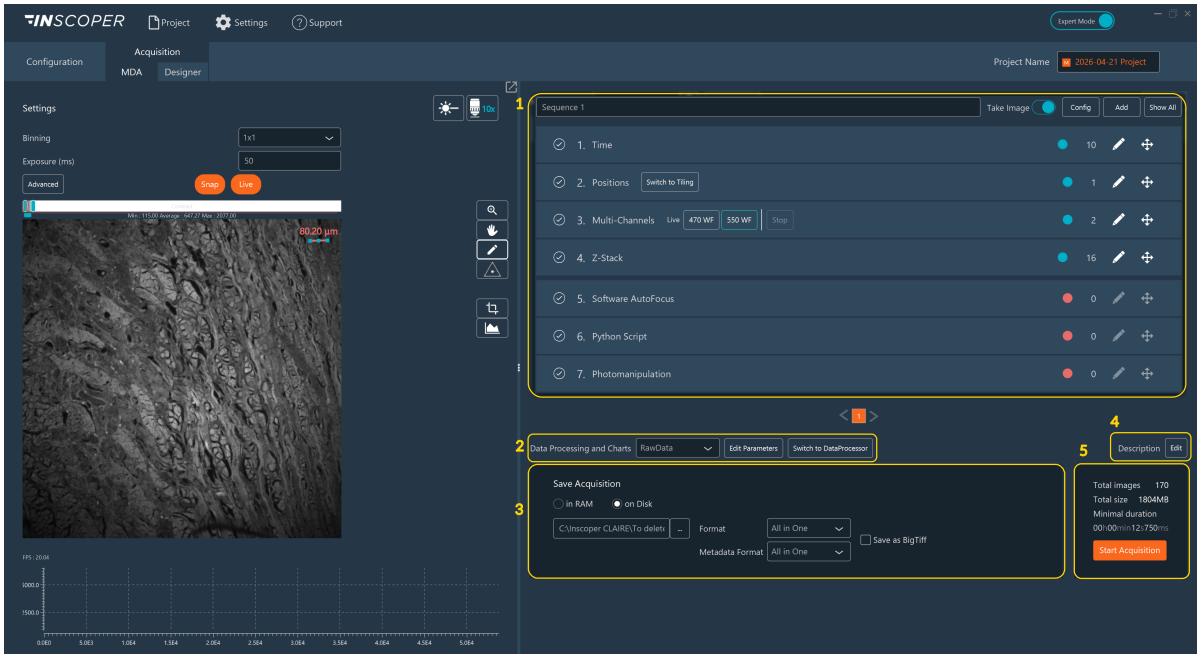
2026-09-08

All rights reserved.

1.4 Acquisition

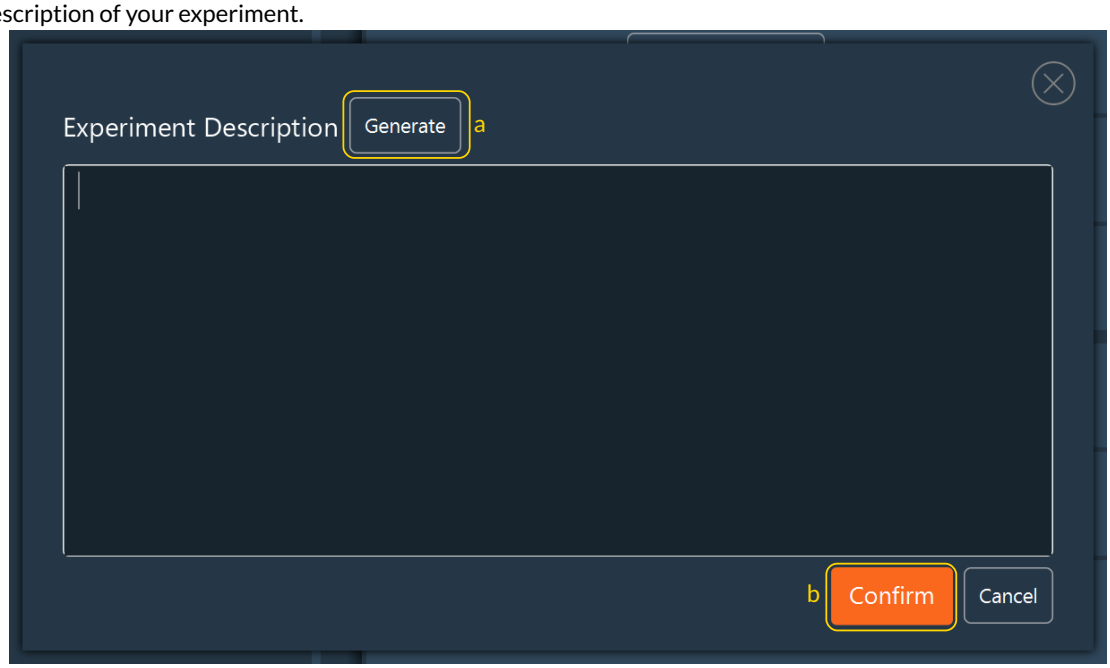
1.4.1 Acquisition

Use this tab to configure complete multi-dimensional acquisition sequences across all available dimension modules.



To set up a **multi-dimensional acquisition**:

1. Select the required dimension(s) from the panel.
2. Choose the appropriate data processor.
3. Specify the save directory for the output files.
4. Edit a description of your experiment.



- a. Click **Generate** to add acquisition sequence description and/or Write your own experiment description.
- b. Confirm to close the window.

5. Launch the acquisition.

All rights reserved.

> **Note**
>
> The interface varies depending on the microscope system hardware, particularly the available dimensions.

All dimensions share a consistent interface. The UI controls for configuring dimensions include:

Icon	Description
	**Inactive**: The sequence excludes the dimension. Click to activate.
	**Active**: The sequence includes the dimension. Click to deactivate.
	**Configured**: The dimension's parameters are set. Here, 11 images will be acquired for this dimension.
	**Edit**: Click to configure the dimension settings.
	**Reorder**: Drag and drop to change the dimension execution order. The dimension order defines the nesting loop of the sequence, executing from top to bottom.

Dimension Execution Order

The order of the active dimensions dictates the nested loop structure of the acquisition sequence. Use the **Reorder** handle to drag and drop dimensions to change their execution order.

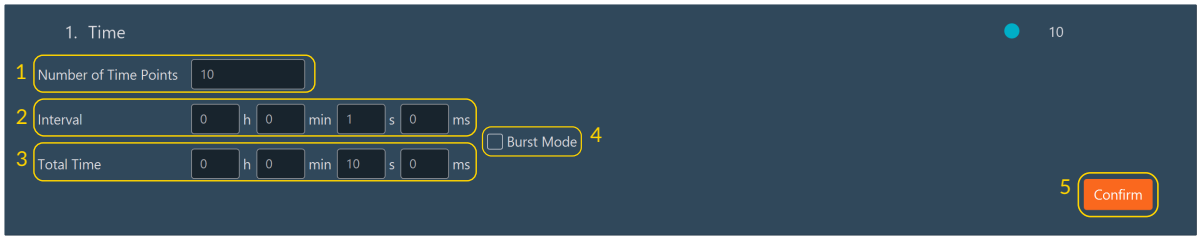
Dimensions at the top of the list act as the outer loops of the sequence, progressing more slowly. Dimensions positioned lower in the list act as the inner loops, executing more frequently. For example, if **Time** is placed above **Channels**, the system acquires all channels for a single time point before progressing to the next time point. Conversely, if **Channels** is placed above **Time**, the system acquires the entire time-lapse sequence for the first channel before switching to the next channel. Permuting these dimensions correctly is critical, as it directly determines how the hardware coordinates the image acquisition.

2026-09-08

All rights reserved.

1.4.2 Time

Use the **Time** dimension to configure time-lapse acquisitions to observe dynamic biological processes.



Configuring a time-lapse

1. Specify the **total number of time points** (frames) required in the acquisition.
2. Define the **interval duration** between consecutive time points (minimum time to wait between frames).

> **Note**
>
> The software adapts the **Total Time** automatically. You may also set the total time manually and infer the interval duration for a fixed number of time points.

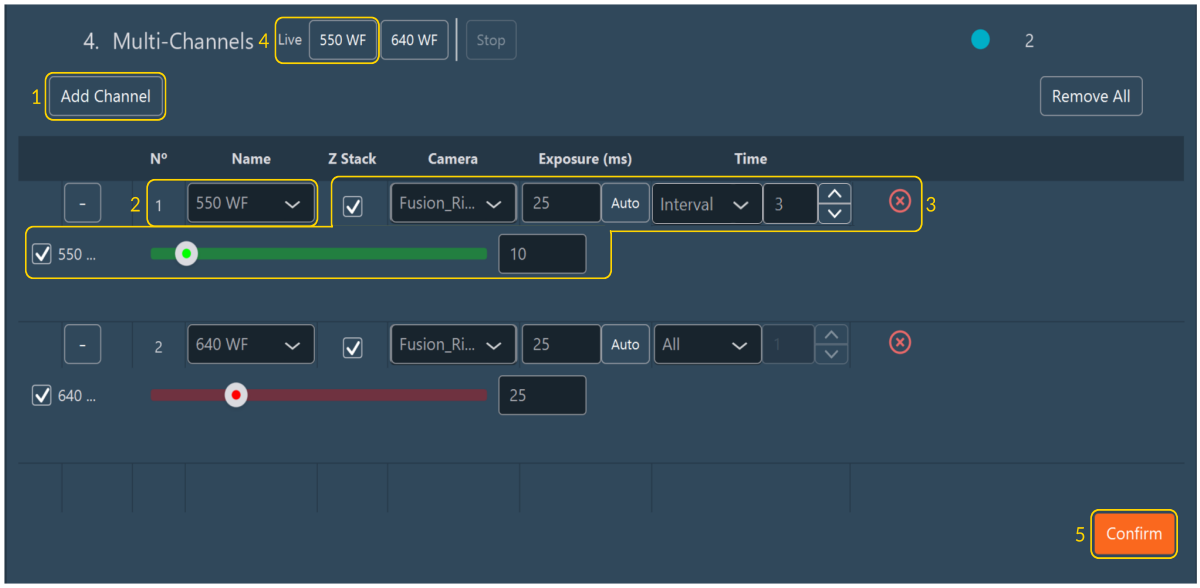
1. Enable **Burst Mode** to acquire images continuously at the maximum hardware frame rate, ignoring the specified interval. Use this to capture rapid kinetics.
2. Click **Confirm** to save the time dimension settings.

2026-09-08

All rights reserved.

1.4.3 Multi-Channels

Use the **Multi-Channels** dimension to configure one or more channels (e.g., fluorescence, brightfield) for the acquisition sequence. Channels must have been defined previously in the Configuration Tab.



CONFIGURING CHANNELS

1. Click **Add Channel**.
2. Select a channel from the list.
3. Adjust the following parameters for each channel independently:
    - **Z-stack**: Enables the Z-stack dimension specifically for this channel. If you uncheck this box, the center plane will be acquired.
    - **Z-Offset**: Applies a spatial Z-offset. This is used when focal planes vary between emission wavelengths (chromatic aberration) or when targets are at different depths.
    - **Shutter Blink**: Optimizes hardware shutter actuation to minimize sample exposure time and prevent photobleaching.
    - **Camera**: Assigns the active camera (for multi-camera systems).
    - **Exposure (ms)**: Defines the sensor exposure time for the channel.
    - **Time**: Defines the acquisition frequency within a time-lapse (e.g., acquire this channel at all time points, only the first time point, only the last time point, or at intervals).
    - **Intensity**: Adjusts the excitation light source intensity (hardware-dependent, usually % power).
    - **Red cross**: Delete a channel
4. To start a real-time preview of a channel, click the button corresponding to the channel's name (located next to the **Live** label in the multi-channel header). To apply any subsequent configuration changes, click the button again.
5. Click **Confirm** to apply the configuration.

> **Note**
>
> If you change the exposure time or intensity during a live preview, click **Live channel** again to refresh the hardware and update the image stream.

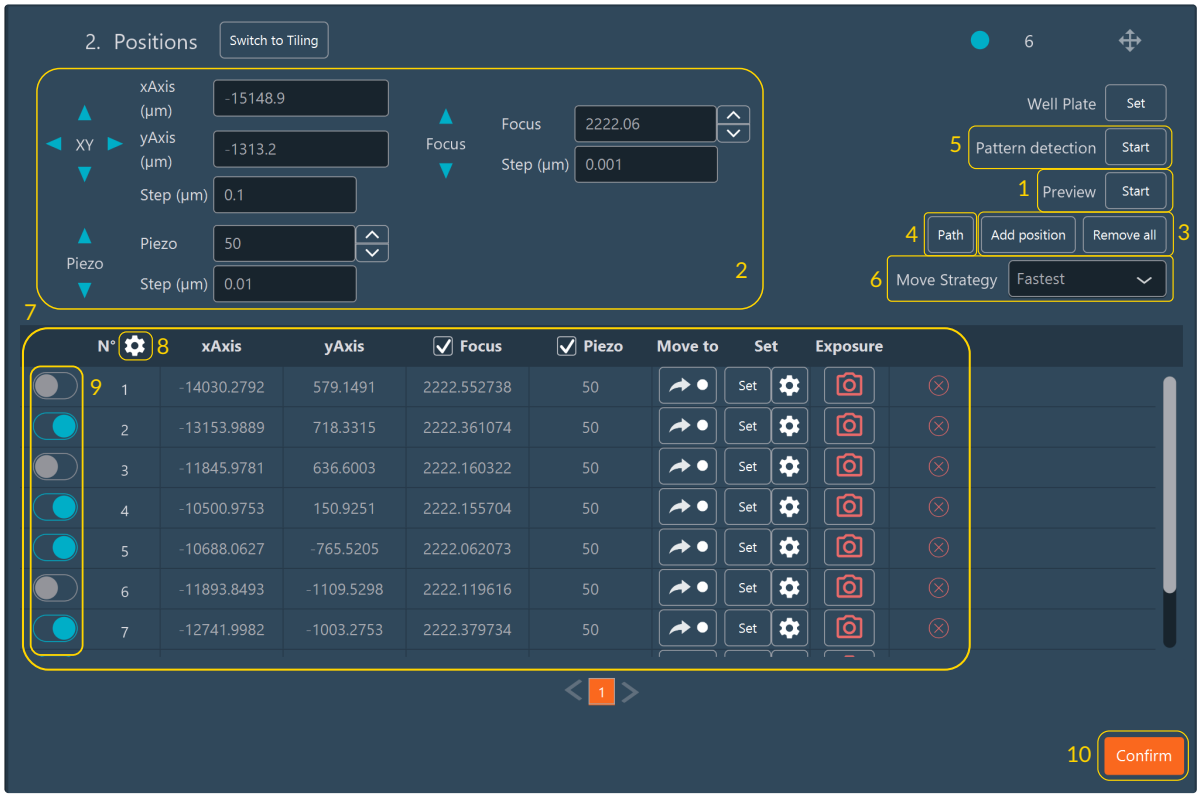
> **Tip**
>
> Reorder the execution sequence of the channels by dragging and dropping them in the list.

2026-09-08

All rights reserved.

1.4.4 Positions

The **Positions** dimension captures multiple discrete XYZ locations across a sample.

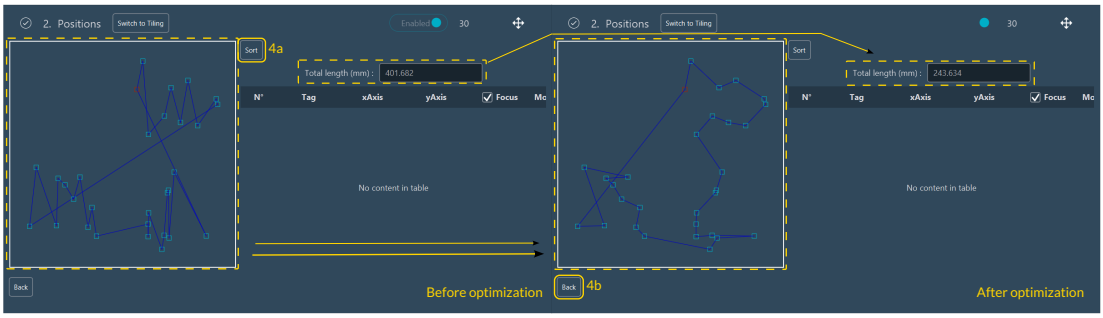


Switch to Tiling

Toggle directly to the **Tiling** dimension using the **Switch to Tiling** button to capture a continuous mosaic rather than discrete points.

Adding and Optimizing Positions

1. Generate a fast Preview of the sample to act as a navigation map.
2. Move the **stage** to the target area and adjust the **focus**.
3. Click **Add position** to record these XYZ coordinates in the position list.
4. To optimize the **stage movement** trajectory across multiple positions, click the **Path** button.

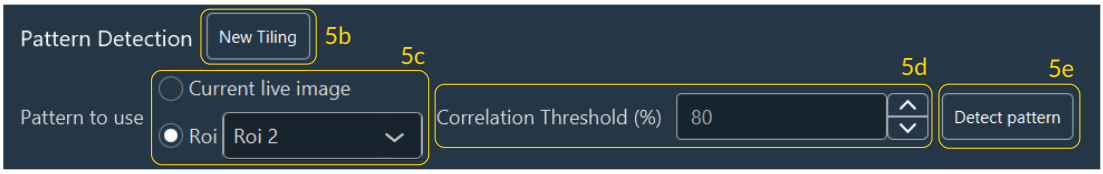


{ .center width=100% }

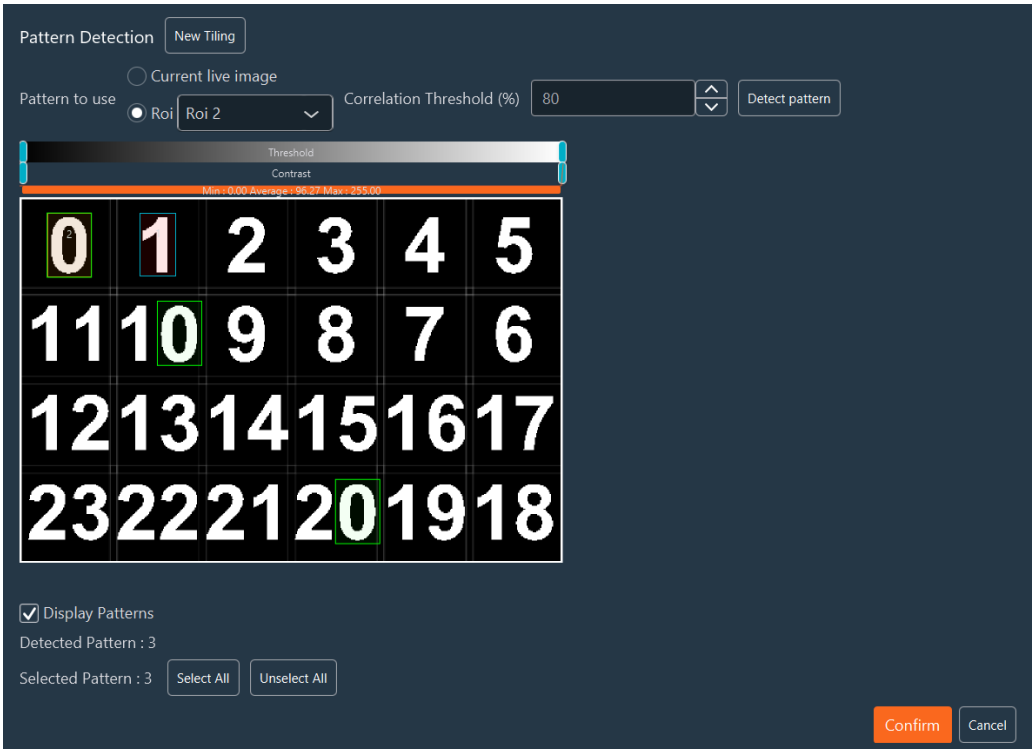
- **a**. By default, the stage visits positions in the order they were added. The interface visualizes this path and calculates the total travel distance. Click **Sort** to compute an optimized route (e.g., shortest path) that minimizes travel time.
- **b**. Once optimization is complete, click the **Back** button to return to the principal position view.

5. Use the **Pattern Detection** feature to automatically find XYZ coordinates matching a visual pattern in the sample.

All rights reserved.

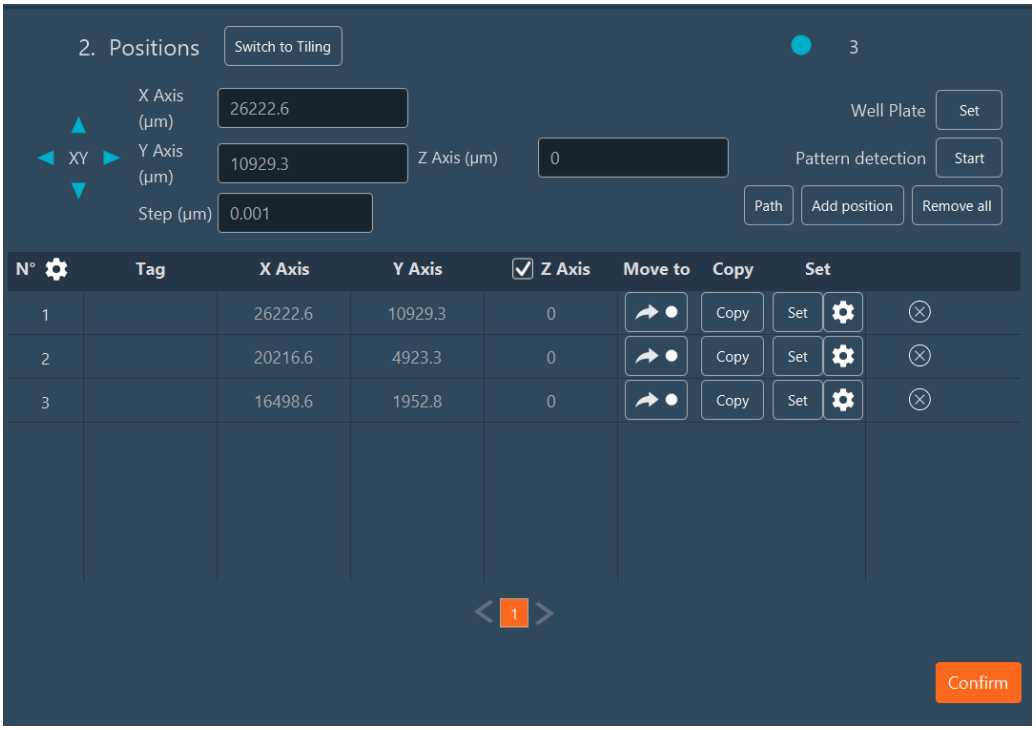


- **a**. Click the **Start** button located in the Pattern Detection section.
- **b**. Create a reference scanning area by clicking **New tiling**. Configure it as described in the Tiling dimension section.
- **c**. Define the source pattern to detect:
  - **Current Live Image**: Extracts the pattern from the current live FOV.
  - **ROI**: Draw an ROI around the specific biological pattern to detect.
- **d**. Set the **Correlation Threshold** (this defines the minimum similarity percentage to identify a match).
- **e**. Click **Detect pattern** to initiate the scanning algorithm.
- **f**. Recognized patterns are highlighted with bounding ROIs over the preview. Manage these selections using **Select All**, **Unselect All**, or by clicking individual ROIs to toggle them.

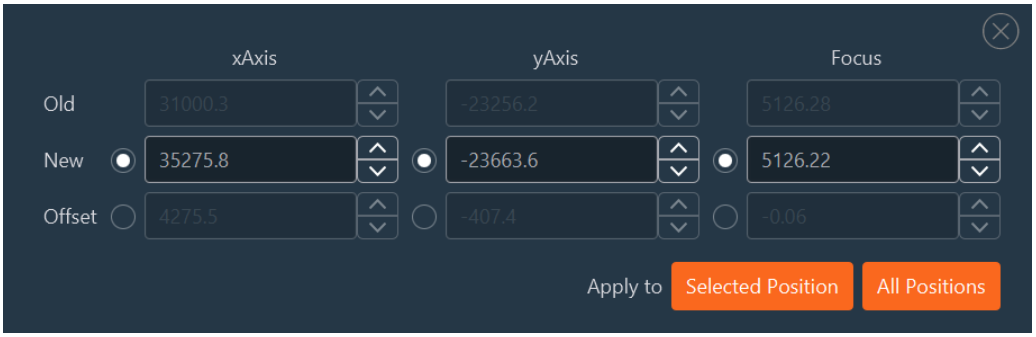


- **g**. Click the **Confirm** button. The geometric coordinates of each detected pattern are appended automatically to the position list.

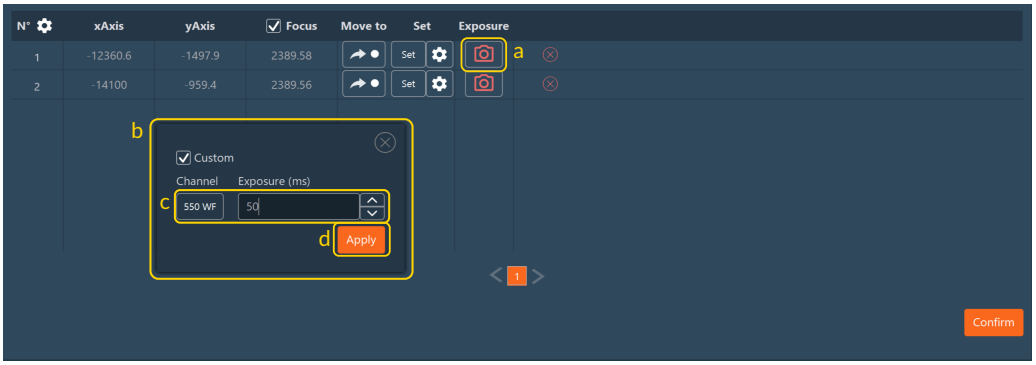
All rights reserved.



6. Configure the **Move Strategy**: For multiposition acquisitions spanning large travel distances, select how the stage translates:
   - **Fastest**: Navigates directly from point A to point B at maximum speed.
   - **XY Split**: Translates between positions incrementally. Specify the movement interval in the **Step (µm)** field.
7. **Position Editing & Overrides**:
   - **X-Axis/Y-Axis**: Manually modify coordinates.
   - **Focus / Z-axis**: Check this box to force the system to use the Z-value saved for this position. If unchecked, the microscope maintains its current live Z-value upon arrival.
   - **Move to**: Immediately drives the stage to this coordinate.
   - **Set**: Overwrites the saved coordinates for this position with the current live XYZ stage position. Clicking the  modifies specific axes (X, Y, or Z) and applies global offsets.



   - **Custom Exposure per Position**: Assign unique camera exposure times to specific spatial positions (useful if sample brightness varies drastically across a tissue section).

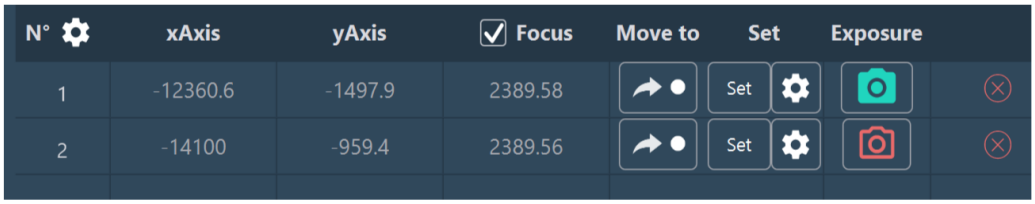


1. Click the **camera icon** on the row corresponding to the position where you want to adjust the exposure time.

All rights reserved.

2. Check the **Custom** box to unlock exposure time modification.
3. Modify the exposure value in milliseconds. This override applies only to the currently active channel.
4. Click **Apply** to confirm the changes.

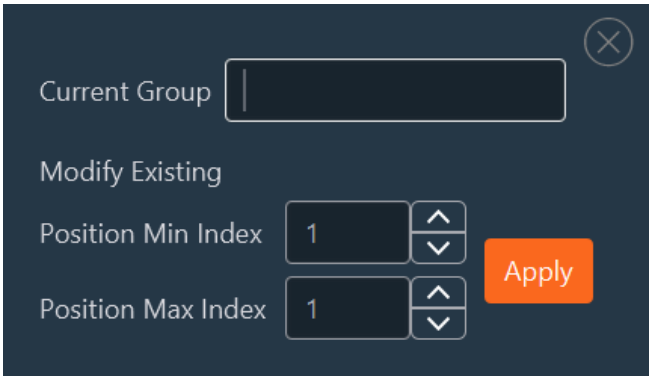
When a row is modified, the **camera icon color** changes from **red to green**, serving as a visual indicator that a custom exposure time is active.



- 

: Permanently deletes this position from the list.

8. **Grouping**: To manage large numbers of positions, cluster them into logical groups. Click the **gear icon** to access group settings:
   - Assign a semantic name to the group.
   - Indicate the contiguous range of position indices to group together (e.g., 1-15).
   - Click **Apply**.
   - Repeat these steps to create multiple, non-overlapping groups.



9. **Activate** or **Disable** the position. This allows you to store a specific position without actively acquiring it.
10. When all position geometries and parameters are finalized, click **Confirm**.

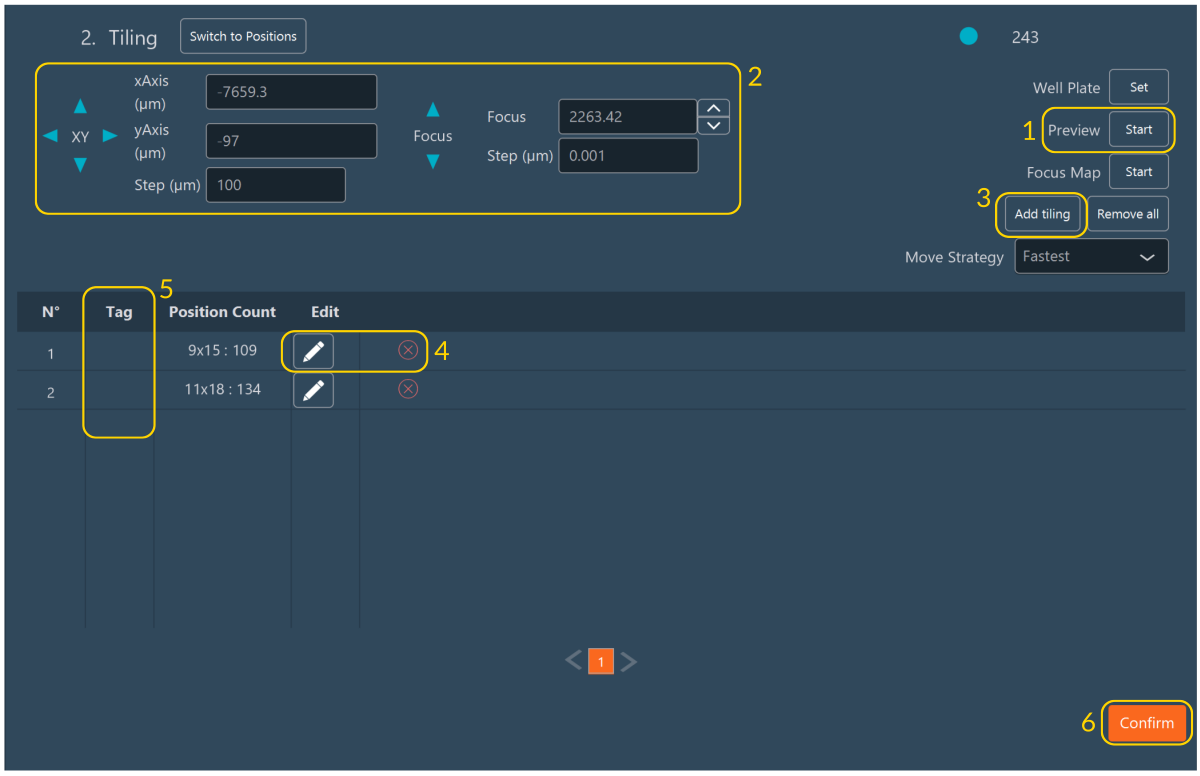
2026-09-08

All rights reserved.

1.4.5 Tiling

Creating a Tiling

Use the **Tiling** dimension to capture large sample areas through automated mosaic acquisition.



Switch to Positions

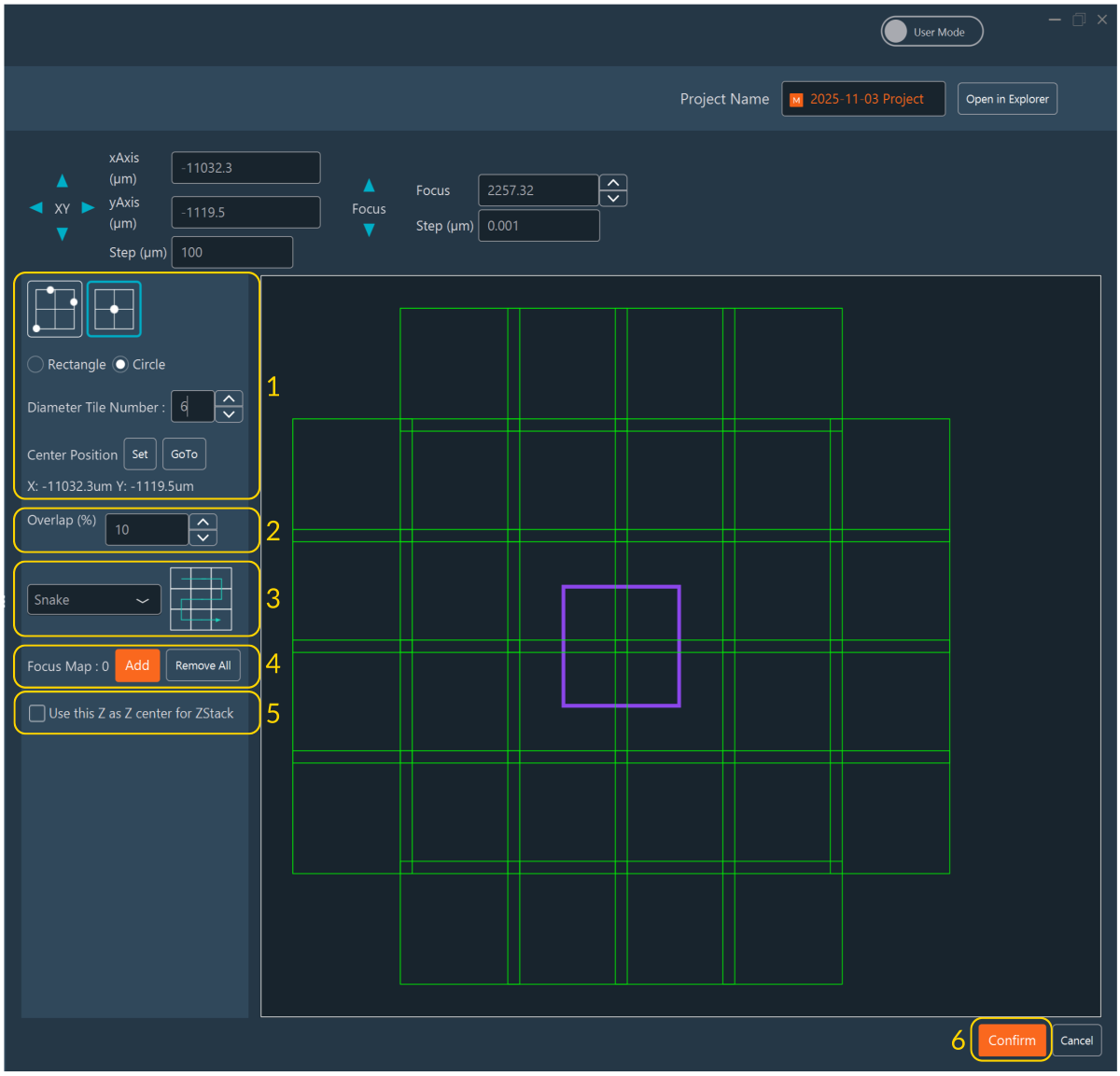
You can toggle directly to the Positions dimension using the **Switch to Positions** button if you require discrete capture points instead of a continuous mosaic.

1. Generate a low-resolution Preview of the sample.
2. Move the stage to the region of interest and focus the image.
3. Click **Add tiling** to append a new tiling configuration block to the dimension list.
4. To configure the parameters of a tiling block, click the **Pen** icon. To delete a block, click the **Red Cross**.
5. Assign a label to each tiling block by modifying its field in the **Tag** column.
6. Click **Confirm** to save the tiling configuration.

Setting Up a Tiling Layout

Define the physical tiling boundary using two topological methods:

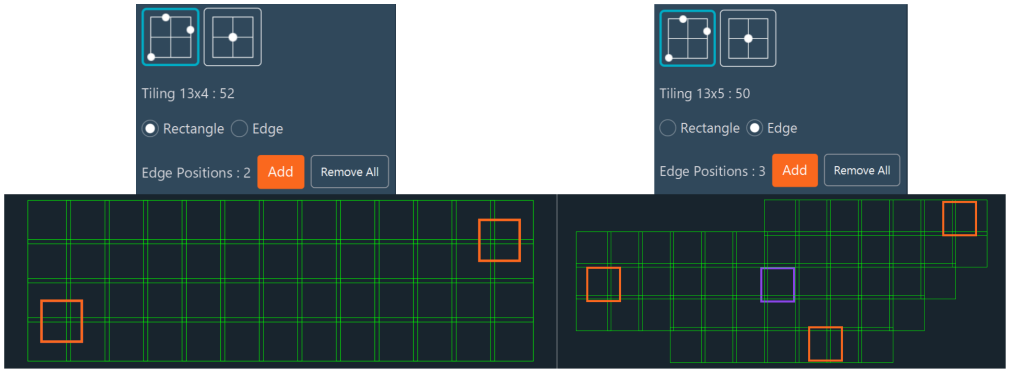
All rights reserved.



1. **Define Boundaries:**

    1. **By Edge**: Click **Add** to drop coordinates outlining the region of interest. The software connects these points to form a polygon.

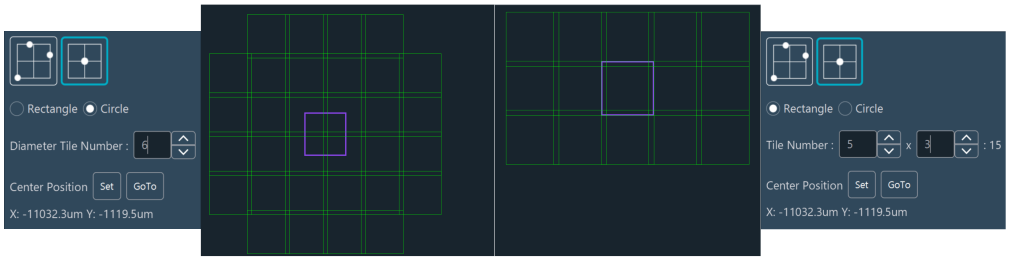
        - Clear the bounding box using **Remove All**, or delete a single vertex by right-clicking it.



    2. **By Center**: Drive the stage to the center of the acquisition region and click **Set**. Then, select either a rectangular or circular tiling geometry.

        - For a rectangle, specify the bounding grid dimensions (e.g., 3x3 tiles). For a circle, specify the bounding diameter.

All rights reserved.

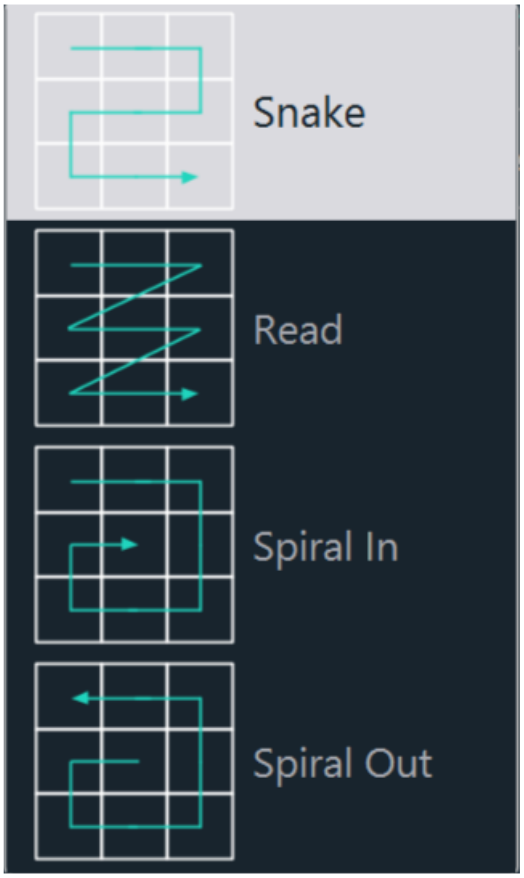


> **Note**
>
> Use the right-hand panel controls to manually jog the hardware stage and manage your defined coordinate vertices.

2. **Define Tile Overlap**: Set the overlap percentage between adjacent frames. A 10% overlap is standard for downstream stitching.
    - Enable **Auto Mode** to let the software calculate the overlap ratio based on your specified coordinate boundaries.
    - To manually dictate the translation distance between individual tiles, enable the **Custom Step** option via the Display Settings menu.

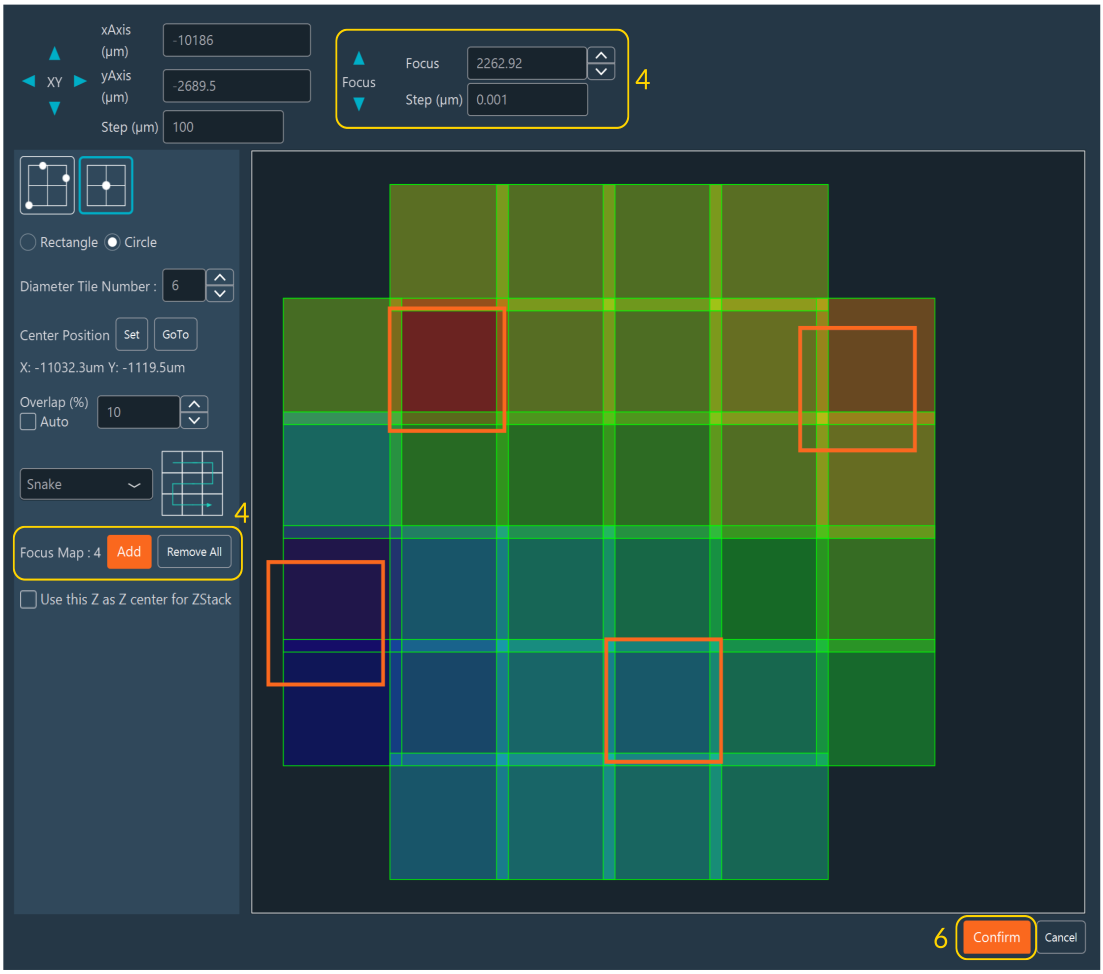
> **Important**
>
> Accurate overlap is necessary for successful post-acquisition image stitching.

3. **Select Read Mode**: Choose the scanning path (e.g., Spiral, Snake) the stage executes during acquisition.



4. **Generate a Focus Map**: To compensate for uneven sample topology across a tiling area, create a Focus Map. Navigate the XY plane using the interface joystick or by clicking within the tiling map. Adjust the Z-focus at various locations, and click **Add** to store each point.
    - Add points as necessary to model the sample's topography. Right-click any point (orange indicator) to remove it.
    - The software applies a color gradient overlay to the map, representing the interpolated Z-surface altitude across the acquisition region.

All rights reserved.



5. **Configure Z-Stack Integration**:
    - Select the corresponding checkbox inside the tiling editor to center Z-Stacks around the interpolated Z-values stored in the Focus Map.
    - If unchecked, the sequence uses either global Z-Stack parameters or the current live Z-position.
6. Click **Confirm** to save the spatial tiling configuration.

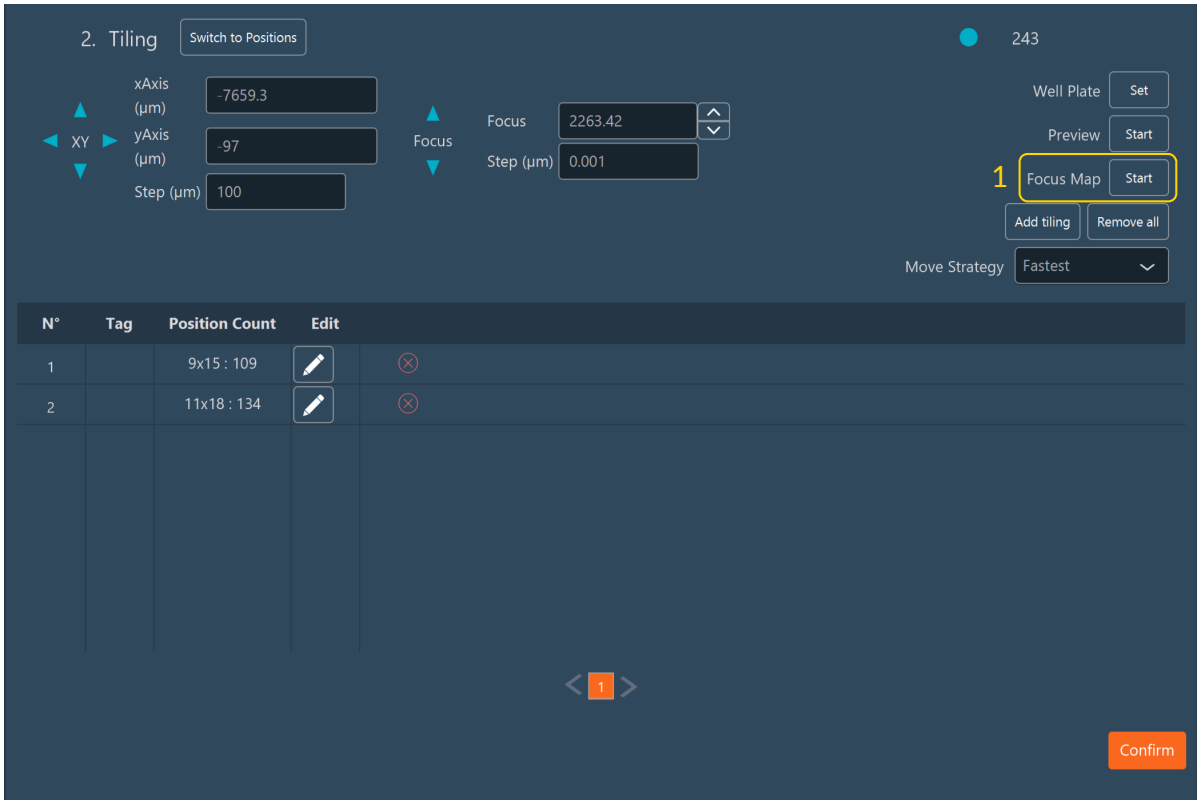
> **Note**
>
> - **Orange squares**: Indicate saved boundary vertices or focus map reference anchors.
> - **Purple square**: Indicates the real-time physical position of the XY stage.

> **Tip**
>
> Use the **GoTo** button to rapidly drive the mechanical stage back to a previously saved topological reference position.

Focus map automated setup for simple and multiple tilings

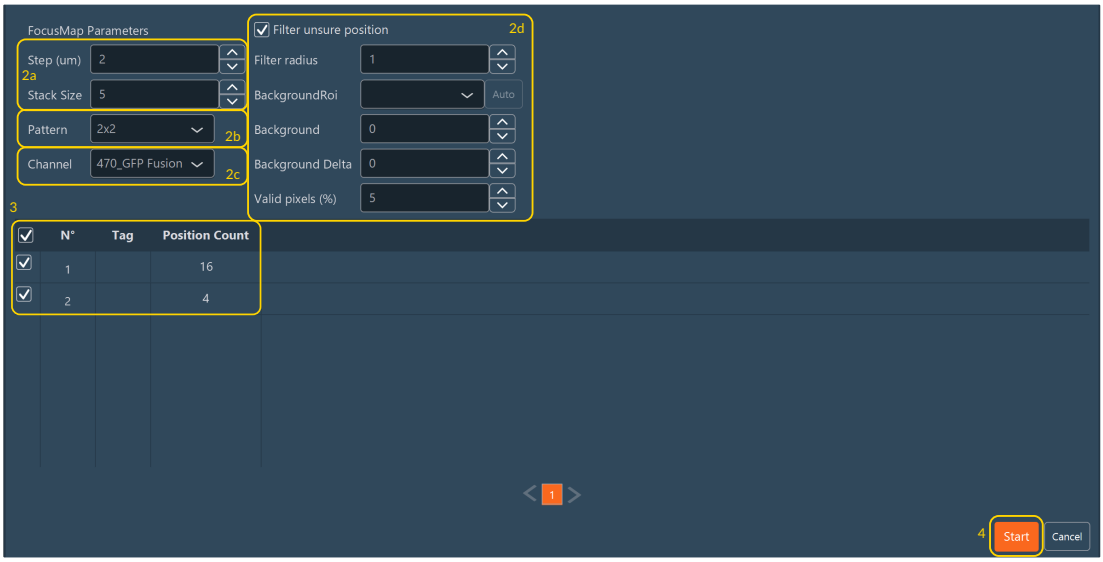
The Automated Focus Map is an optional software feature of the Inscoper I.S. that automatically generates a relief map of the biological sample when performing tiling imaging.

All rights reserved.

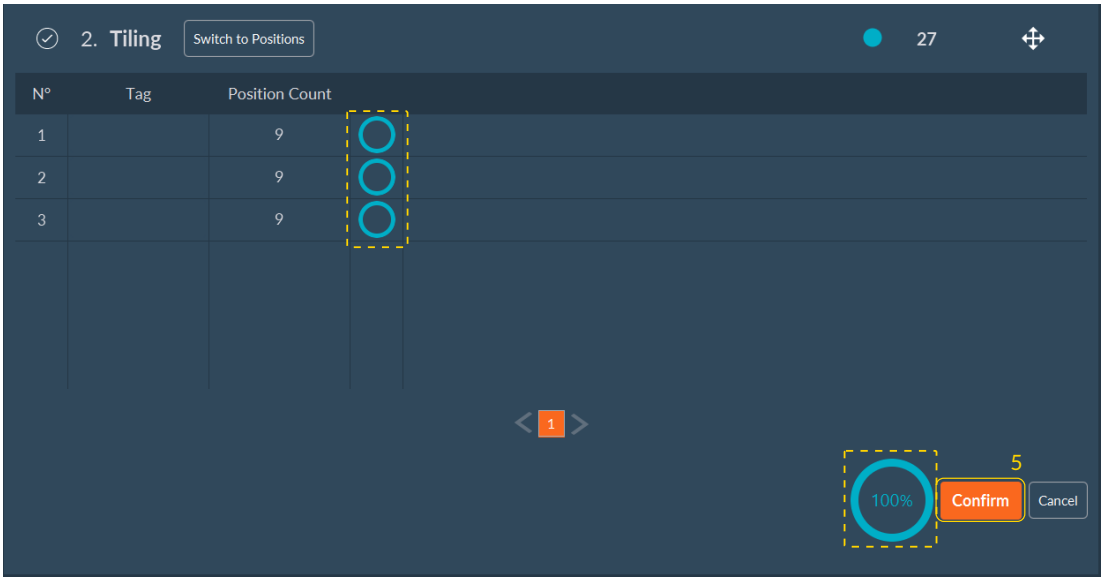


1. Click **Focus Map Start** within the **Tiling** dimension to begin configuration.
2. Set the following parameters for both simple and multiple tilings:
    - a. Select the step and size of the Z-stack.
    - b. Select an existing pattern.
    - c. Select the channel used to acquire the stack.
    - d. (Optional) Select **Filter unsure position** and configure the corresponding parameters:
        - i. **Filter Radius**: Smooth noise by replacing each pixel with the median of its neighbors (e.g., 0 = no smoothing, 1 = median of a 3x3 square surrounding the pixel, 2 = median of a 5x5 square surrounding the pixel).
        - ii. **Background ROI**: Select the Region of Interest (ROI) used to automatically calculate background values.
        - iii. **Background**: Define the mean background value.
        - iv. **Background Delta**: Define the acceptable noise amplitude.
        - v. **Valid Pixels**: Define the minimum percentage required to validate a position.
3. Select the tiling region for which to create the Focus Map.
4. Click **Start**. The system scans all points, generates the Focus Map, and displays the determined focal points. A progress bar indicates the current progression per tiling region. Once the Focus Map is complete, the software redirects you to the **Tiling** dimension interface.

All rights reserved.



5. Click **Confirm** to validate and proceed with the acquisition sequence setup.



Live Visualization of a Tiling

To follow the mosaic as it is built during acquisition, configure a Data Processor before starting the sequence.

RECOMMENDED: LIVE PREVIEW, STITCHING AFTER ACQUISITION

1. Open the Data Processor and add a new node to the **RawData** root node.
2. Set the node **Type** to `SIMPLE_TILING`.
3. Start the acquisition. Each frame is placed at its grid coordinates as it arrives, so the mosaic assembles live in the visualization panel. No stitching is applied.

The `SIMPLE_TILING` output is not written to disk. It remains available for review in the **Visualization** tab after the acquisition, listed as `SIMPLE_TILING`.

To obtain the final stitched image, run a separate post-acquisition processing with the `STITCH_TILING` operation and export the result to disk. See Tiling Stitching.

> **Tip**
>
> Stitching after the acquisition allows the stitching parameters to be tested and adjusted on the acquired data, and repeated if the result is unsatisfactory.

ALTERNATIVE: STITCHING DURING ACQUISITION

A serial pipeline can also stitch on the fly:

1. Add a `SIMPLE_TILING` node to **RawData**, then chain a `STITCH_TILING` node to it.

All rights reserved.

2. Before starting the acquisition, click **Edit parameters** next to the Data Processor and set the stitching options.
3. Start the acquisition. Stitching is performed at the end of the sequence.

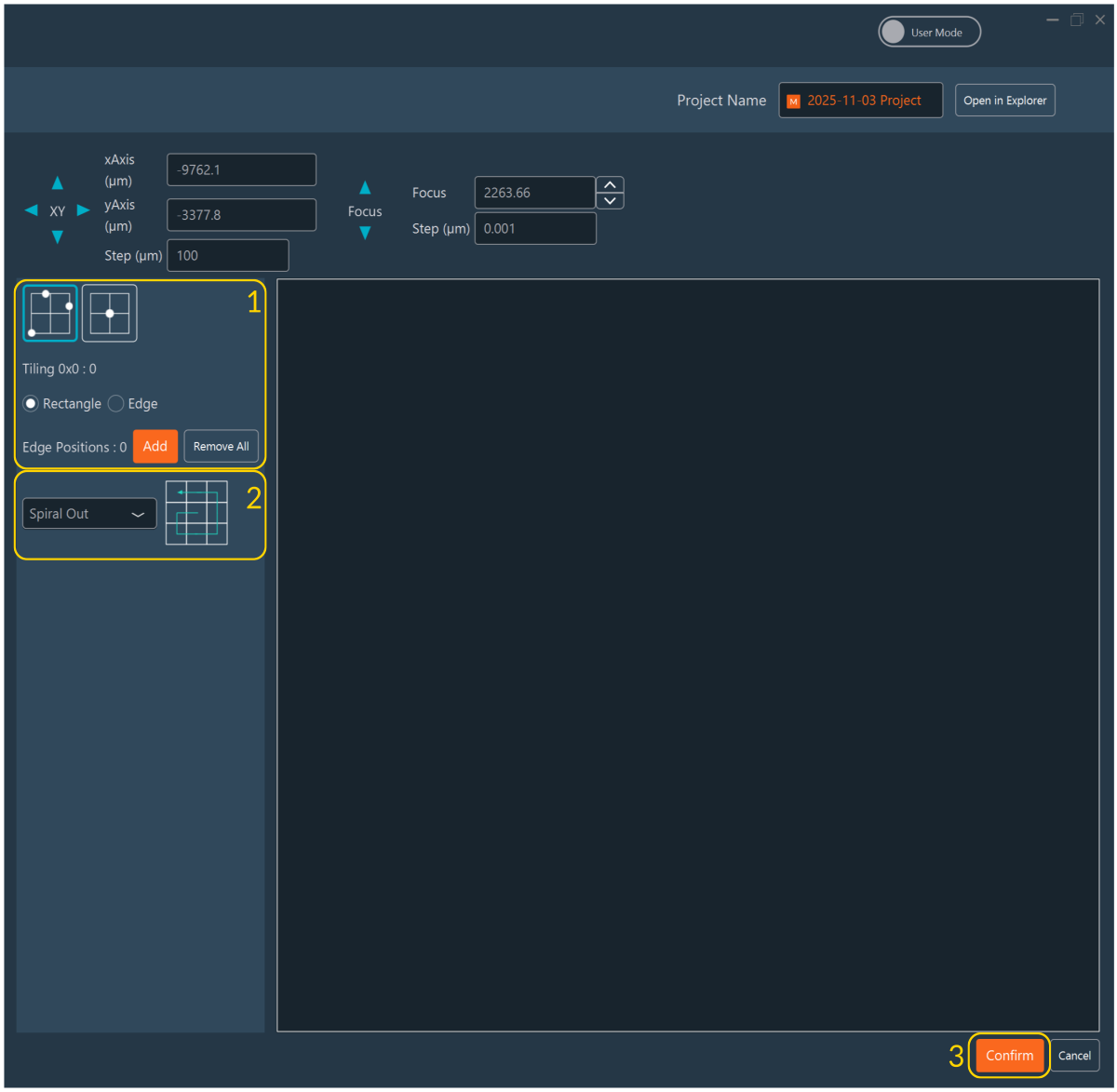
> **Warning**
>
> With this pipeline, the tiling is not displayed live, and the stitching parameters cannot be adjusted once the acquisition has started. Unsuitable parameters produce a failed stitching that can only be corrected by reprocessing the data. Post-acquisition stitching is therefore recommended.

2026-09-08

All rights reserved.

1.4.6 Preview Feature

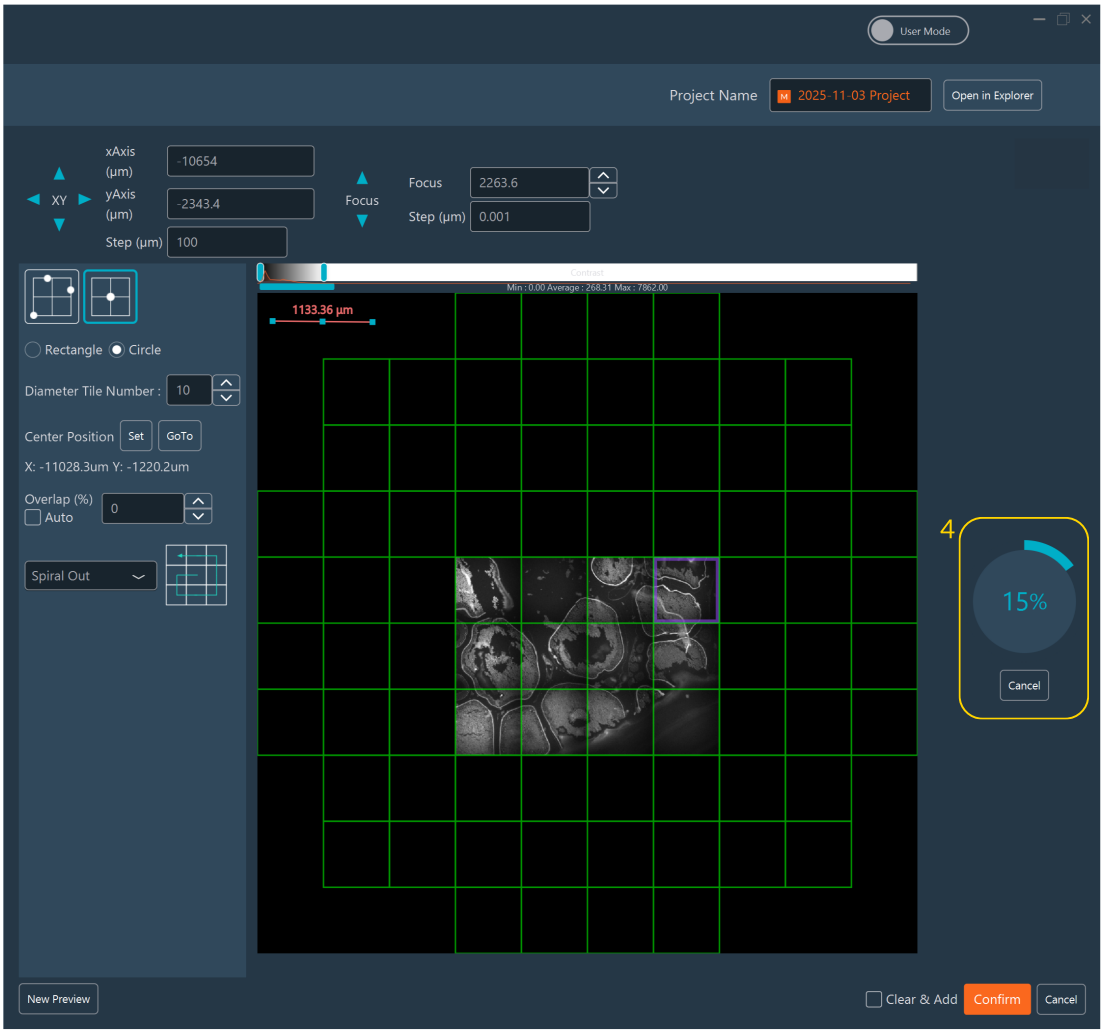
The **Preview Feature** performs a fast, low-resolution scan of the sample to generate an interactive map. Use this map to identify regions of interest, set up multiposition coordinates, or define mosaic boundaries without exposing the entire sample to high-intensity light or capturing full-resolution data.



Setting up a Preview Scan

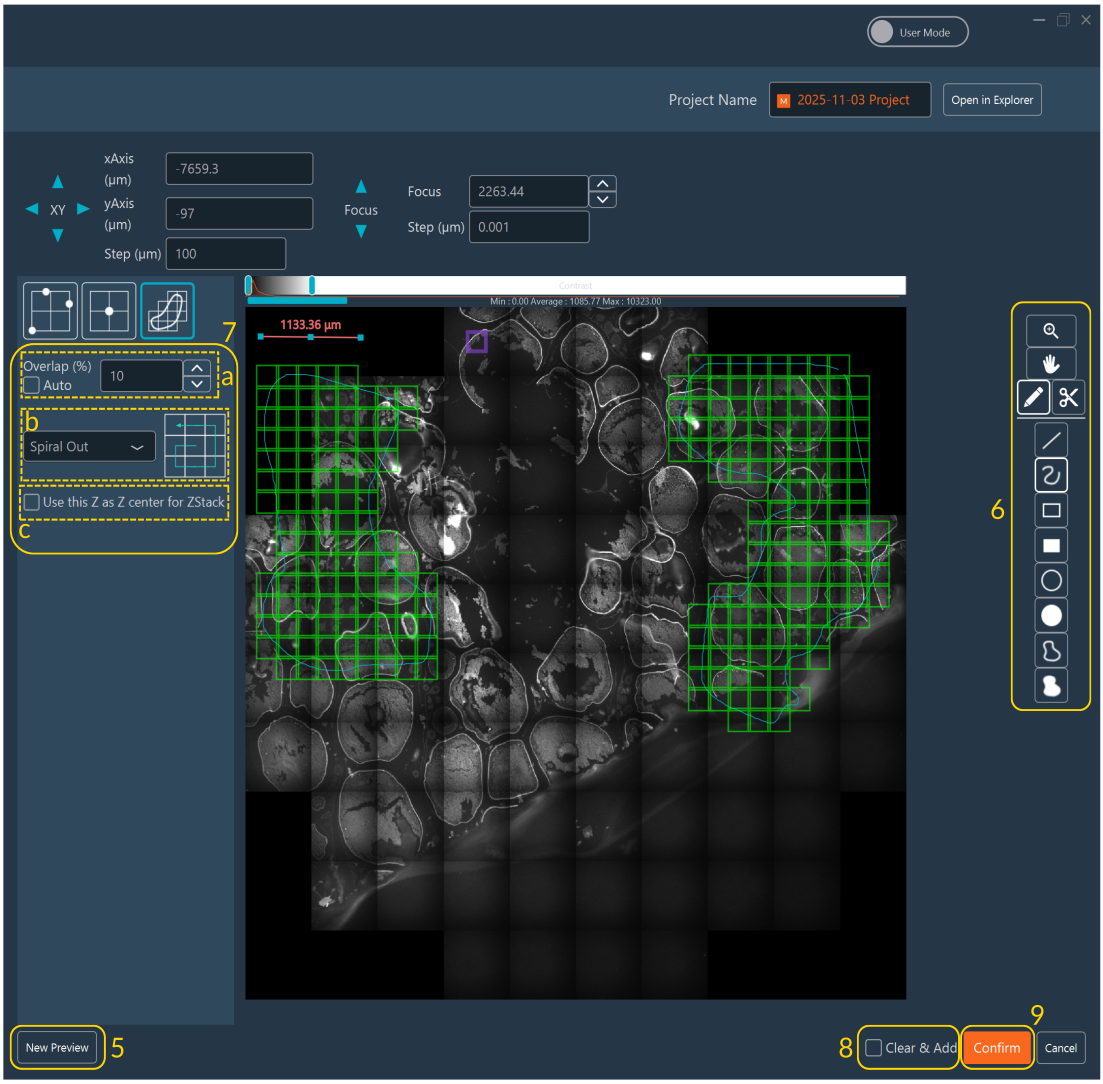
1. Configure the boundaries of the **Preview** area. Similar to the Tiling dimension, you have two setup options:
   - Click **Add** at multiple boundary points while moving the stage to trace a custom polygon outlining your region of interest.
   - Alternatively, click **Set** to define the center coordinate, then specify the bounding grid dimensions (horizontal/vertical tile count).
2. Select a **Read Mode** for how the stage traverses the area (e.g., Spiral, Snake).
3. Click **Confirm** to lock in the configuration.

All rights reserved.



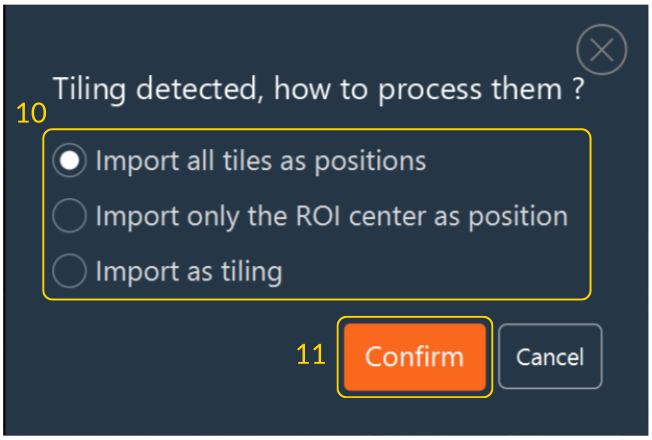
4. The preview scan begins, rendering the map in real-time. You can abort the acquisition at any time by clicking **Cancel**.

All rights reserved.



5. Click **New Preview** to discard the current map and start over in a different region.
6. Draw **Regions of Interest (ROIs)** directly onto the map to mark targets. Use the provided tools: **point** (ideal for discrete focal points used in the Positions dimension), **freehand**, **circle**, or **rectangle**.
7. Adjust the overlapping tiling settings for the selected ROIs:
    1. Enable **Auto Mode** to let the software calculate the image overlap threshold (available only for Tiling imports).
    2. Choose the specific scanning **Read Mode**.
    3. Check the **Z-reference** box to explicitly center subsequent Z-Stacks around the Z-value established during this preview/focus map phase. If unchecked, the system reverts to the global Z-Stack parameters.
8. Check **Clear & Add** if you want these new targets to overwrite previously defined tilings or positions. Leave it unchecked to append them.
9. Click **Confirm** to validate the ROI selections.
10. If you drew ROIs that encompass multiple camera tiles but intend to import them into the **Positions dimension**, the software will prompt you on how to handle the spatial discrepancy:
    - **Import all tiles as positions**: Generates a discrete capture point for every tile covered by the ROI area.
    - **Import only ROI center as position**: Registers a single coordinate at the geometric center of the ROI.
    - **Import as tiling**: Reroutes the coordinate set to the Tiling dimension for continuous mosaic acquisition.

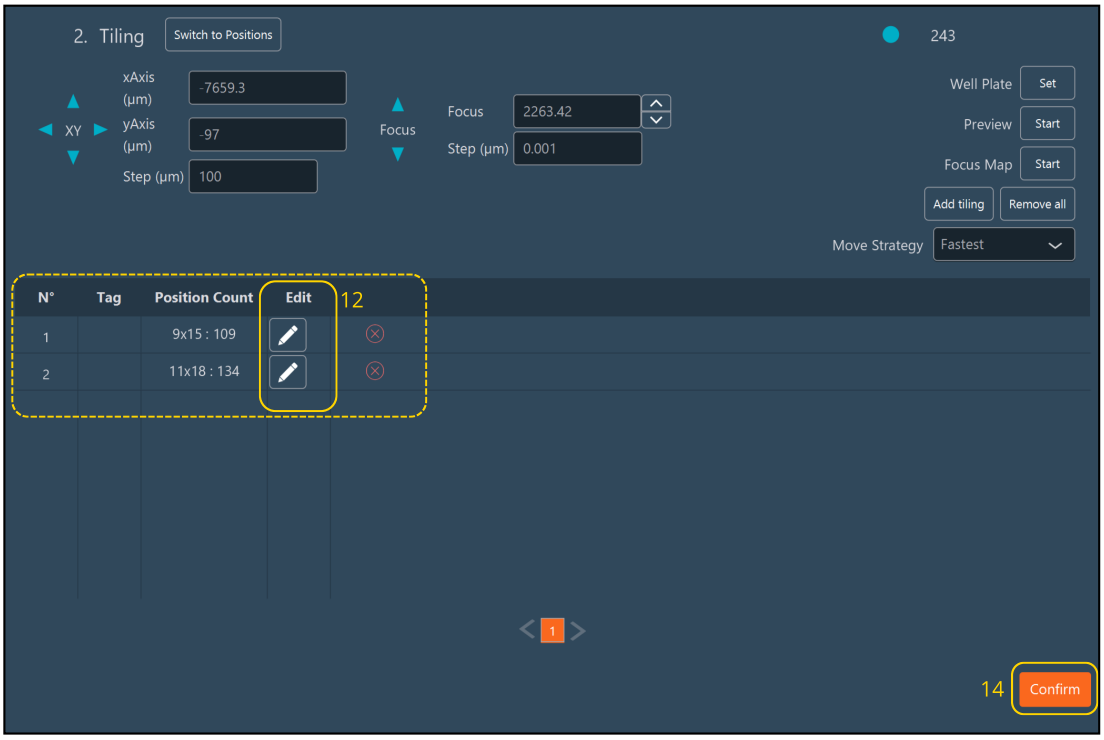
All rights reserved.

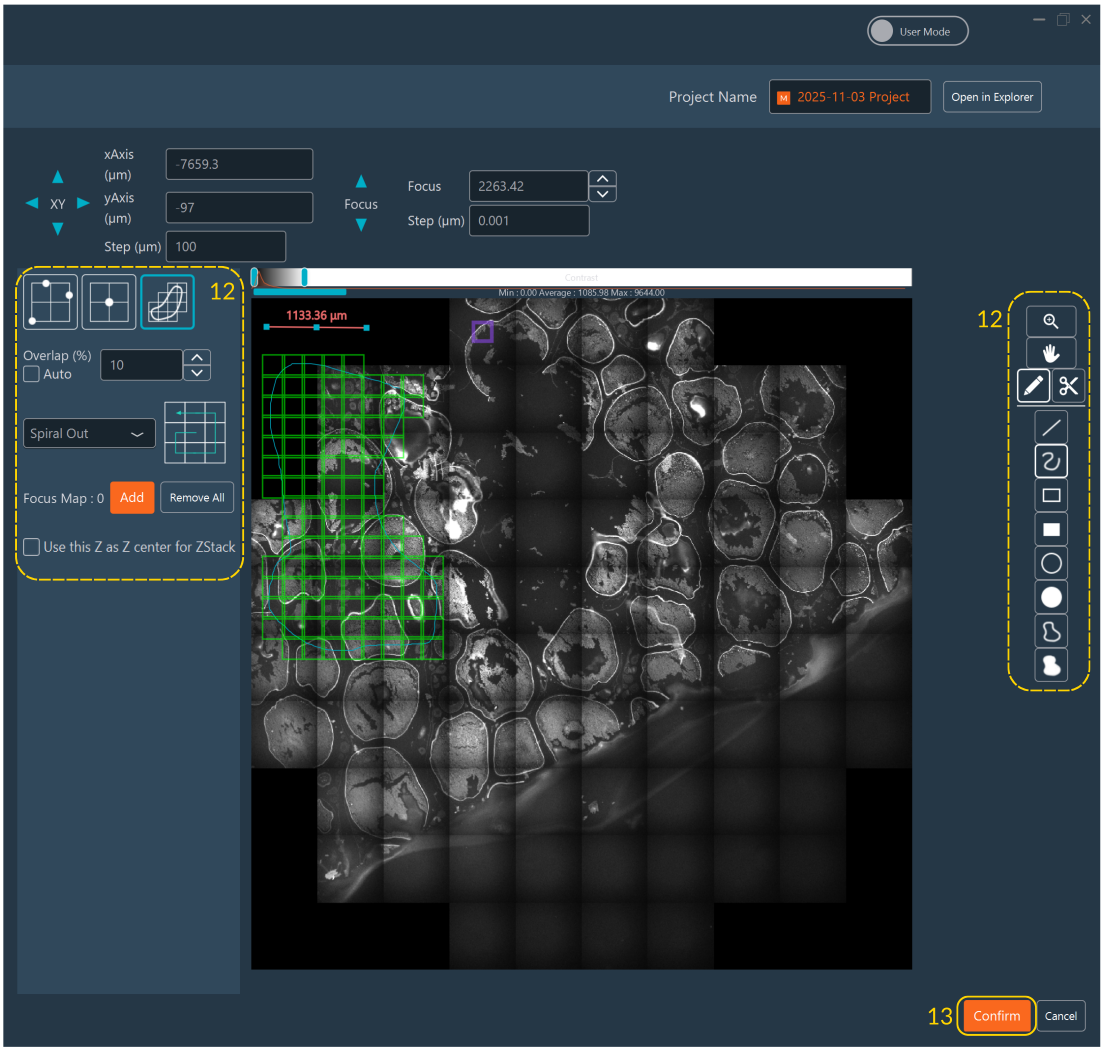


11. Click **Confirm** to apply your routing choice. After confirmation, all generated positions or tilings appear in their corresponding dimension tabs.
12. In the Tiling dimension tab, click the **Pen** icon next to an imported entry to edit it against the original preview map:
    - **Move**, **modify**, or **delete** an ROI visually.
    - Change the scanning **Read Mode**.
    - Add a local **Focus Map** within the ROI if the sample is not flat, or if the Z-focus plane needs adjusting across the ROI.

    Select whether to use the Z-position saved during this preview as the absolute center of your Z-Stack. If left unchecked, the Z-Stack will default to the Z-value defined in the standalone Z-Stack Dimension parameters.

All rights reserved.





13. Click **Confirm** to save your modifications to the individual ROI.
14. Finally, click **Confirm** in the main panel to validate the entire **Tiling Dimension** sequence.

Tip

To save the visual output of your preview scan as a reference image, **right-click** anywhere on the preview rendering window.

All rights reserved.

2026-09-08

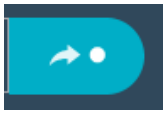
All rights reserved.

1.4.7 Multiwell Plate Mode

Overview

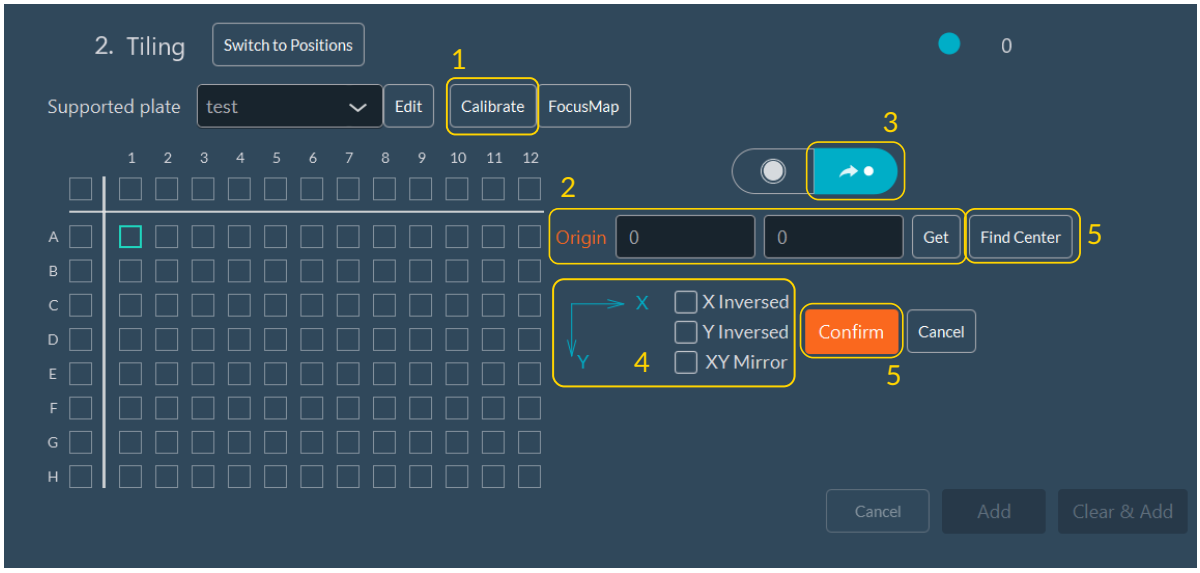
The **Multiwell Plate Mode** automates the imaging of standardized microplates. Access this feature from either the **Positions** or **Tiling** dimensions by clicking the **Set** button.



Button	Description
	**Select Wells**: Click to highlight the wells you intend to image.
	**Auto Move**: Once the plate is calibrated, activate this toggle to automatically physically move the XY stage when you select a well.
	**Unselected**: The well is excluded from the acquisition.
	**Selected**: The well is included in the acquisition.
	**Current Position**: Indicates the real-time position of the objective relative to the plate.

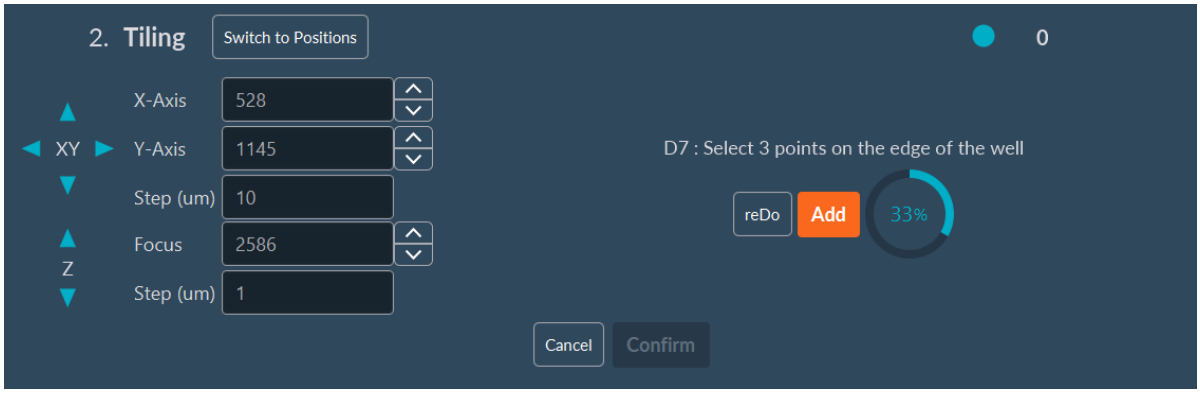
Calibration

Calibrate the system to map the virtual coordinates of the software to the physical dimensions of the well plate. Without calibration, XY stage moves do not land on the intended well.



1. Click **Calibrate** to align the stage movement with the multiwell plate.
2. Click **Get** to set the origin point of your plate.
3. **Move** the stage from the origin to a different well to test the alignment.
4. If the stage movement is not correct, **adjust** the orientation.
5. Click **Confirm** to save the stage alignment, or click **Find Center** to calibrate the center of the well.

All rights reserved.



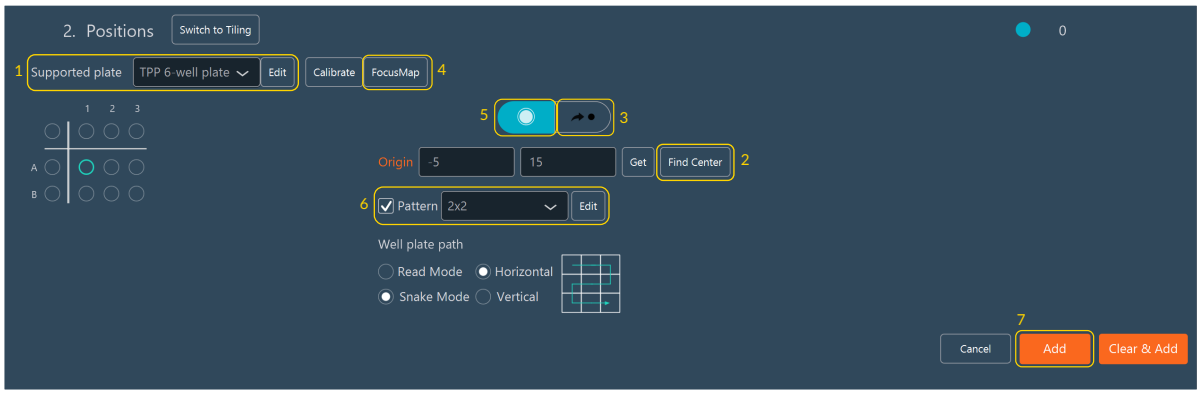
HOW TO PERFORM FIND CENTER CALIBRATION?

- Click **Find Center** to start the semi-automated calibration.
- Select a well on the virtual map to serve as the calibration reference.
- Move the stage to that well, if it is not already there.
- While viewing the live image, move the XY stage to align the virtual crosshair with the edge of the well, then click **Add**.
- Repeat this alignment process two more times on different sides (e.g., top, bottom, left) of the same well to establish the geometry.
- Click **Confirm** to complete the calibration.

Once calibration is complete, the software navigates to any well on the plate.

Multiwell Mode in the Positions Dimension

Use this mode to capture distinct, single-field-of-view images within designated wells.

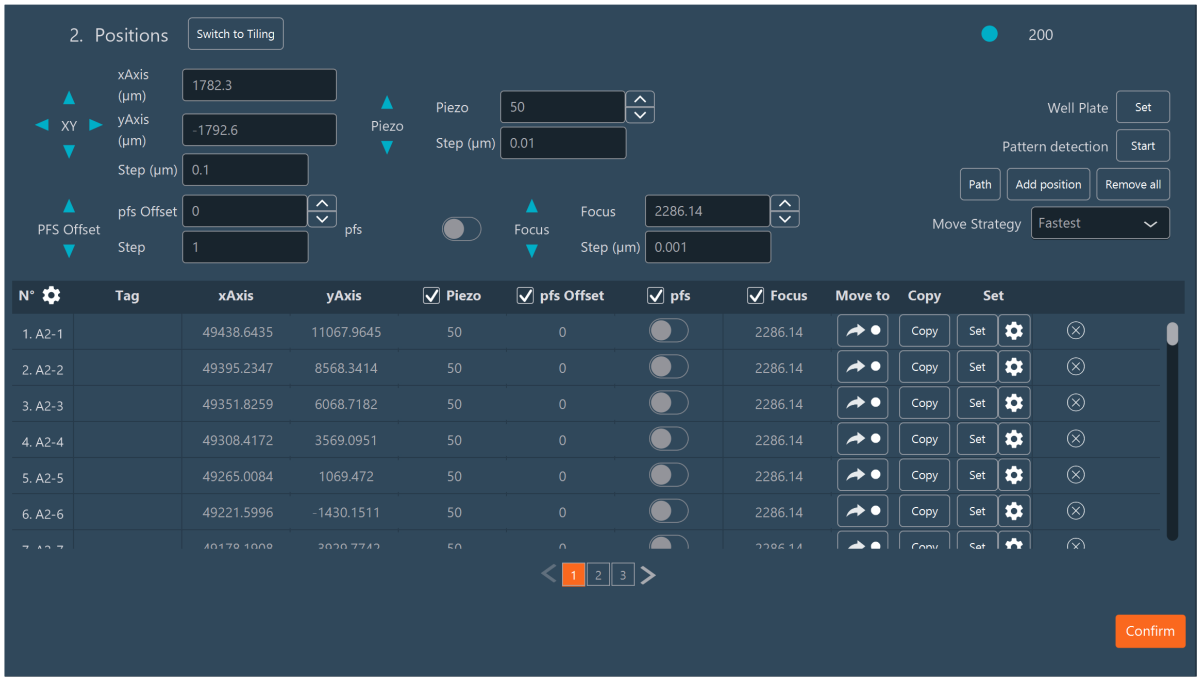


1. Select the plate format (e.g., 96-well, 384-well) from the **Supported plate** dropdown list.
2. Calibrate the well plate using the **Find Center** tool. If this has not already been done.
3. You can now automatically move the stage directly to specific wells.
4. Create a **Focus Map** for the desired wells, if necessary. A focus map maintains focus across uneven plate bottoms.
5. Select the wells to image on the virtual map.
6. To generate an automated imaging pattern within each well, check the **Pattern** box and choose the pattern desired in the drop-down menu. If another pattern is needed, ckick **Edit** to modify existing pattern or create a new one.
7. Click **Add** to confirm the settings and populate the position list.

> **Tip**
>
> **Fast Selection**: In the virtual well plate map, click a row letter or column number to select the entire row or column. You can also click the top-left corner to select the entire plate. To select multiple discontinuous wells, click and drag your mouse to create a selection box.

All generated coordinates are added to the Positions dimension list.

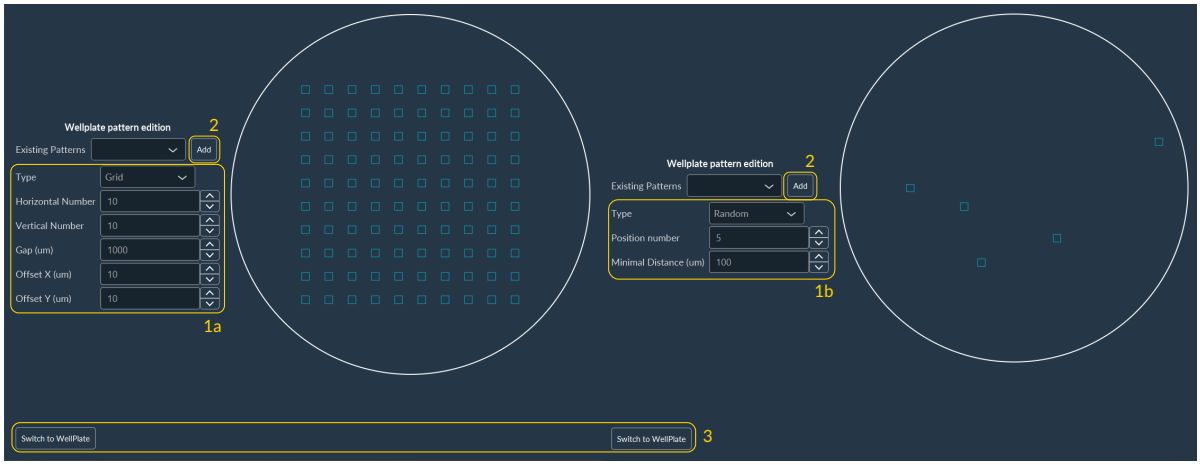
All rights reserved.



EDITING A PATTERN

Click **Edit** next to a pattern to modify its distribution parameters:

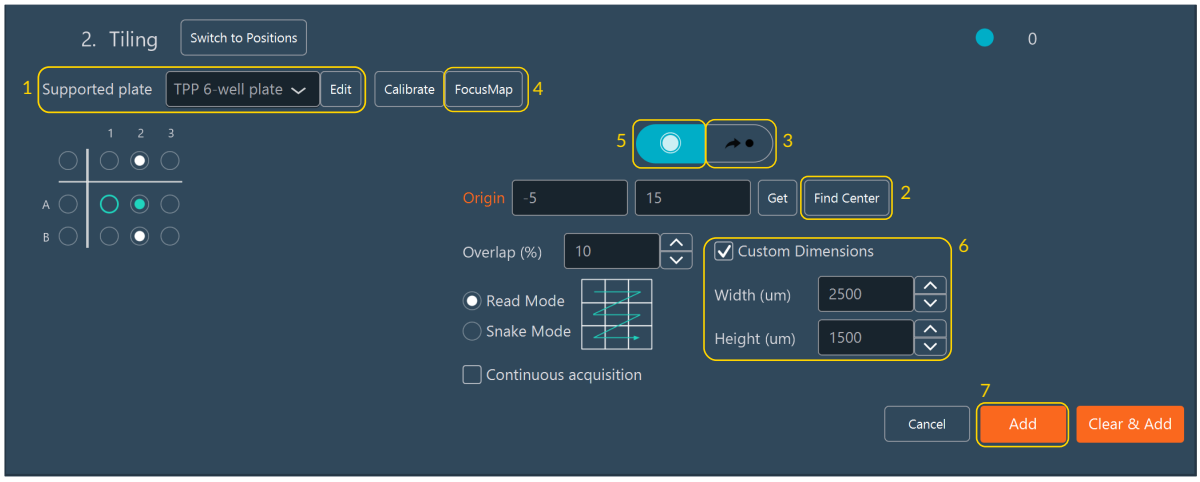
1. Select the pattern strategy: **Grid** or **Random**.
   - For a **Grid** (a), specify the number of horizontal and vertical images, the spacing distance between adjacent positions, and the X/Y offset (to translate the entire grid relative to the well center).
   - For a **Random** pattern (b), specify the total number of positions and the minimum required exclusion distance between them to prevent overlap.
2. Click **Add** to update the pattern.
3. Click **Switch to WellPlate** to return to the interactive plate overview.



Multiwell Mode in the Tiling Dimension

Use this mode to acquire large, stitched mosaic images covering entire wells or significant portions of them.

All rights reserved.

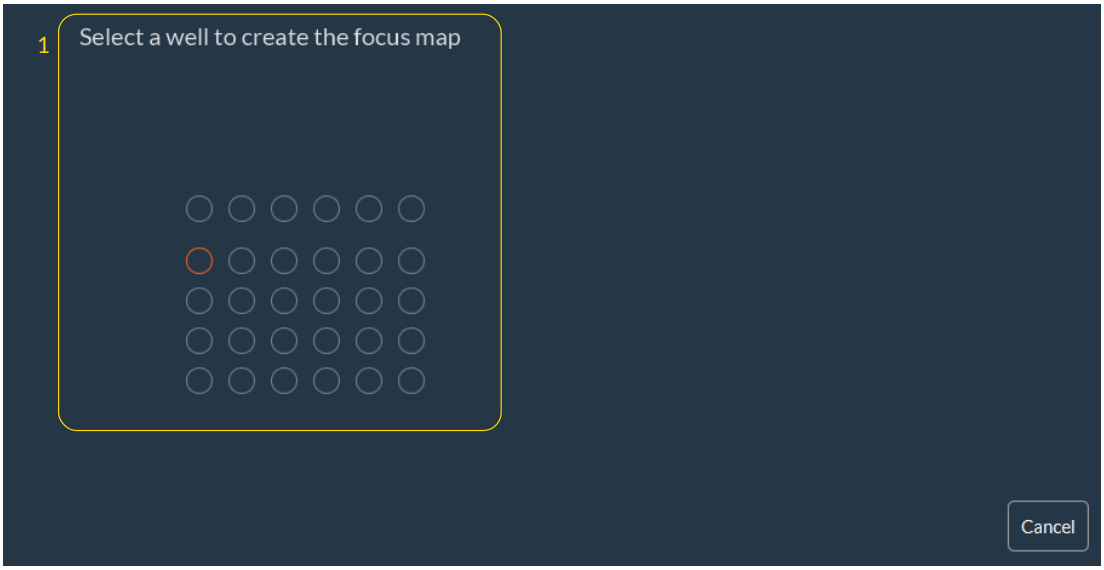


1. Select your plate format from the **Supported plate** dropdown list.
2. Calibrate the well plate using the **Find Center** tool. If this has not already been done.
3. You can now automatically move the stage exactly to specific wells.
4. Create a **Focus Map** for the desired wells, if necessary.
5. Select the wells you intend to image on the virtual map.
6. Check the **Custom Dimensions** box to configure the specific tiling size, tile overlap percentage, and scanning read mode to be applied uniformly within each selected well.
7. Click **Add** to confirm your settings and generate the tiling matrices.

Manual Focus Map Setup

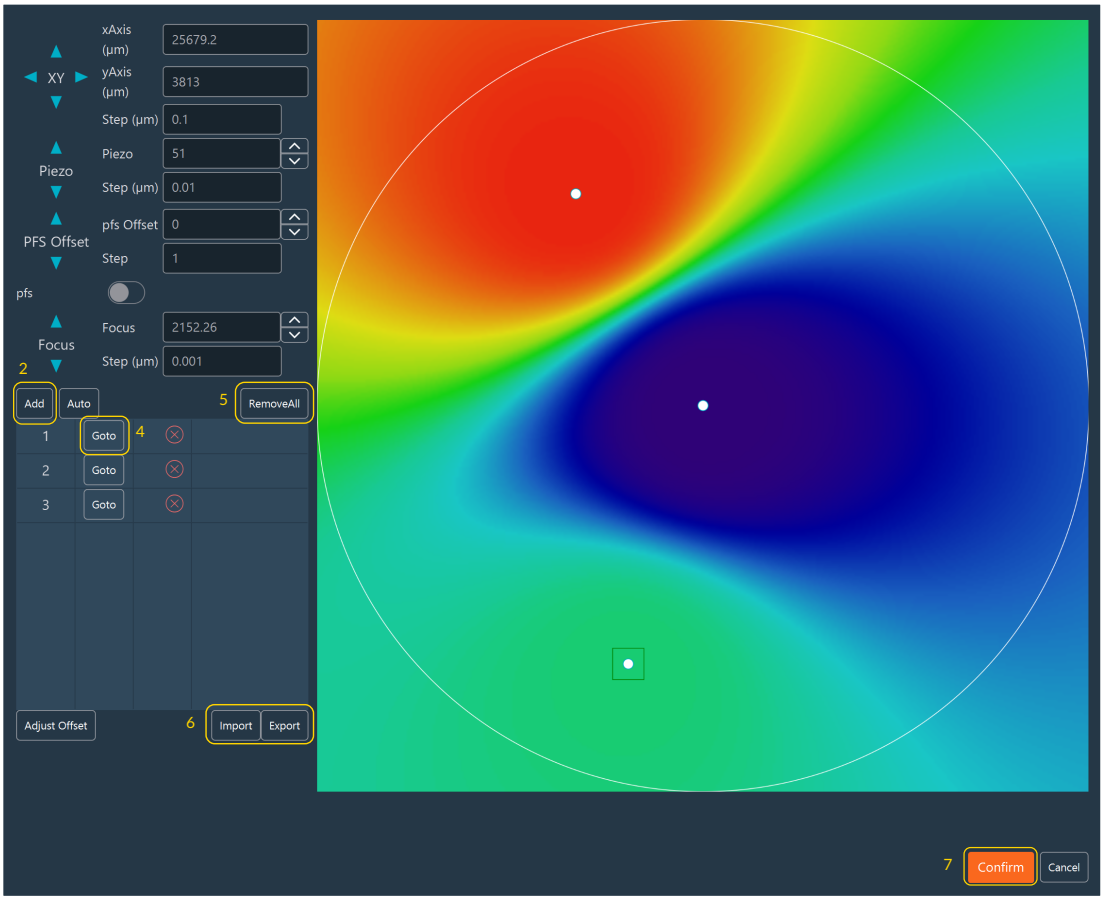
A Focus Map creates a topological Z-surface to compensate for sample drift or uneven plate substrates.

1. Select the specific well for which you want to create a Focus Map.



2. Move the stage to a desired topological anchor point within the well, manually adjust the focus until sharp, and click **Add**.
3. Repeat this process at multiple disparate points until the Focus Map adequately covers the imaging area. The software interpolates the focus between these points.
4. Click **Go to** to move the stage back to a previously saved focus anchor point for verification.
5. Click **Remove all** to clear all points from the current map.
6. You can also import or export a saved Focus Map using the respective buttons (useful for standardized, recurring assays).
7. Click **Confirm** to lock in your Focus Map.

All rights reserved.

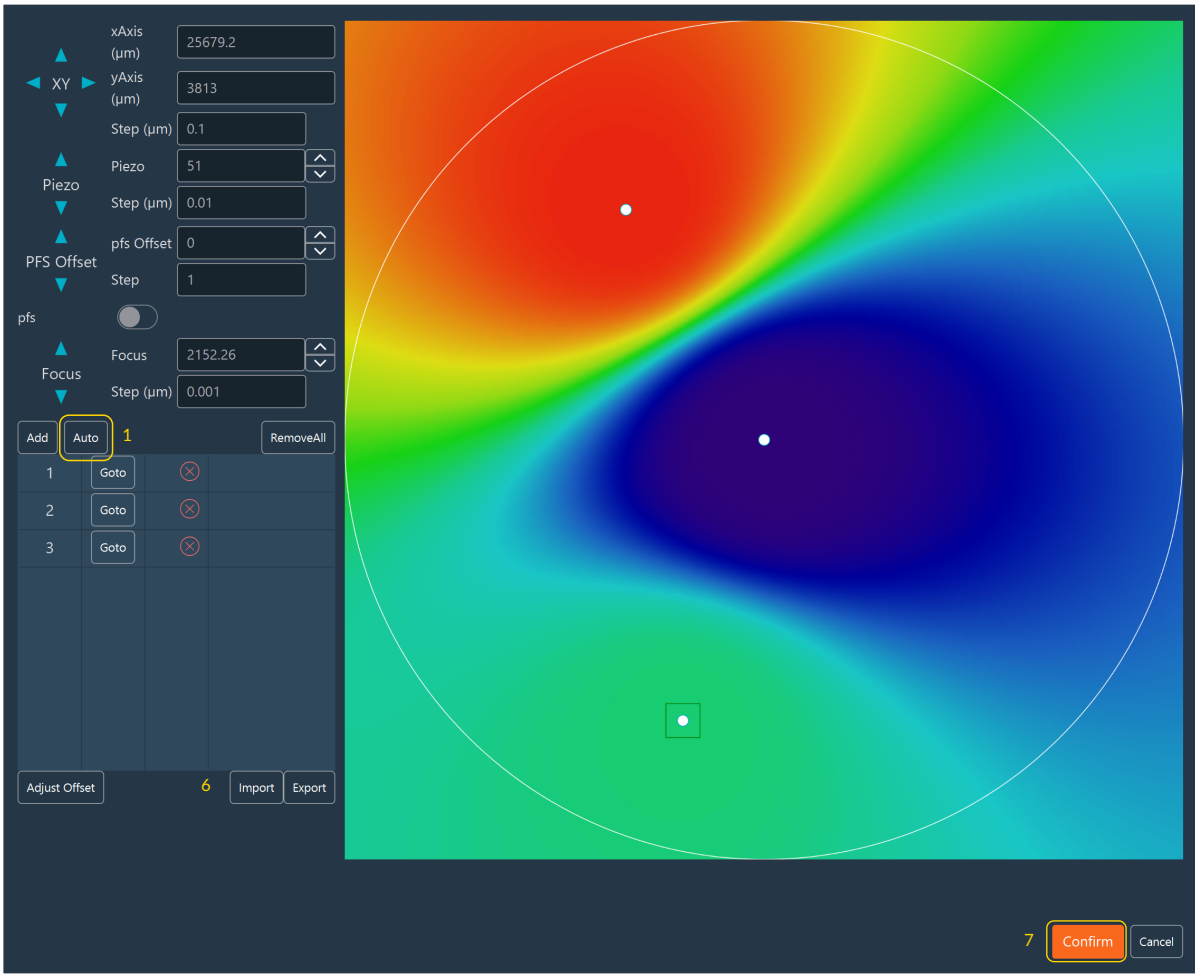


Automated Focus Map Setup

The **Automated Focus Map** is a software feature of the Inscoper I.S. that automatically generates a hardware-driven relief map of the biological sample before executing tiling imaging.

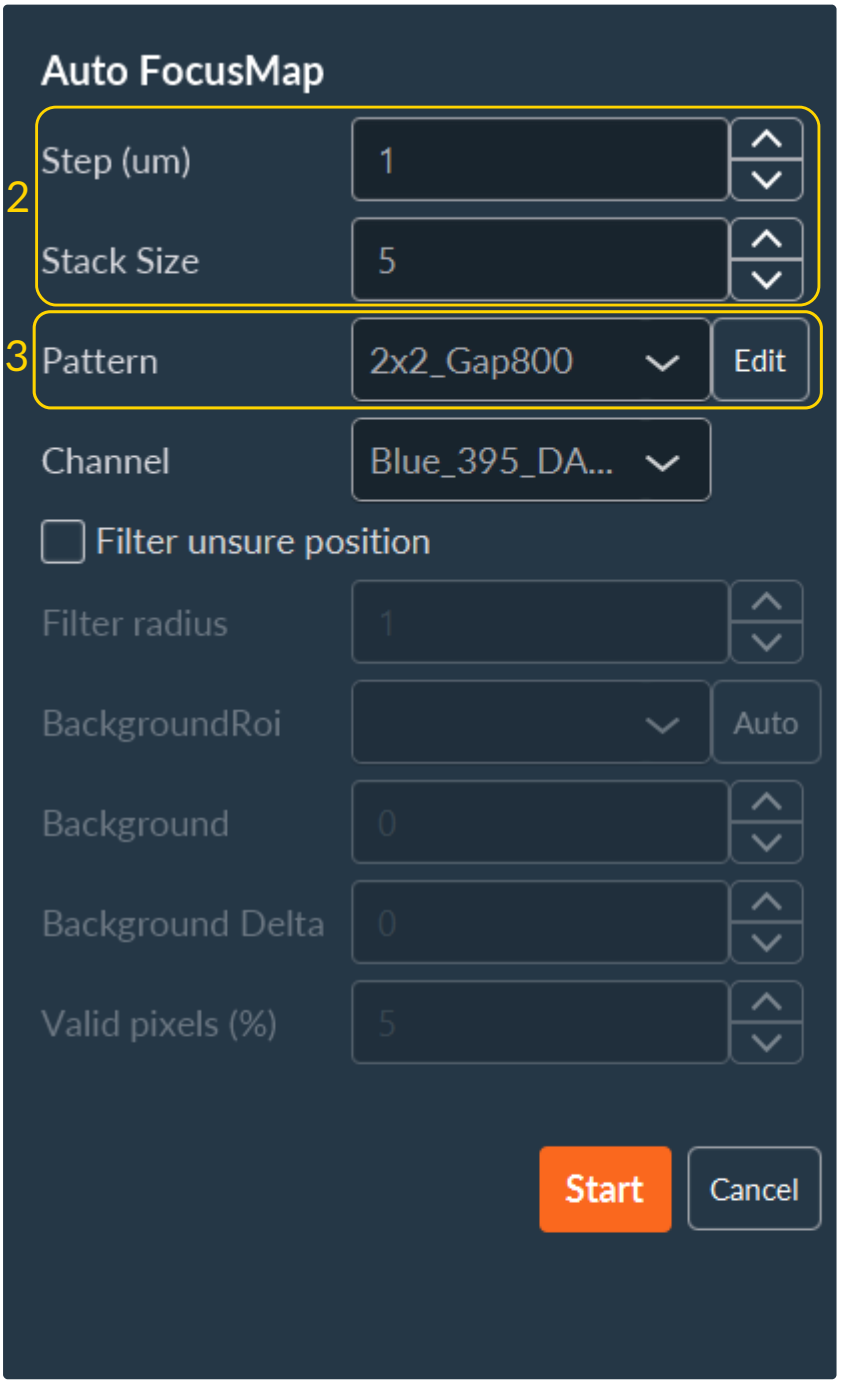
auto on one well:

All rights reserved.



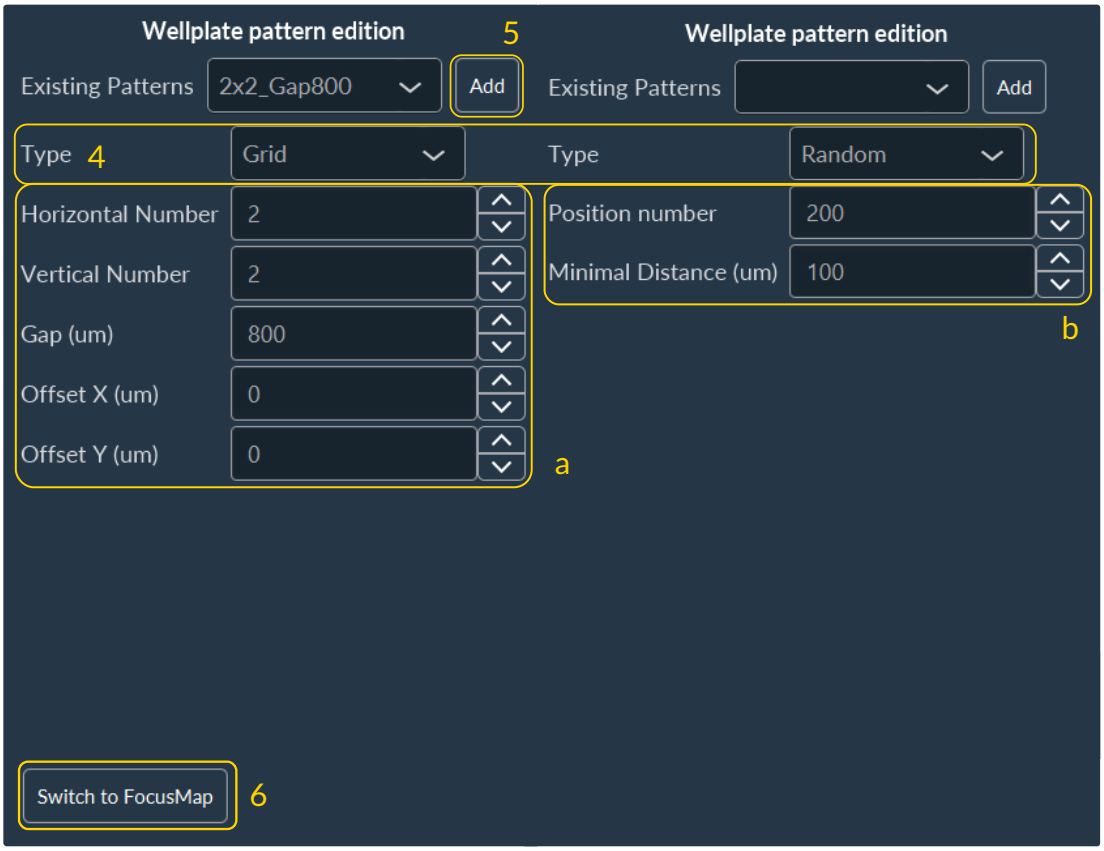
1. click Auto
2. Define the **step** interval and **size** for the Z-stack.
3. Select an existing sampling pattern from the dropdown list (and proceed to step 9), or click **Edit** to define a new spatial pattern.

All rights reserved.



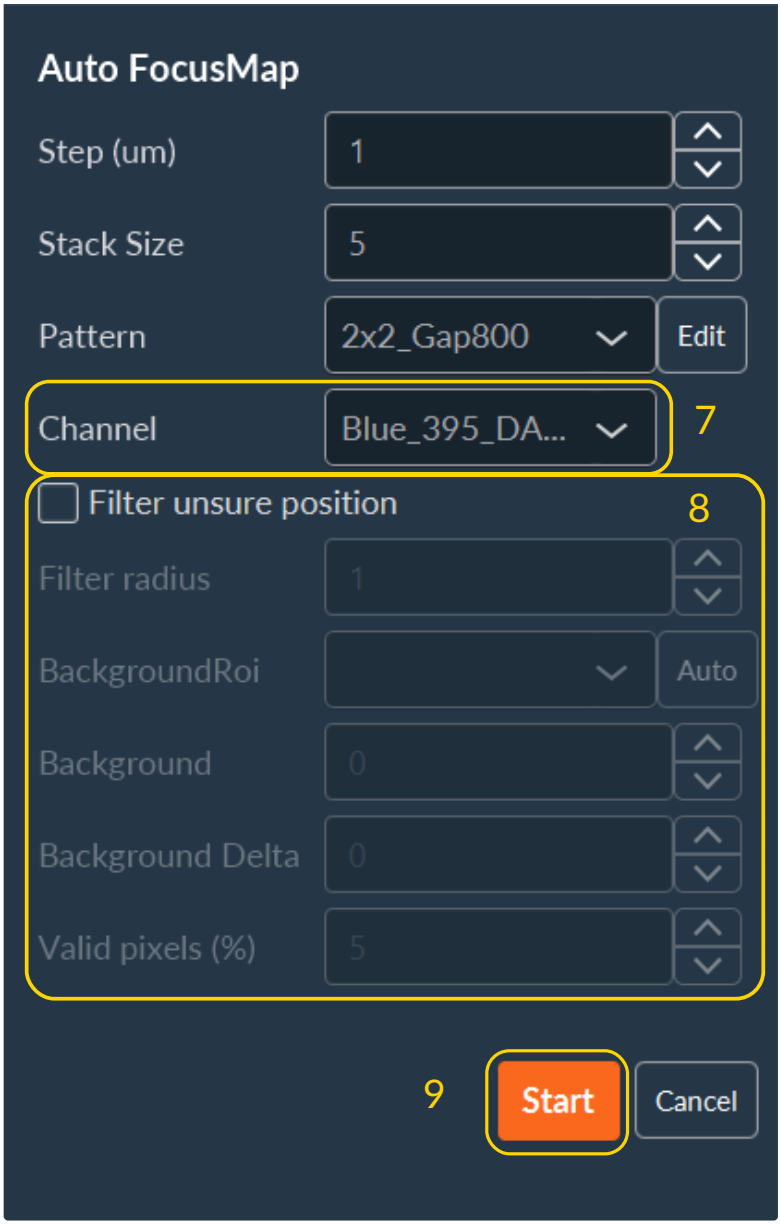
4. If creating a new pattern, select either a **Grid** or **Random** geometric distribution:
    - **Grid (a):** Specify the total number of images to acquire horizontally and vertically, along with the gap distance between adjacent image boundaries. The X and Y offset values translate the entire pattern grid across the well.
    - **Random (b):** Specify the total number of positions and the minimum required exclusion distance between coordinate points.
5. Click **Add** to save the distribution pattern.
6. Click **Switch to FocusMap** to return to the primary dimension configuration.

All rights reserved.



7. Select the **channel** designated for acquiring the Z-stack during the Focus Map generation.
8. To reject optical artifacts and empty fields, select the **Filter unsure position** checkbox and configure the rejection parameters:
    - **Filter Radius:** Applies a spatial median filter to reduce high-frequency noise before evaluation (e.g., 0 = no smoothing; 1 = median based on a 3x3 pixel kernel; 2 = median based on a 5x5 pixel kernel).
    - **Background ROI:** Automatically calculates absolute Background and Background Delta values by evaluating a user-drawn Region of Interest (ROI). The Background equals the mean pixel intensity of the ROI, and the Background Delta equals two times the standard deviation.
    - **Background:** Specifies a static mean background intensity value.
    - **Background Delta:** Specifies the permitted noise amplitude threshold.
    - **Valid Pixels:** Sets the required percentage of pixels whose intensity must exceed the background threshold to validate the coordinate position.
9. Click **Start** to generate the focus map.

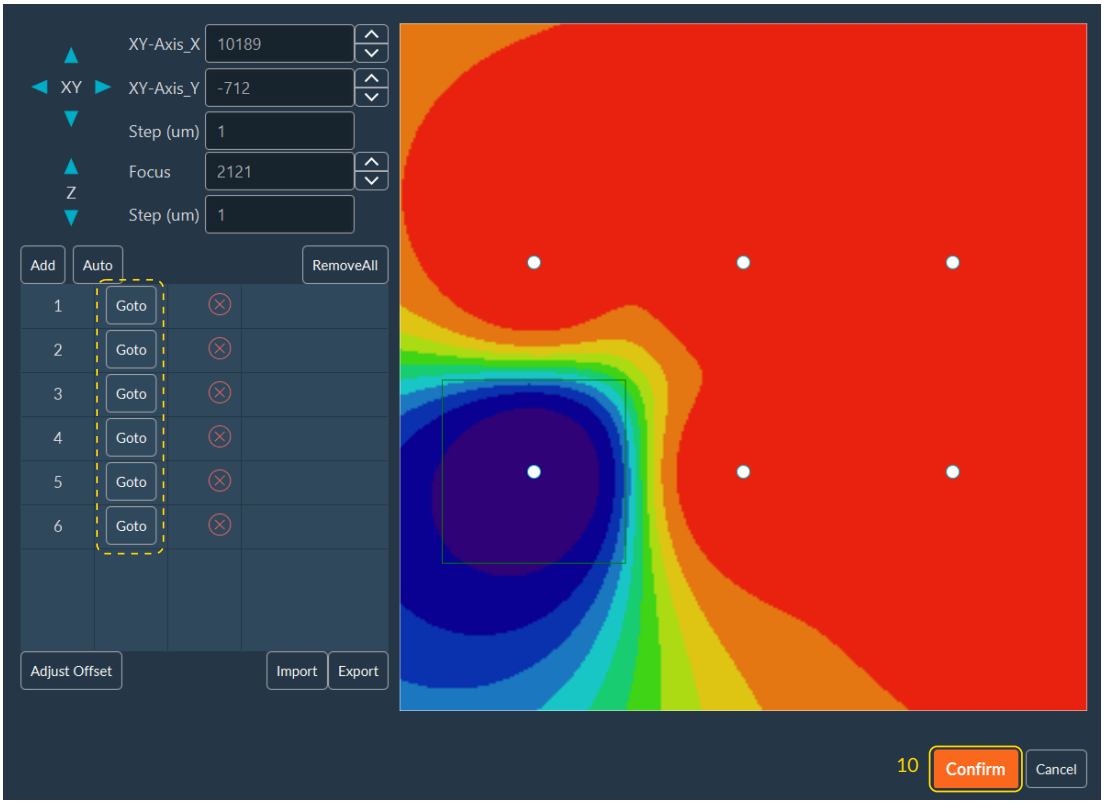
All rights reserved.



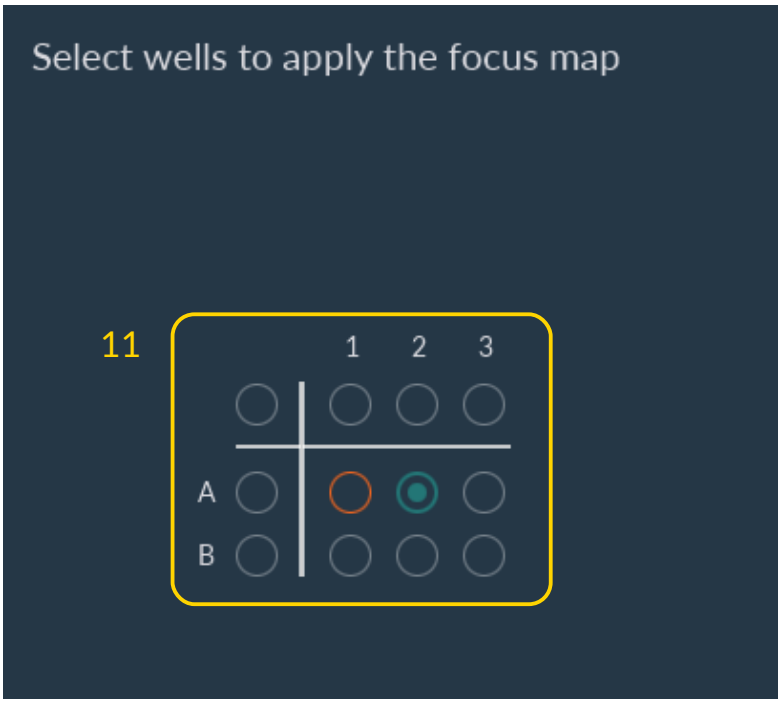
The system navigates to all defined coordinates to create the Focus Map and displays a list of validated anchor points. Click **GoTo** adjacent to the coordinate or click directly on the thumbnail projection to verify physical focus at any anchor point.

10. Click **Confirm** to finalize the generated Focus Map.

All rights reserved.

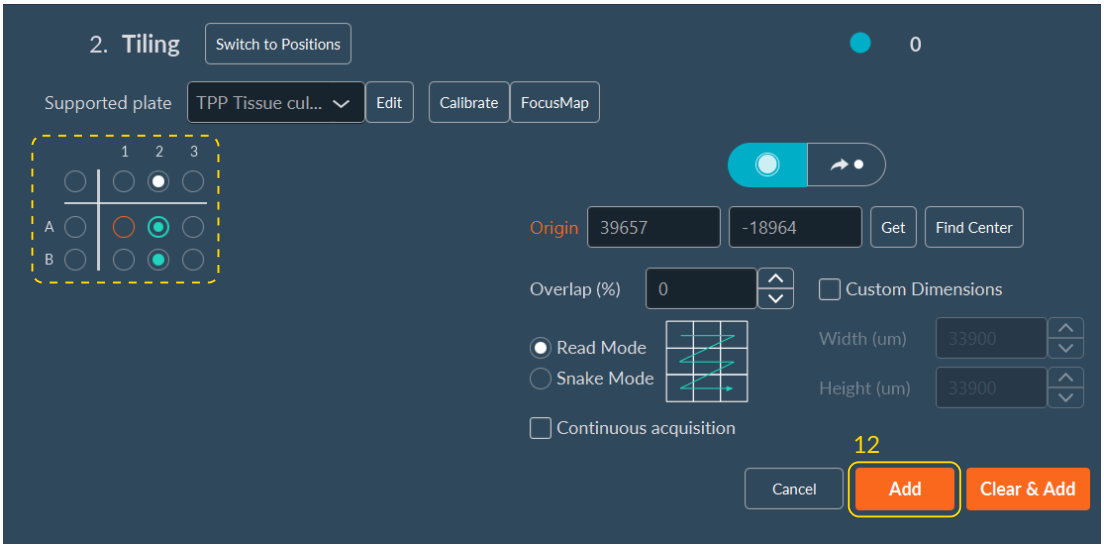


11. Select the wells to apply the Focus Map to, then click **Confirm**.

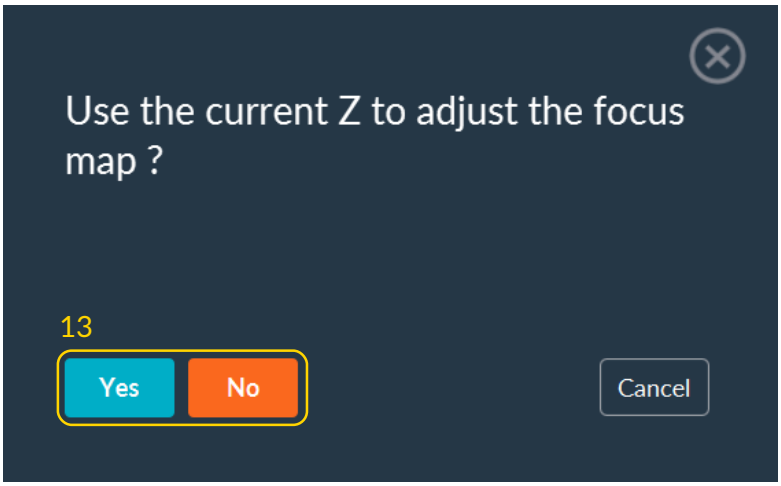


12. Select the wells to acquire, then click **Add**.

All rights reserved.



13. Select whether to use the current Z-position to adjust the Focus Map.



After this step, you can continue configuring your acquisition sequence.

2026-09-08

All rights reserved.

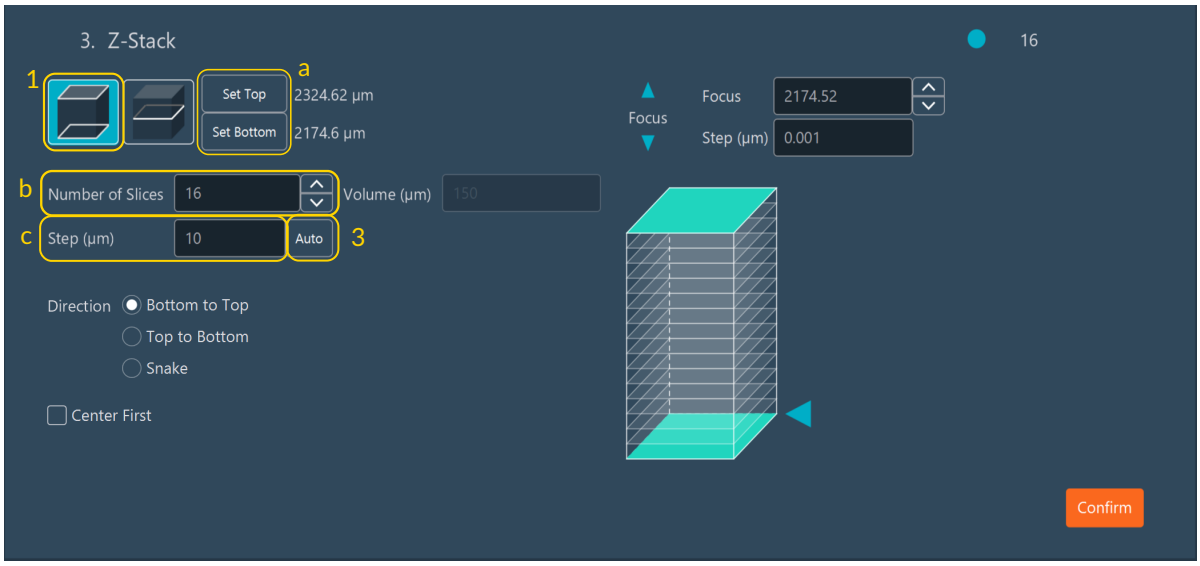
1.4.8 Z-stack

Use the **Z-Stack** dimension to capture volumetric data by acquiring a series of slices along the Z-axis.

Configuring a Z-Stack

Configure the volumetric boundaries using one of two primary methods:

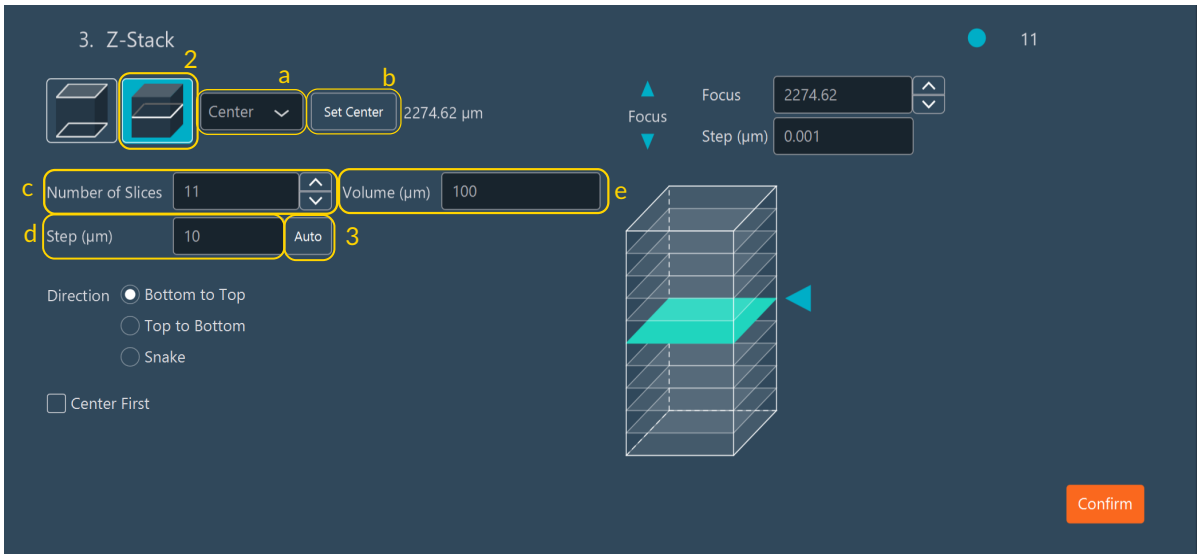
DEFINE TOP AND BOTTOM LIMITS



Move the focus to the desired upper and lower focal limits (1):

- a. click **Set Top** and **Set Bottom** to establish the boundaries.
- b. Specify the **number of slices** (focal planes) within this volume.
- c. Set the **step size** (in µm) between adjacent focal planes. Enter your own value or click **Auto** (3) to use Nyquist Calculation.

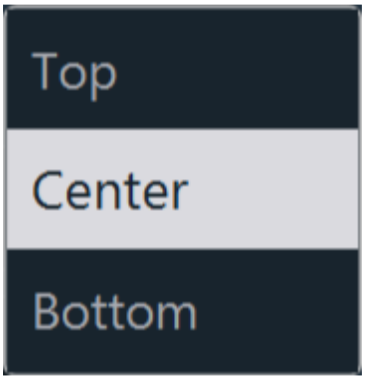
DEFINE VIA A REFERENCE PLANE



Move the focus to the desired center focal plan (2):

- a. Select whether the current live focal point represents the **center**, **top**, or **bottom** of the intended Z-stack.

All rights reserved.



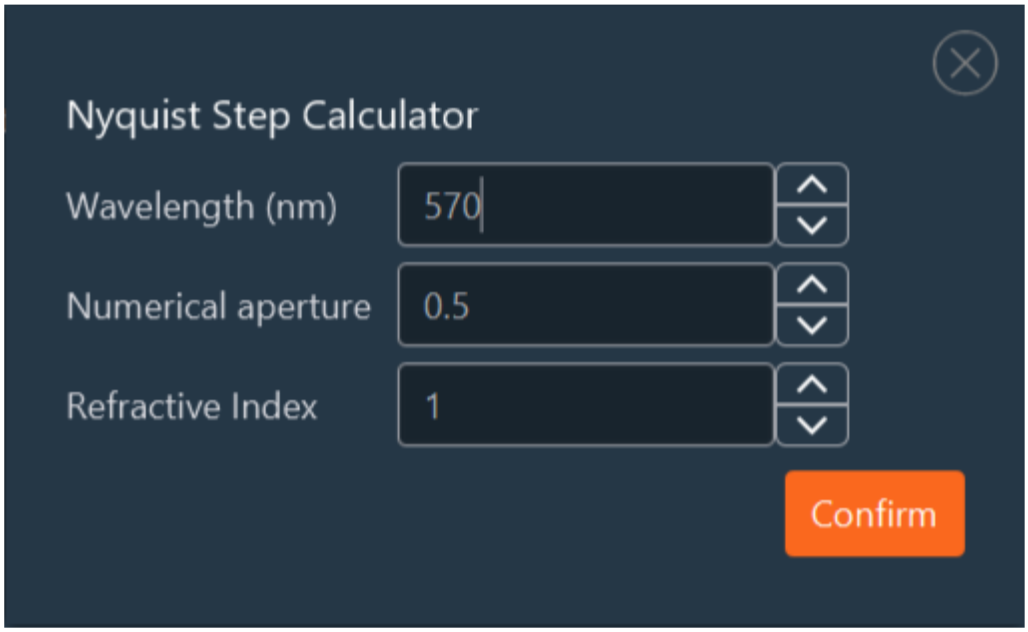
- b. Click the corresponding registration button (**Set Center**, **Set Top**, or **Set Bottom**) to log the current focal plane as the anchor.
- c. Specify the total **number of slices**.
- d. Set the **step size** (in µm). Enter your own value or click **Auto** (3) to use Nyquist Calculation.
- e. Alternatively, define the **volume depth** of the Z-stack (in µm). The software computes the remaining variables automatically.

NYQUIST CALCULATION

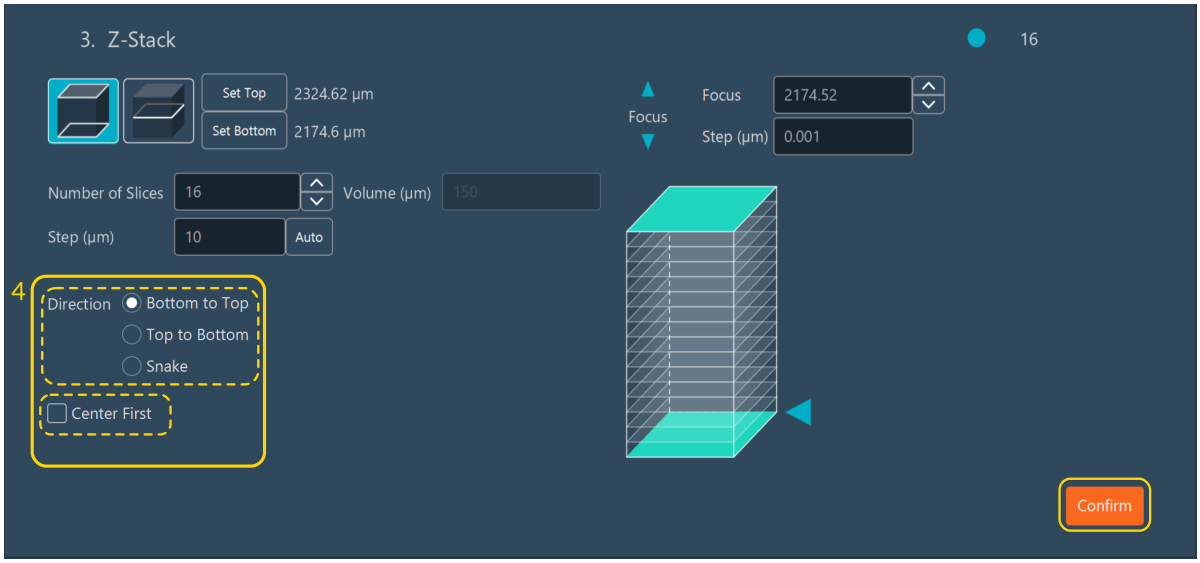
Use this method to automatically compute the optimal step size for maximum 3D optical resolution without oversampling the volumetric data. Click **Auto** to generate the ideal Nyquist Z-sampling rate.

- The software calculates the ideal physical step size based on the **longest emission wavelength** configured in the multi-channel parameters.
- These values remain **editable**. Manually override the calculated step size or tune the inputs, such as the emission wavelength, the objective's numerical aperture (NA), and the refractive index (RI) of the immersion medium.
- Click **Confirm** to save the computational settings.

All rights reserved.



Execution settings



Once the geometric volume is defined, set the execution parameters (4):

- Select the **acquisition direction**: either **Top to Bottom**, **Bottom to Top**, or **Snake** (bidirectional alternating)
- Enable **Center First** to force the sequence to begin acquisition at the **central focal plane**.

> **Tip**
>
> This setting is required with hardware autofocus systems (e.g., PFS, Definite Focus), which must anchor before the volume is swept.

Click **Confirm** to save the Z-stack configuration block.

> **Tip**
>
> Use the interactive **Z-stack diagram** to verify the setup — drag the **blue arrow** or click the diagram to drive the stage and preview the stack bounds.

> **Important**
>
> When configuring a Z-stack from a center reference plane, the **Volume**, **Step**, and **Number of Slices** fields are linked. Adjusting one recalculates the others. If the **Positions dimension** is active and the Z-axis override is enabled for a position, the local focus value of that position overrides the center coordinate of the Z-stack.

All rights reserved.

2026-09-08

All rights reserved.

1.4.9 Software Autofocus

Use the **Software Autofocus** module to keep your sample in focus during the acquisition sequence. To calculate Z-axis corrections, the software captures autofocus evaluation frames on the fly. These frames are processed in memory and are not saved to the disk.



Configuring Software Autofocus

1. Specify the number of Z-planes to acquire and analyze in the **Focus Number** field.
2. Define the Z-axis step size in the **Step** field (in the units specified in the software settings) to determine the distance between successive planes.
3. Select the active **Channel** used to evaluate focus.
4. Select the active **Camera** and specify the sensor **Exposure** time (in ms) for the autofocus capture frames.
5. Select the autofocus algorithm from the **Autofocus Algo** dropdown menu:
    - Maximum: Optimized for bright fluorescent signals.
    - Sharpness: Optimized for dense morphological structures (such as brightfield or phase contrast), where the algorithm detects the highest high-frequency contrast.

    > **Note**
    >
    > The sharpness algorithm calculation is performed exclusively on the central 20% area of the image.

6. Under **Do Autofocus**, specify the execution frequency of the autofocus routine. Enter the interval in the **Every Time** and **Every Position** fields to define how often the routine executes relative to time points and stage positions.
7. Select the **Advanced Parameters** checkbox to configure mechanical device integration:
    - Under **Do autofocus with**, select the mechanical device that drives the autofocus routine (such as a fast piezo stage or the microscope internal Z-drive).
    - Under **Apply offset to** (optional), select the mechanical device to which the system applies the calculated Z-offset.
8. Click **Test AutoFocus** to test the autofocus protocol on the sample immediately.
9. Click **Confirm** to save the configuration.

> **Note**
>
> Please note that if you are using a software mirror for your camera, the autofocus image will not have the same orientation as your live view.

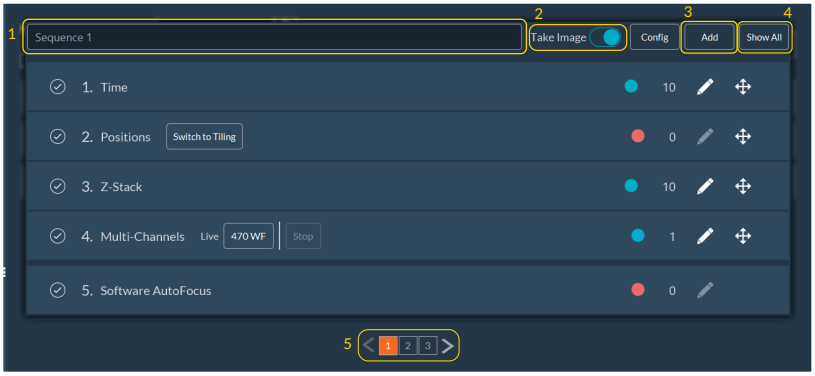
All rights reserved.

2026-09-08

All rights reserved.

1.4.10 Multi-Sequence Acquisition

After configuring an isolated acquisition sequence, launch it immediately or append additional sequences to build a **Multi-Sequence Acquisition** for automated workflows.



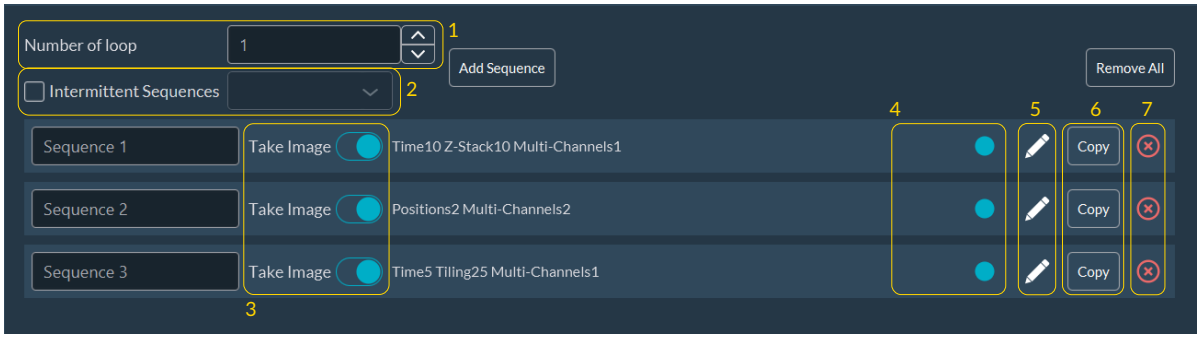
Each sequence within the interface includes the following setup options:

1. **Rename**: Assign a name to the sequence for identification during the experiment.
2. **Toggle Image Acquisition**: Disable or enable image capture for a specific sequence by toggling the **Take Image** button (useful for setting up delays or hardware moves without saving data).
3. **Add Sequence**: Append a new, empty sequence to the multi-sequence stack.
4. **Show All**: View a summary of all prepared sequences in a bottom panel.
5. **Navigation**: Jump between sequences. The currently selected sequence highlights in orange.

Clicking **Show All** (4) opens the **Summary Tab**, providing sequence management:

1. **Loop Count**: Define how many times the entire multi-sequence stack should loop.
2. **Intermittent Sequences** (Optional): Check this box to trigger the sequence intermittently based on a selected dimension (e.g., execute this sequence only on every 3rd time point).
3. **Take Image Toggle**: Quickly enable or disable image capture for individual sequences from the summary view.
4. **Active Status**: Activate or deactivate an entire sequence by clicking its colored indicator (red indicates inactive, blue indicates active).
5. **Edit**: Click the **Pen** icon to open the configuration for that sequence.
6. **Duplicate**: Clone an existing sequence by clicking **Copy**.
7. **Delete**: Remove a sequence by clicking **Delete**.

All rights reserved.



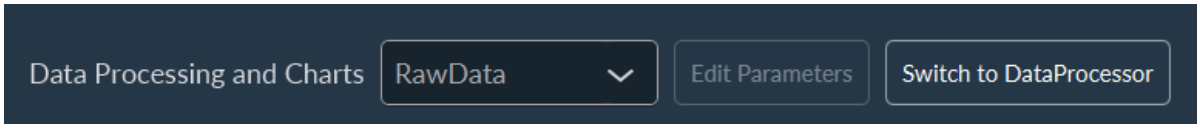
2026-09-08

All rights reserved.

1.4.11 Data Processor

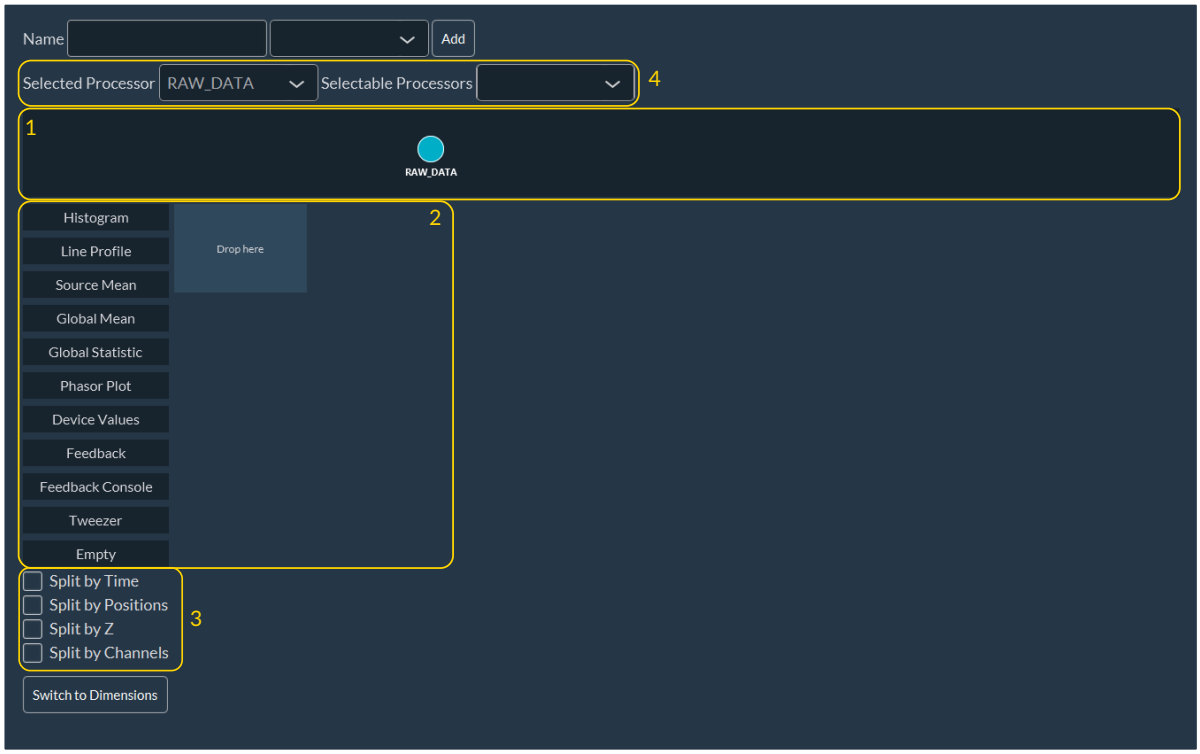
Use the **Data Processor** to customize data visualization and analysis pipelines during and after image acquisition. View raw image streams, monitor fluorescence intensity graphs, or apply post-processing.

Interface Description



In **User Mode**, access pre-configured data processing pipelines. Select a pipeline from the drop-down menu and begin the acquisition.

In **Expert Mode**, click the **Switch to Data Processor** button to build custom data processing workflows.



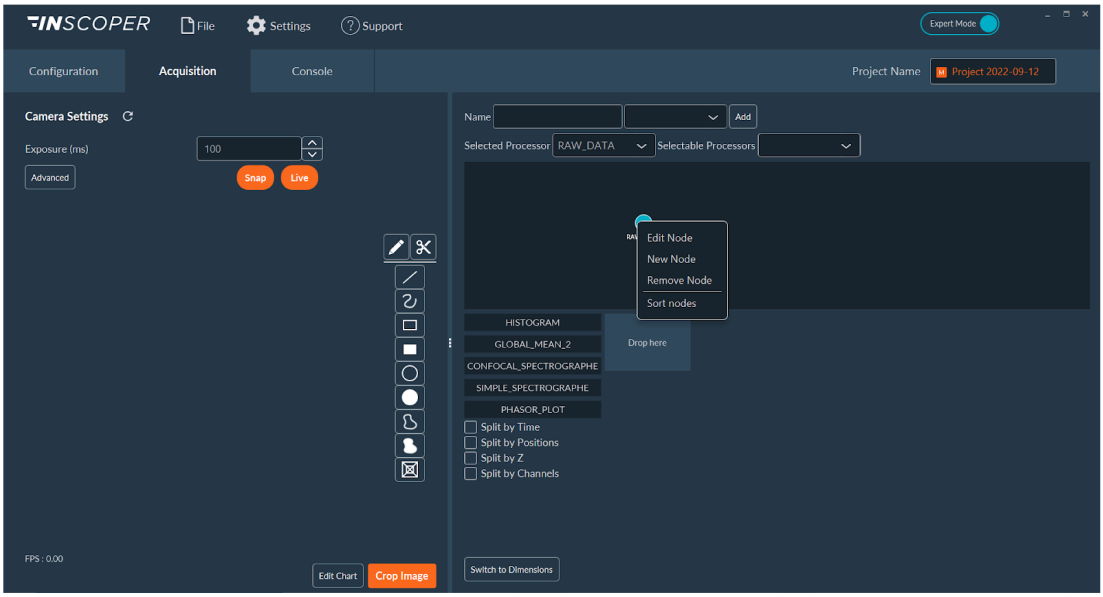
1. **Workspace**: The canvas for creating and routing data processing workflows.
2. **Visualization Tools**: Controls for customizing how data is displayed.
3. **Dimension Analysis Tools**: Tools for analyzing data separated by dimension (e.g., assessing Z-stacks or time-lapses).
4. **Image Visualization Customization**: Controls for viewing images post-processing:
    - **Selected Processor**: Choose which specific processing node's output to view live during acquisition.
    - **Selectable Processor**: Choose which node outputs will remain available for review in the visualization tab after the acquisition concludes. By default, the system selects all processing steps.

> **Note**
>
> The available processing options depend on the system's specific hardware and license configuration.

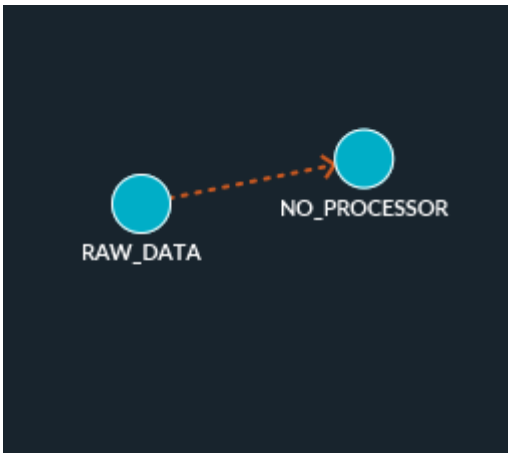
Creating a New Data Processing Workflow

1. Every workflow originates with the raw images, represented by a root node named **RawData**. To process these images, create a new node by right-clicking **RawData** and selecting **New Node**.

All rights reserved.



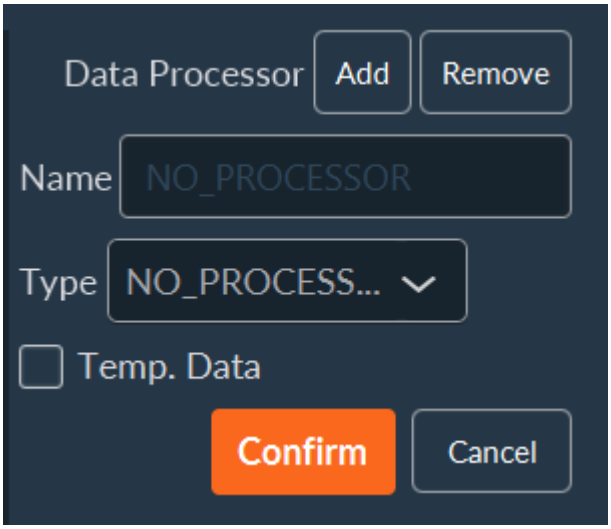
2. The system generates a new, empty node. To assign a processing operation to this node, double-click it, or right-click and select **Edit Node**.



> **Tip**
>
> For improved readability, right-click the workspace and select **Sort Nodes** to automatically arrange the pipeline. You can also manually reposition nodes by dragging and dropping them.

3. In the node properties window, use the **Type** drop-down menu to select the desired processing algorithm (e.g., Stitching, Maximum Projection, Background Subtraction).
    - Check the **Temp. Data** box to prevent saving the output of this specific step to disk. This reduces processing overhead and minimizes storage space requirements.

All rights reserved.



AVAILABLE PROCESSING OPERATIONS

All rights reserved.

Processor	Description
`NO_PROCESSOR`	Passes the data through without modifying it.
`SIMPLE_TILING`	Positions each image at its coordinates within a tiling grid.
`STITCH_TILING`	Stitches adjacent images of a tiling acquisition by blending overlap regions. (Note: Requires `SIMPLE_TILING` to be applied first).
`STANDARD_DEVIATION_ON_FLY`	Calculates pixel intensity standard deviation across an image stack.
`SHADING_CORRECTION`	Corrects uneven illumination and optical artifacts (e.g., sensor dust).
`FILTER`	Reduces noise by removing anomalous pixels (despeckling).
`TIME_MIN`	Performs a minimum intensity projection across the Time dimension.
`TIME_MEDIAN`	Performs a median intensity projection across the Time dimension.
`TIME_MAX`	Performs a maximum intensity projection across the Time dimension.
`TIME_AVERAGE`	Calculates the average pixel intensity across the Time dimension.
`TIME_STANDARD_DEVIATION`	Calculates the intensity standard deviation pixel across the time dimension.
`TIME_SUM`	Calculates the sum pixel intensity across the time dimension. .
`FOCUS_MIN`	Performs a minimum intensity projection across the Z-stack dimension.
`FOCUS_MEDIAN`	Performs a median intensity projection across the Z-stack dimension.
`FOCUS_MAX`	Performs a maximum intensity projection across the Z-stack dimension.
`FOCUS_AVERAGE`	Calculates the average pixel intensity across the Z-stack dimension.
`FOCUS_STANDARD_DEVIATION`	Calculates the intensity standard deviation pixel across the Z-stack dimension.
`FOCUS_SUM`	Calculates the sum pixel intensity across the Z-stack dimension.
`POSITION_MIN`	Performs a minimum intensity projection across the position dimension.
`POSITION_MEDIAN`	Performs a median intensity projection across the position dimension.
`POSITION_MAX`	Performs a maximum intensity projection across the position dimension.
`POSITION_STANDARD_DEVIATION`	Calculates the intensity standard deviation pixel across the position dimension.
`POSITION_SUM`	Calculates the sum pixel intensity across the position dimension.
`CHANNEL_MULTICOLOR`	Merges images from multiple channels into a multi-color composite.
`SUBTRACT_BACKGROUND`	Subtracts the background signal to improve image contrast.
`CHANNEL_RATIO`	Calculates the ratio of intensities between channels to measure relative changes.

All rights reserved.

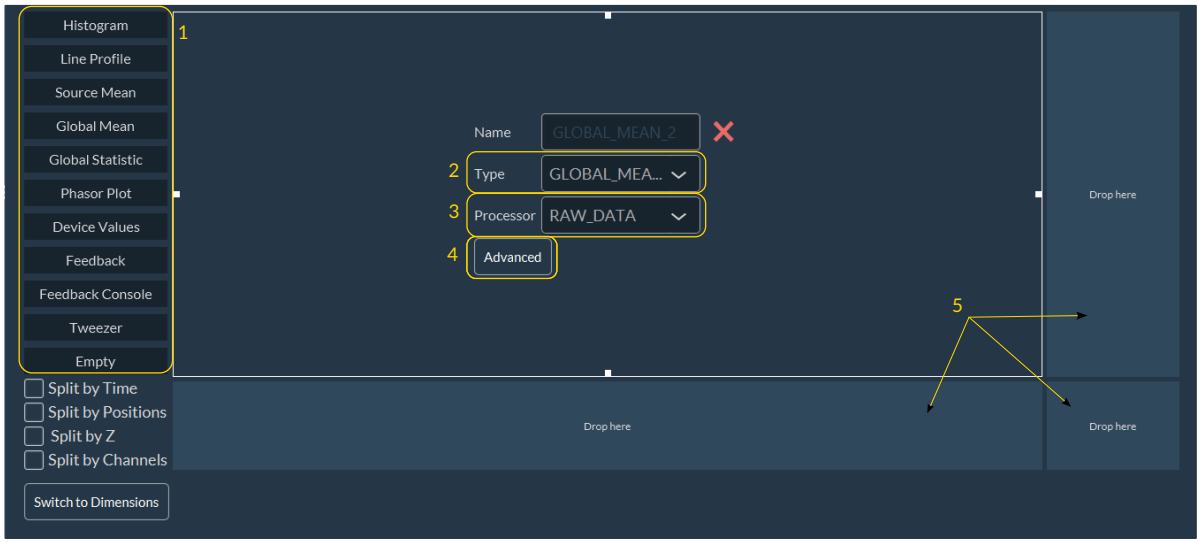
Processor	Description
`MULTI_CHANNEL_MERGE`	Merges channels natively (primarily used in SPIM workflows).

Tip

You can execute all processing steps either during or after acquisition. When run during the acquisition, the processed data updates continuously and saves directly to the final output file.

Customizing Data Visualization

Customize the display of acquired data, such as tracking the evolution of fluorescence intensity over time.



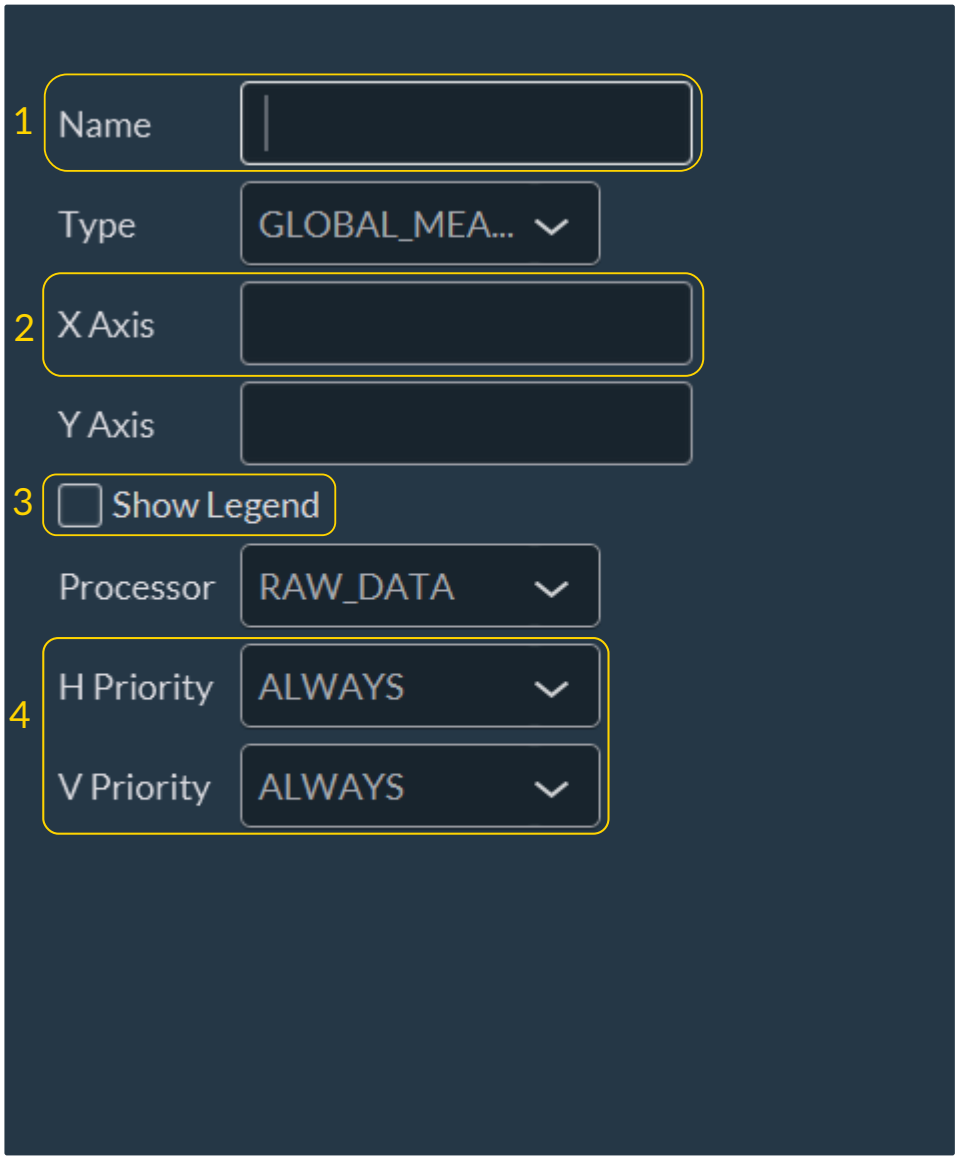
SETTING UP THE VISUALIZATION WINDOW

1. **Select the Data Type**: Choose what metric to display during the acquisition:
    - Histogram: Monitors the evolution of the pixel intensity distribution.
    - Global mean: Monitors the average image intensity across an acquisition sequence.
    - Global statistic: Similar to Global mean, but includes additional statistical metrics (e.g., variance).
    - Device values: Periodically retrieves and plots telemetry from connected hardware devices.
2. **Assign to a Panel**: Drag and drop the selected data type into a view panel. Change this data type later using the panel's drop-down menu.
3. **Specify Source**: Define the data source (node) for the visualization.
4. **Customize Graph**: Adjust visual graph options, such as axis titles and curve labels.
5. **Add Graphs**: To display multiple graphs simultaneously, drag and drop additional data types into empty view panels.

GRAPH CUSTOMIZATION OPTIONS

1. Set a custom title for the graph.
2. Label the X and Y axes.
3. Toggle the visibility of curve legends.
4. Enable the **Always** option to ensure the graph remains visible throughout the experiment.

All rights reserved.



Note

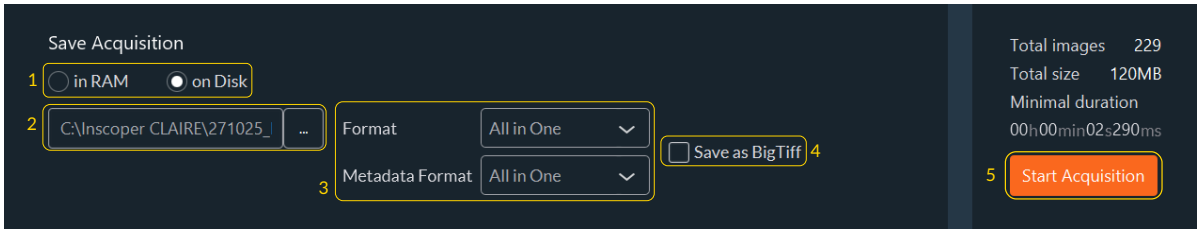
Visualization customization is optional. It aids in data interpretation during acquisition and does not affect the saved acquisition data.

2026-09-08

All rights reserved.

1.4.12 Saving Images and Metadata

The saving interface configures the destination path and data architecture for the automated sequence outputs.



Configure how the acquired images and metadata are handled:

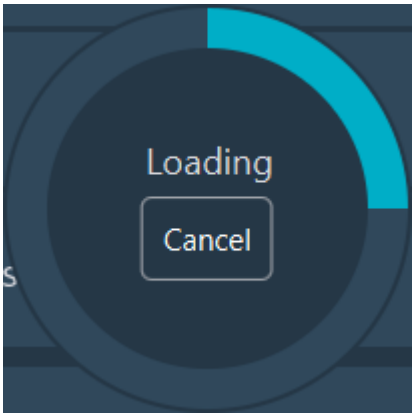
1. Select the data storage destination:
    - **RAM**: (Not recommended for prolonged acquisitions). Data stored in RAM is volatile and is lost if the application closes or the system crashes. Use only for temporary test captures.
    - **Hard disk, SSD, or Server**: (Recommended). Writes data to a physical drive or network storage.
2. Specify the destination folder path.
3. Choose the image saving format:
    - **All in one file**: Combines the dimensional dataset (Z-stacks, Time points, Channels) into a single, multi-page image file.
    - **One TIFF per image**: Saves every captured frame as a discrete TIFF file (e.g., `Image_T01_Z03_C01.tiff`). Use this if post-processing tools require sequential files instead of multi-page containers.
4. (Optional) Check **.bigTiff** to support files exceeding the standard 4 GB legacy TIFF limit (recommended for large volumetric datasets).

> **Image compression**
>
> TIFF files are uncompressed by default. To enable lossless ZLib compression, set **Tiff Compression** in Display Settings > Preferences.

> **Tip**
>
> To configure standardized file naming conventions before initiating an automated acquisition, refer to the Display Settings guide.

1. Start acquisition

The sequence will be generated and uploaded to the Inscoper device controller. You can abort the process at any time by clicking **Cancel** on the loading progress indicator.



2026-10-05

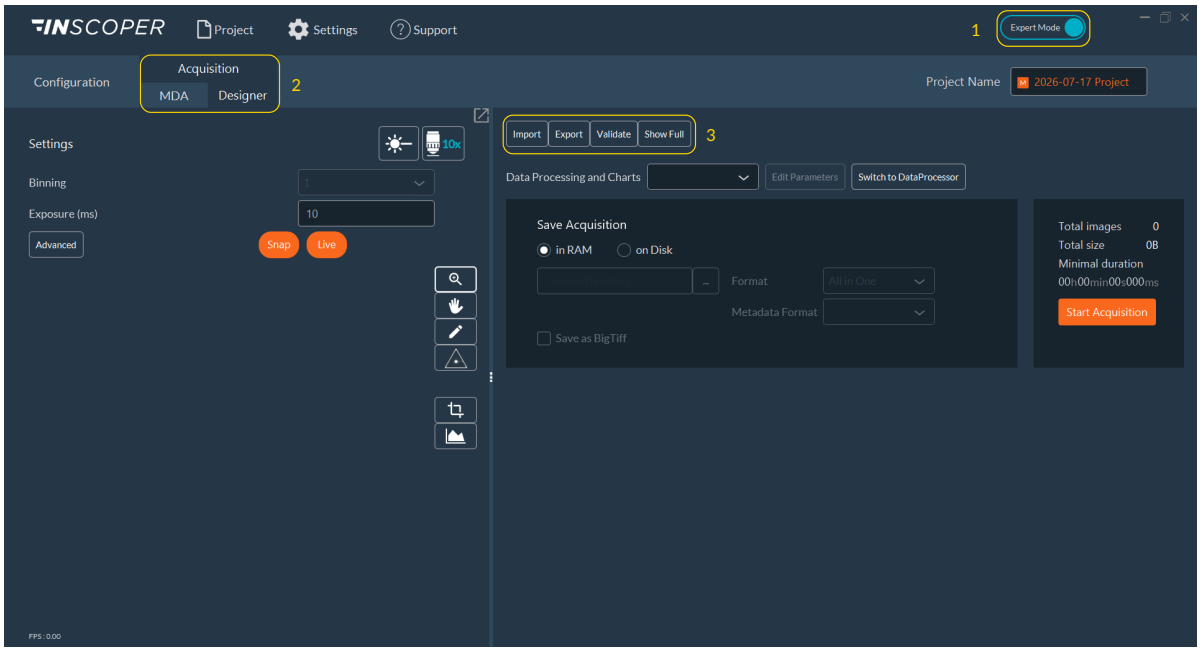
All rights reserved.

1.4.13 Acquisition Designer

The **Designer** module is a graphical workspace for building custom acquisition protocols. Combine logical and hardware blocks into an interactive flowchart to define a tailored, multi-dimensional workflow.

How to Access the Designer

Enable **Expert Mode** (1), open the main **Acquisition** tab, then select the **Designer** sub-tab (2).



> **Note**
>
> **Validate** and **Show Full** are only available in Expert Mode (1). In User Mode, the action bar (3) is reduced to **Import** and **Export**.

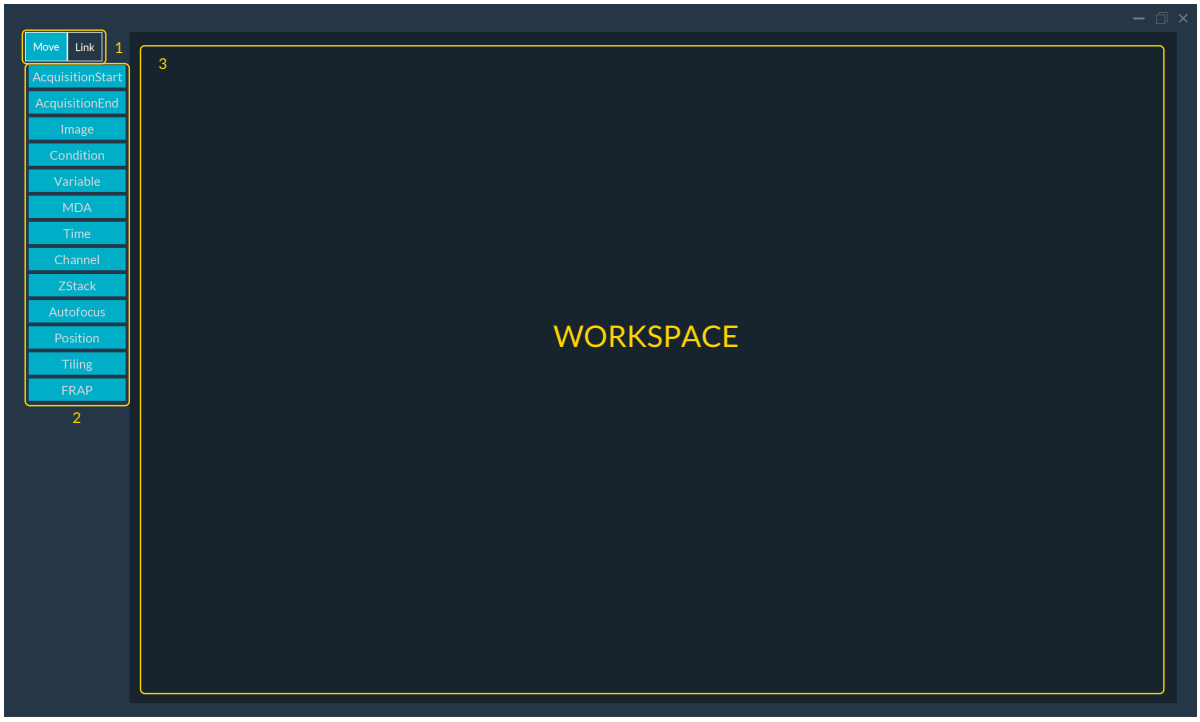
The action bar (3) at the top right of the tab provides four commands:

- **Import:** load an existing protocol from a JSON file. The file browser opens in the `complexProtocols` directory. A confirmation pop-up appears once the protocol is loaded.
- **Export:** save the current protocol as a JSON file, by default in the same `complexProtocols` directory.
- **Validate:** register the protocol modifications and recompute the total number of images.
- **Show Full:** open the complete graphical workspace used to build the flowchart.

How to Create a Protocol

Click **Show Full** to open the workspace. The workspace opens as a separate maximized window, on the secondary screen when one is available. Assemble the protocol from blocks placed and linked on the canvas.

All rights reserved.



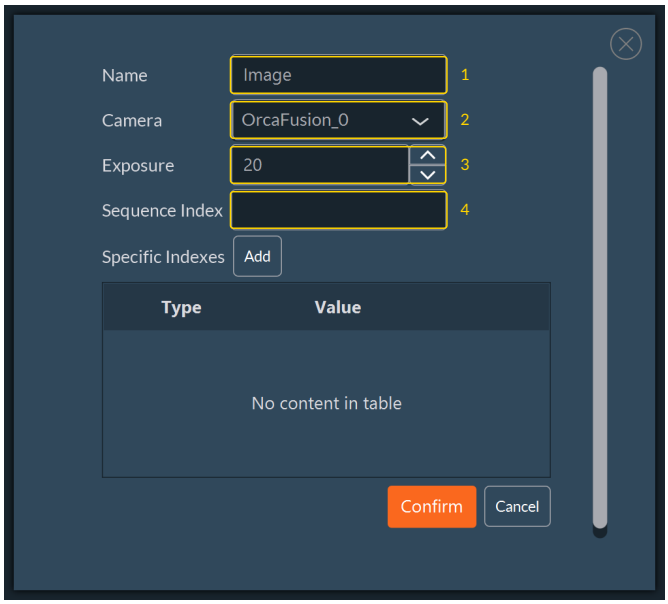
The window is organized in three areas: the tool selector (1), which switches between **Move** and **Link**; the block palette (2), which lists every block type available; and the canvas (3), which holds the flowchart.

HOW TO INTERACT WITH BLOCKS

- **Add a block:** drag a block from the palette (2) onto the canvas (3) and release the mouse button at the target location.
- **Move a block:** activate the **Move** tool (1), then drag the block across the canvas.
- **Connect two blocks:** select the **Link** tool (1), click the source block, then click the target block. The Designer draws a directional flow arrow between them.
- **Configure a block:** switch back to **Move** mode (1), then double-click the block to edit its parameters.

LIST OF AVAILABLE BLOCKS

- **AcquisitionStart**: mandatory marker defining the protocol start point.
- **AcquisitionEnd**: mandatory marker defining the protocol end point.
- **Image**: captures raw data. Set the block **Name** (1), the **Camera** (2), the **Exposure** time in milliseconds (3), and the **Sequence Index** (4).



- **Condition**: evaluates a logical expression and splits the flow into two branches.

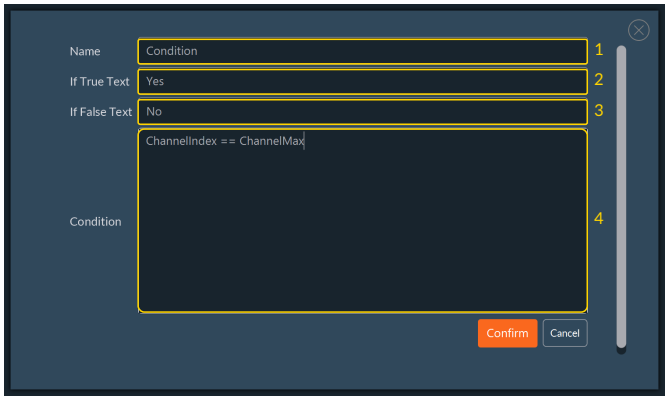
All rights reserved.



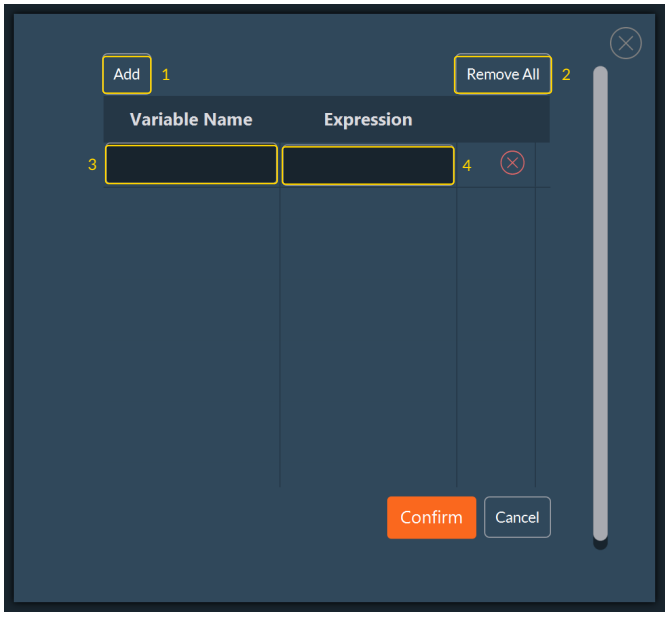
Four parameters are available: **Name** (1), **If True Text** (2), **If False Text** (3), and **Condition** (4). The two text fields (2) and (3) set the labels drawn on the outgoing branches; they default to `Yes` and `No` and may be renamed.

- **No** (loop back): if the expression is false, the flow returns to a designated block and repeats the step.
- **Yes** (loop exit): if the expression is true, the flow leaves the loop and advances to the next step.

A numeric expression is accepted in the **Condition** field (4) in place of a boolean one: any non-zero result is treated as true. An expression left empty or set to `TRUE` always follows the true branch. See Condition and Variable Syntax.

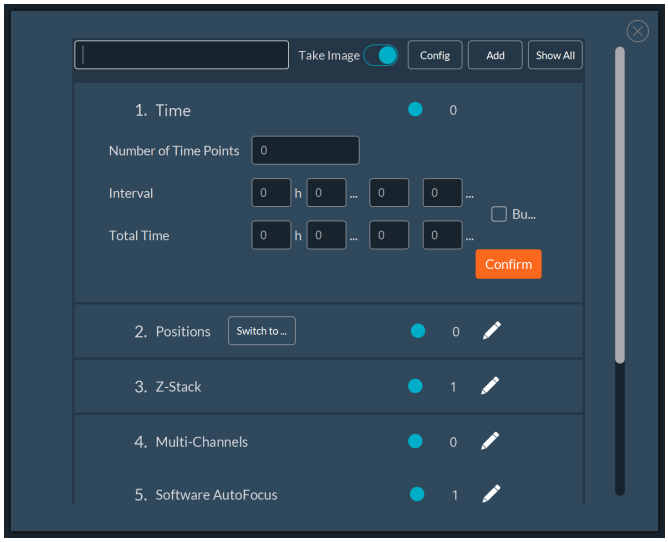


- **Variable**: declares one or more named values, each defined by a **Variable Name** (3) and an **Expression** (4). Use **Add** (1) to append a declaration and **Remove All** (2) to clear the list. Every expression is re-evaluated each time the flow passes through the block, so a variable may reference dimension indices and previously declared variables, not only constants.
  - **Function**: stores or modifies a value used by a secondary dimension of a complex acquisition.
  - **Prerequisite**: place the block before the first block that references the variable.

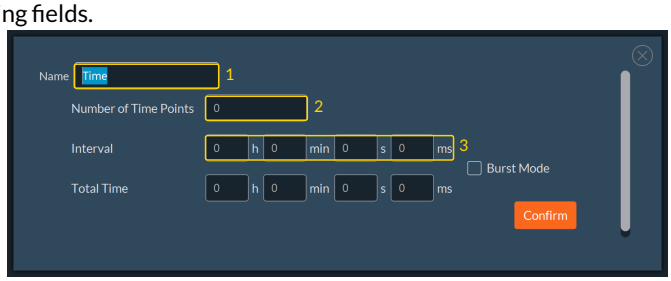


- **MDA**: encapsulates a complete multidimensional sequence in a single block. Its dimensions are configured in the numbered list of the block window: **Time**, **Positions**, **Z-Stack**, **Multi-Channels**, and **Software AutoFocus**. Use it when a standard routine is sufficient inside a larger custom protocol, instead of building the equivalent chain of blocks manually.

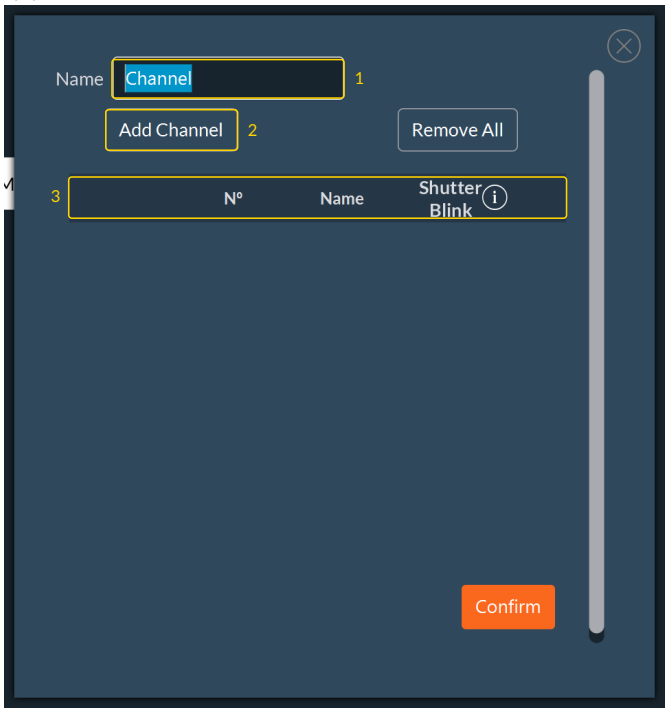
All rights reserved.



- **Time**: defines a time-lapse dimension from the block **Name** (1), the **Number of Time Points** (2), and the **Interval** (3). **Total Time** is derived from the two preceding fields.

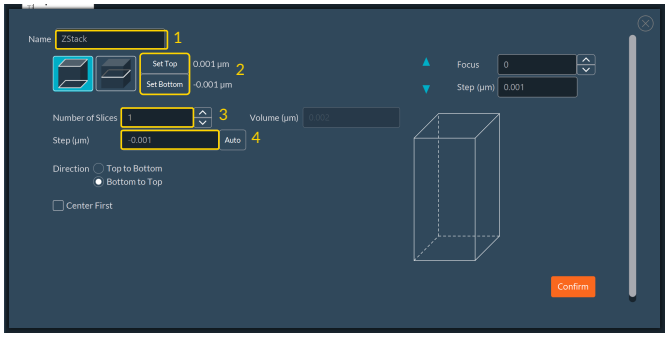


- **Channel**: defines one or several illumination wavelengths. Set the block **Name** (1), then use **Add Channel** (2) to append a wavelength to the channel table (3).

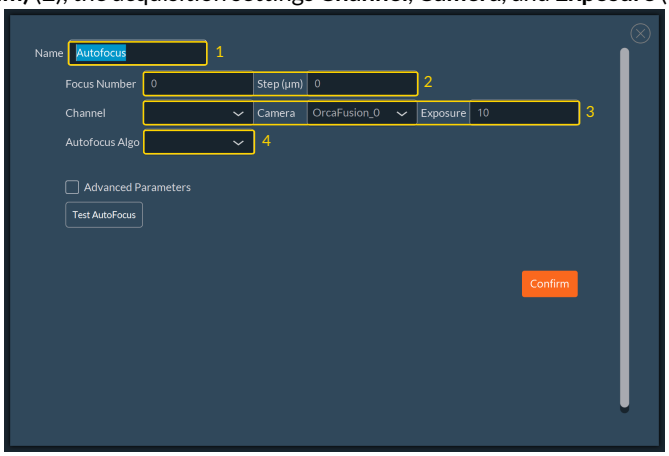


- **ZStack**: defines acquisition along the Z axis from the block **Name** (1), the top and bottom points set with **Set Top** and **Set Bottom** (2), the **Number of Slices** (3), and the **Step (µm)** (4).

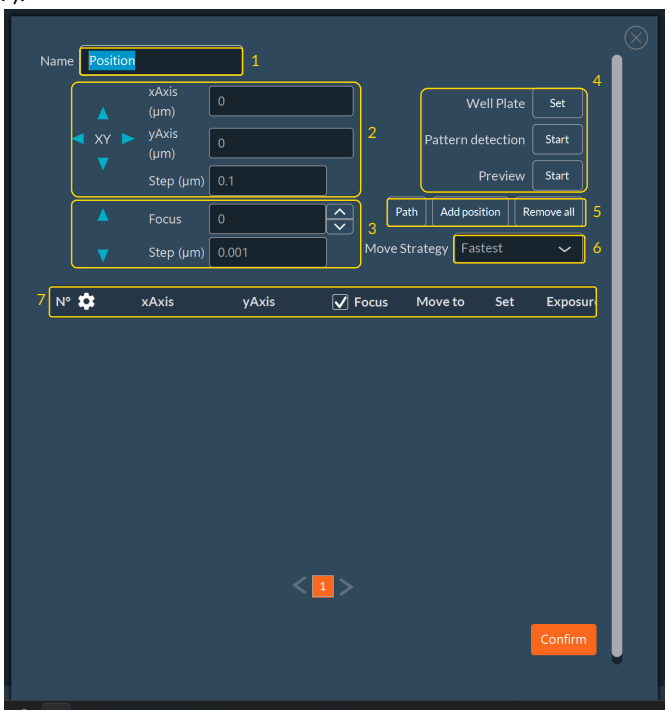
All rights reserved.



- **Autofocus**: finds the optimal focus plane. Set the block **Name** (1), the number of sampling images and the travel between them with **Focus Number** and **Step (µm)** (2), the acquisition settings **Channel**, **Camera**, and **Exposure** (3), and the **Autofocus Algo** (4).

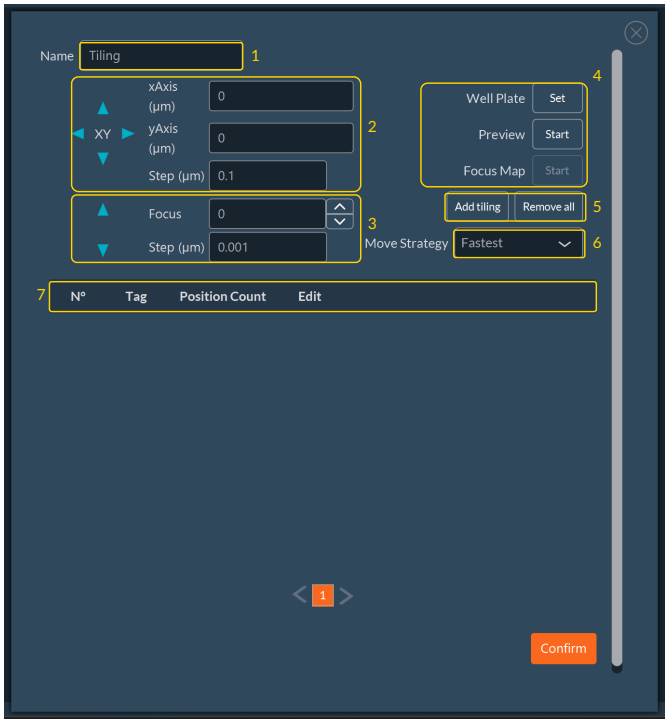


- **Position**: defines the XY stage coordinates visited by the protocol. Set the block **Name** (1), the **XY** coordinates and their step (2), and the **Focus** value and its step (3). The right column groups the **Well Plate** definition, the **Pattern detection**, and the **Preview** (4), the position commands **Path**, **Add position**, and **Remove all** (5), and the **Move Strategy** (6). Every position registered is listed in the table (7).

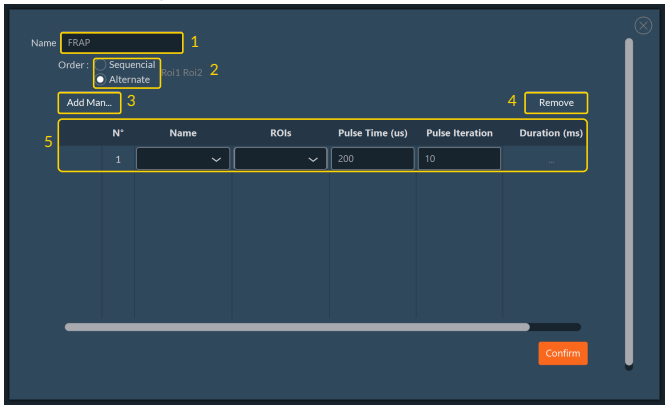


- **Tiling**: generates an image mosaic from adjacent fields of view. Set the block **Name** (1), the **XY** coordinates and their step (2), and the **Focus** value and its step (3). The right column groups the **Well Plate** definition, the **Preview**, and the **Focus Map** (4), the mosaic commands **Add tiling** and **Remove all** (5), and the **Move Strategy** (6). Every mosaic registered is listed in the table (7).

All rights reserved.



- **FRAP**: optional module block for photobleaching and photoactivation experiments. Set the block **Name** (1) and the **Order** in which the regions are illuminated, **Sequencial** or **Alternate** (2). Use the manual entry button (3) to append a pulse and **Remove** (4) to delete the selected one. Each pulse is listed in the table (5) with its **Name**, **ROIs**, **Pulse Time (us)**, and **Pulse Iteration**. Configure the block according to the laser setup.



Important

- **Mandatory unique naming**: every block in the workspace must have a unique name. Each dimension block publishes its counters under its own name, so two blocks sharing a name overwrite each other and **Condition** blocks evaluate the wrong values.
- **Structural errors are flagged, logical errors are not**: the workspace marks a block whose links are incomplete, for example a **Condition** block with an unconnected branch. It does not check the content of an expression. A condition that references a misspelled or undeclared name is only detected when the protocol runs.
- **Silent failure behavior**: the Designer does not display pop-up warnings for logical errors. If clicking **Start Acquisition** produces no action, inspect the links, the block names, and the loop criteria.

CONDITION AND VARIABLE SYNTAX

Condition and **Variable** blocks share the same expression language. Expressions are evaluated against the acquisition context, which holds the counters published by every dimension block and the variables already declared. A name that is not present in the context at the moment of evaluation aborts the acquisition with an error, so a **Variable** block must always be placed upstream of the first block that references it.

All rights reserved.

Available Names

Each dimension block publishes two names built from the block name:

Name	Meaning
`[Block Name]Index`	Number of steps already completed in that dimension. It is `0` before the first step and reaches `[Block Name]Max` once the last step is done.
`[Block Name]Max`	Total number of steps in that dimension: the number of slices, time points, positions, or channels declared in the block.

A **Tiling** block publishes three additional names:

Name	Meaning
`[Block Name]TilingIndex`	Index of the current mosaic.
`[Block Name]TilingNumber`	Total number of mosaics.
`[Block Name]TilingLineIndex`	Index of the current line within the mosaic.

A **Variable** block publishes each declared variable under its own name. The value of a variable is itself an expression, re-evaluated every time the flow passes through the block.

Important

- Names are case-sensitive and must match the block name exactly. A name must start with a letter or an underscore and may contain only letters, digits, and underscores. Dotted names such as `ZStack1.Index` are not valid.
- An **MDA** block does not publish counters. Its internal Time, Channel, and ZStack dimensions cannot be referenced from a **Condition** expression. Use separate dimension blocks when the protocol has to branch on their indexes.

Operators and Functions

Category	Symbols
Arithmetic	`+`, `-`, `*`, `/`, `%` (modulo), `^` (power)
Comparison	`==`, `!=`, `>`, `>=`, `<`, `<=`
Boolean	`&&` (and), `||` (or), `!` (not)
Boolean constants	`TRUE`, `FALSE` (uppercase)
Grouping	`( )`

All rights reserved.

The following functions are available. Angles are expressed in radians.

Function	Description
`sqrt(x)` , `exp(x)` , `ln(x)`	Square root, exponential, natural logarithm
`abs(x)` , `floor(x)` , `ceil(x)`	Absolute value, rounding down, rounding up
`cos(x)` , `sin(x)` , `tan(x)`	Trigonometric functions
`acos(x)` , `asin(x)` , `atan(x)`	Inverse trigonometric functions
`atan(y, x)`	Two-argument arc tangent
`deg(x)` , `rad(x)`	Conversion between radians and degrees
`if(condition, a, b)`	Returns `a` when the condition is true, otherwise `b`

Operator Precedence

Operators are grouped in the following order, from the highest precedence to the lowest. Operators of equal precedence are evaluated from left to right.

Rank	Operators
1	`!` , unary `-`
2	`^`
3	`*` , `/`
4	`+` , `-`
5	`%`
6	`==` , `!=` , `>` , `>=` , `<` , `<=`
7	`&&` , `||`

Warning

Modulo has a lower precedence than addition and subtraction, which is the opposite of the convention used in most programming languages. `TimeIndex - 1 % 2` is evaluated as `(TimeIndex - 1) % 2` , not as `TimeIndex - (1 % 2)` . Use explicit parentheses whenever `%` appears next to `+` or `-` .

Typical loop and interval expressions:

```
[Block Name]Index == [Block Name]Max
[Block Name]Index % [Variable Name] == 0
```

In the **Variable** block below, each variable name (1) is paired with the expression that defines it (2). The names `laser1` , `laser2` , and `laser3` are then available to every downstream **Condition** block.

All rights reserved.



Note

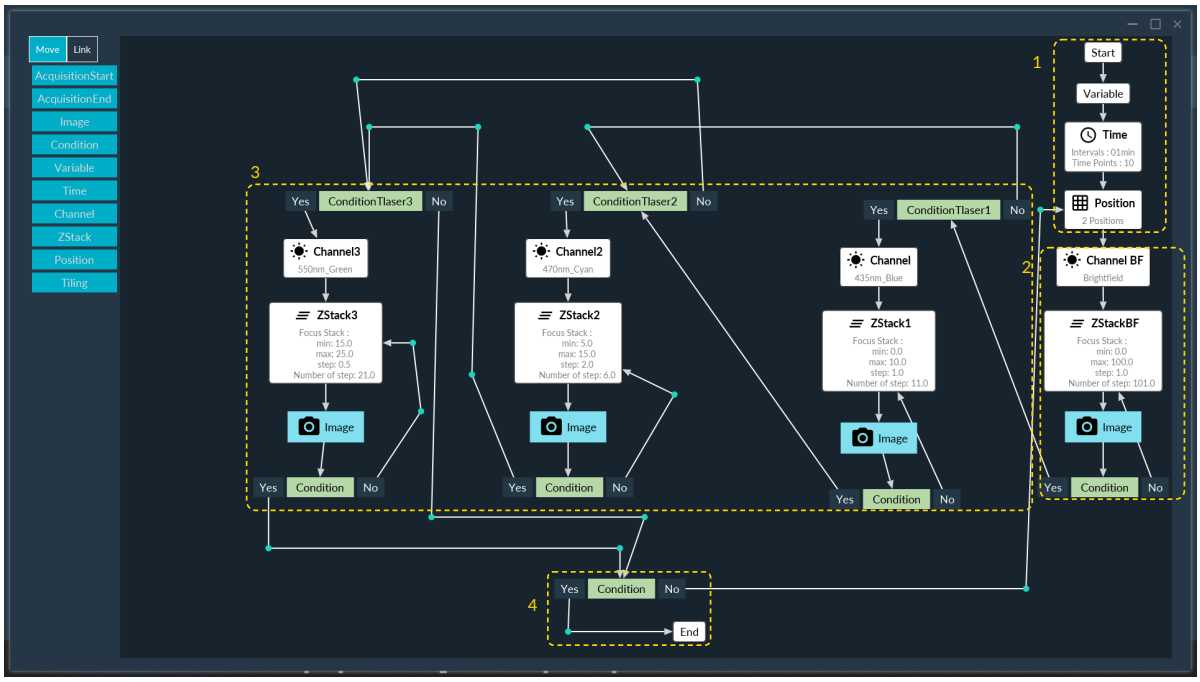
Condition examples:

- `ZStack1Index == ZStack1Max` : every Z slice declared in the **ZStack1** block has been acquired. Use it to close a Z-stack loop.
- `PositionIndex == PositionMax` : every position declared in the **Position** block has been visited. Use it to close a position loop.
- `TimeIndex % laser2 == 0` : the current time point is a multiple of the interval stored in the variable `laser2` . Use it to acquire a channel on a subset of the time points.
- A multi-channel loop (e.g., 405, 488, and 638) closes on the same rule: `[Block Name]Index == [Block Name]Max` .

COMPLEX ACQUISITION EXAMPLES

Example 1: One Interval Time per Channel

Time-lapse, multi-position, and Z-stack acquisition with four channels — BF, DAPI, GFP, and Texas Red — using a different interval time for each channel.



Description: the numbers below match the regions outlined on the flowchart.

1. Initial setup (1)
    - Start the acquisition.
    - Set the variables: declare the interval times used by the time-lapse.

All rights reserved.

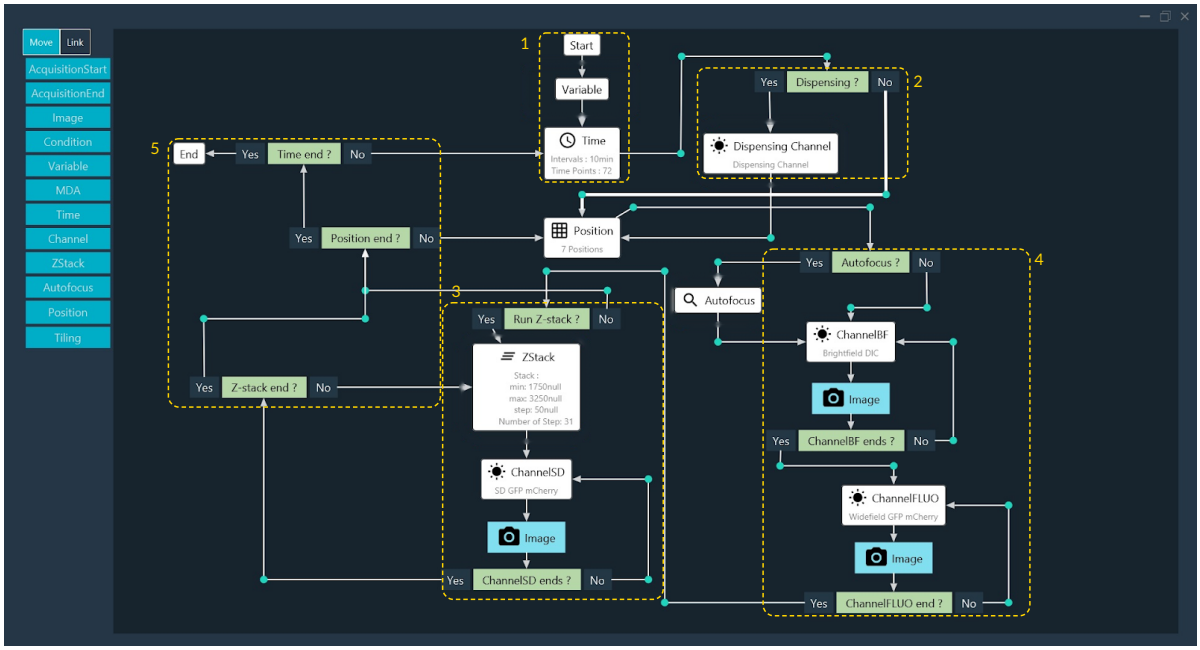
- Set up the time-lapse: configure the timing parameters.
- Choose the positions: define the XY coordinates.

2. Default sequence: brightfield (BF) (2)
    - Select the first channel (BF).
    - Set up the Z-stack.
    - Acquire the image.
    - Condition: is the Z-stack finished?
        - **No**: continue the Z-stack acquisition.
        - **Yes**: proceed to the channel changes.
3. Channel conditional sequence (3)
    - Laser 1 condition (DAPI): is the current time point a valid interval for DAPI acquisition?
        - **Yes**: select the DAPI channel, run the Z-stack, and acquire the image.
        - **Sub-condition**: is the Z-stack finished? If no, continue the acquisition. If yes, proceed to the next channel.
        - **No**: skip to the second channel.
    - Laser 2 condition (GFP): is the current time point a valid interval for GFP acquisition?
        - **Yes**: select the GFP channel, run the Z-stack, and acquire the image.
        - **Sub-condition**: is the Z-stack finished? If no, continue the acquisition. If yes, proceed to the next channel.
        - **No**: skip to the third channel.
    - Laser 3 condition (Texas Red): is the current time point a valid interval for Texas Red acquisition?
        - **Yes**: select the Texas Red channel, run the Z-stack, and acquire the image.
        - **Sub-condition**: is the Z-stack finished? If no, continue the acquisition. If yes, proceed to the position check.
        - **No**: skip to the position check.
4. Position loop and end (4)
    - Condition: are all positions acquired?
        - **No**: move to the next position, then restart the sequence from step 2.
        - **Yes**: end the acquisition.

Example 2

The following protocol combines the same building blocks in a longer branched architecture. It alternates a spinning-disk Z-stack with a widefield channel pair, at seven positions and over 72 time points.

All rights reserved.



Description:

1. Initial setup (1): start the acquisition, declare the variables, and configure the time-lapse.
2. Dispensing branch (2): a **Condition** block tests whether a reagent has to be dispensed at the current time point. If dispensing is required, the dispensing channel is selected before the flow reaches the **Position** block.
3. Spinning-disk sequence (3): a **Condition** block tests whether the Z-stack has to run. If it does, the **ZStack** block, the **ChannelSD** block, and the **Image** block are executed until the Z-stack loop closes.
4. Widefield sequence (4): an **Autofocus** condition runs the autofocus when required, then the **ChannelBF** and **ChannelFLUO** blocks are acquired in turn, each closing on its own channel condition.
5. Loop exits (5): the Z-stack, position, and time conditions are evaluated in that order. The protocol ends once the last time point has been acquired.

Finalizing Acquisition Settings

REVIEW AND VALIDATE THE ACQUISITION PROTOCOL

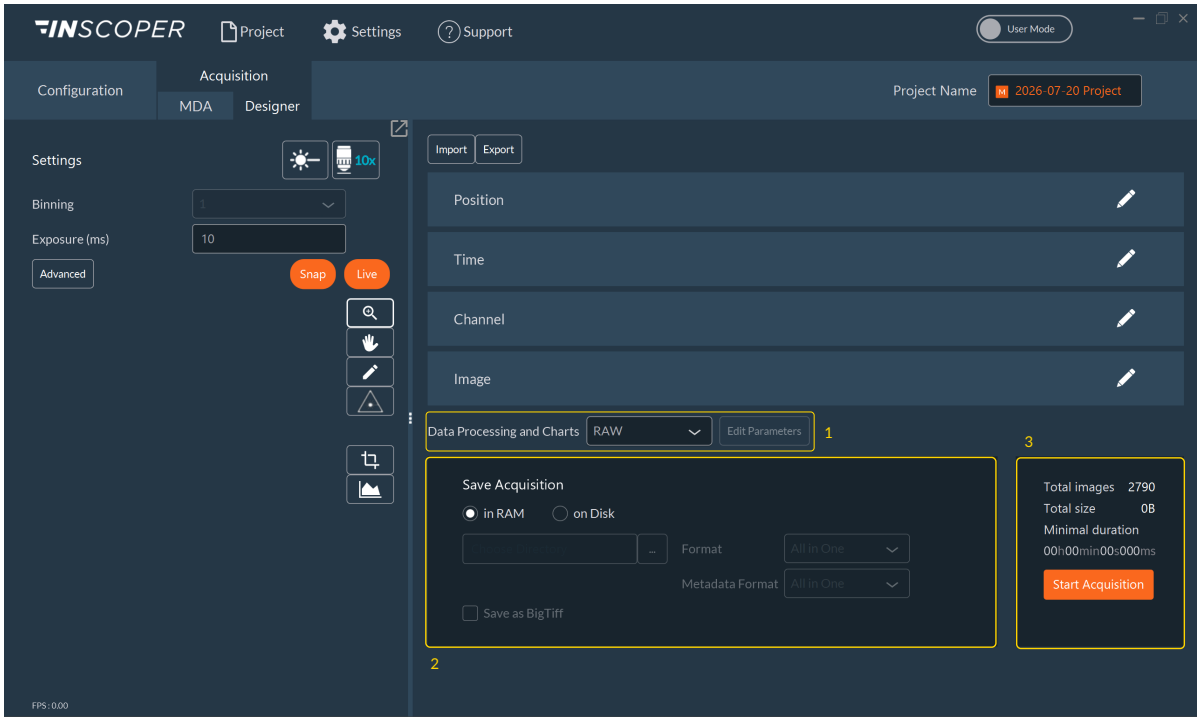
Once the protocol is built, close the **Show Full** window to return to the main **Designer** tab for an overview and final adjustments.

Click **Validate** to register and verify the design.

> **Note**
>
> If the total image counter remains at **0** after clicking **Validate**, the flow architecture contains an error. Re-evaluate the block logic and the links.

All rights reserved.

DATA PROCESSING AND SAVING



Configure the following before launching the acquisition:

- The **Data Processing and Charts** selection (1), which controls real-time image visualization during and after the acquisition. Click **Edit Parameters** to configure the selected DataProcessor.
- The **Save Acquisition** settings (2): the saving location, **in RAM** or **on Disk**, and, for a saving on disk, the target directory, the **Format**, and the **Metadata Format**.

> **Note**
>
> The summary panel (3) displays the total number of images, the estimated file size, and the minimum duration of the experiment in real time.

Click **Start Acquisition** (3) to run the experiment.

2026-09-08

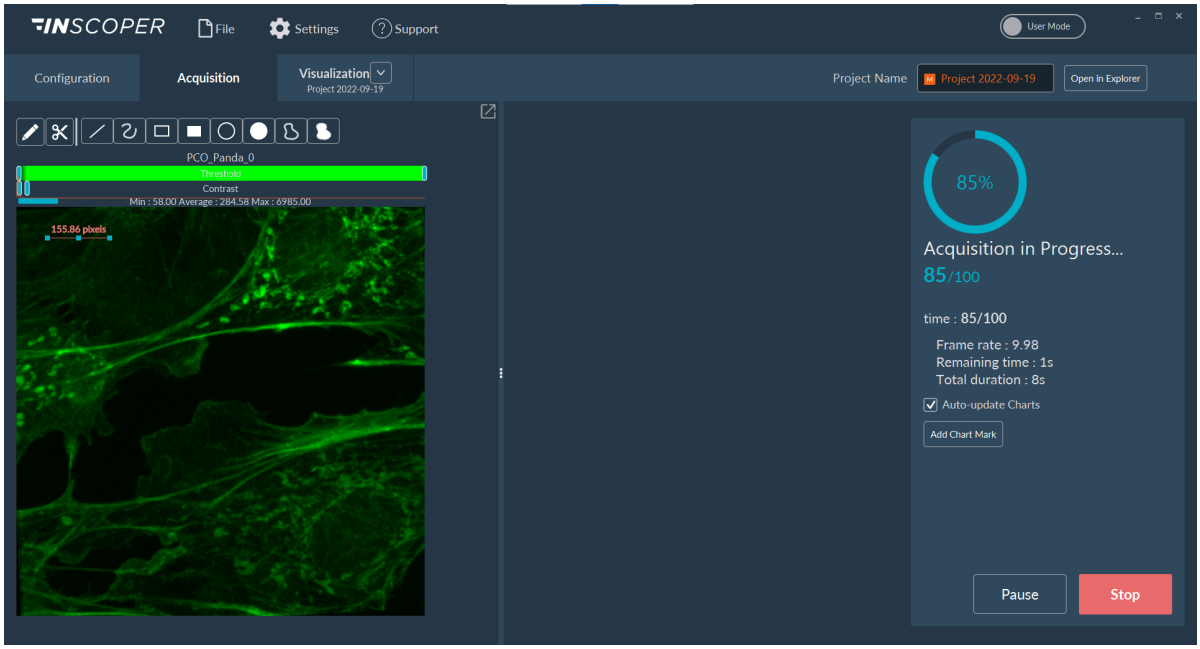
All rights reserved.

1.5 Visualization & Analysis

Inscoper I.S. provides tools for visualizing data during and after acquisition.

1.5.1 Visualization During Acquisition

The **Visualization** tab allows for real-time monitoring of the acquisition sequence.



The interface is divided into three functional areas:

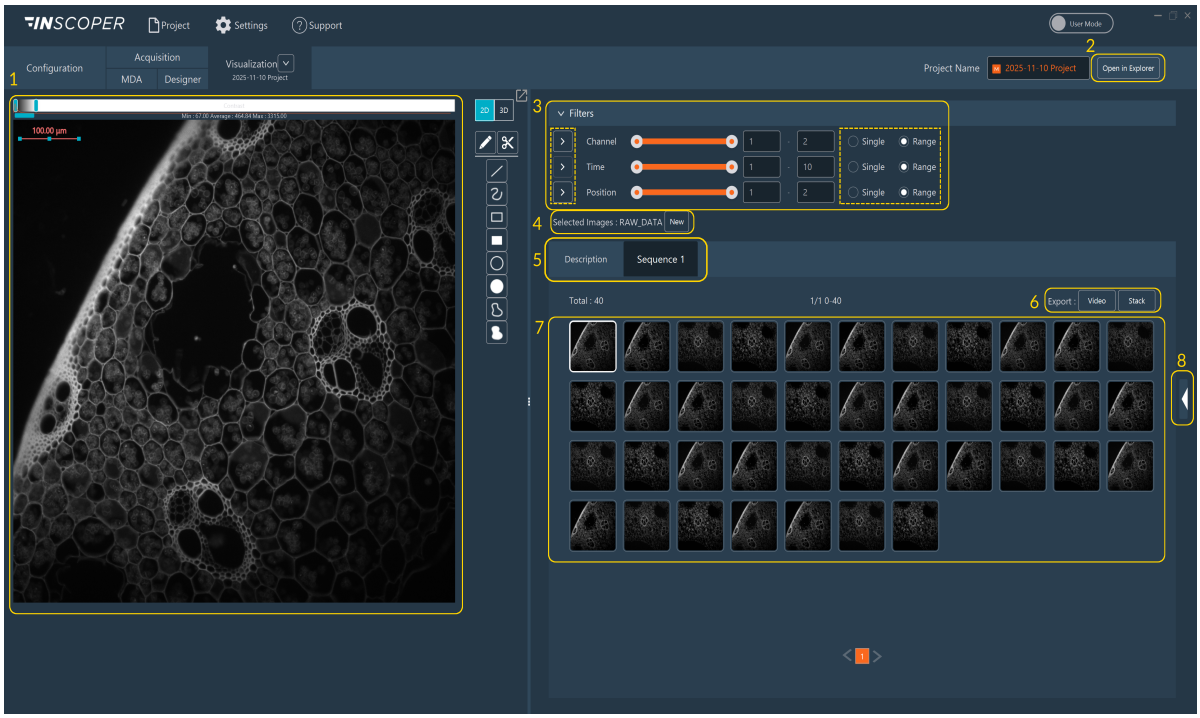
1. **Left Panel:** Displays the live, incoming images from the current acquisition sequence.
2. **Center Panel:** Renders corresponding graphical data (e.g., intensity evolution or statistical plots), depending on the configured Data Processor pipeline.
3. **Right Panel:** Provides acquisition progress metrics and system controls to pause or abort the active sequence.

Large Area Tiling

During extensive tiling acquisitions, the left panel dynamically updates to display the full macroscopic mosaic as individual tiles are acquired and aggregated into the global view.

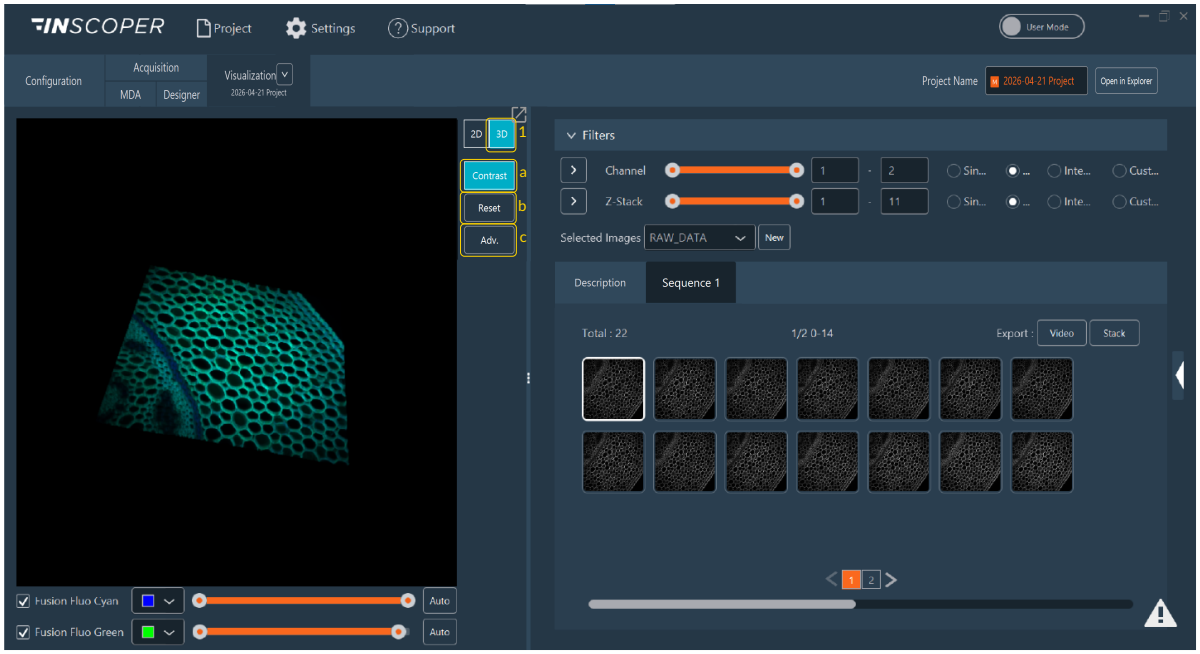
All rights reserved.

1.5.2 Visualization After Acquisition



1. **Image Canvas**: View images in the central viewer.
2. **Open Directory**: Navigate directly to the local dataset storage path (applicable only if data is saved to disk).
3. **Filters**: Use dimension-based filters to search your dataset. Select individual images or full groups. Click the **Play** button adjacent to a dimension name to animate the sequence.
4. **Data Processing**: Select output images for post-acquisition processing.
5. **Mode**: Switch viewing modes to visualize quantitative data charts and Description of your acquisition.
6. **Export Options**: Save the currently viewed sequence as an MP4 **Video** or a TIFF **Stack**.
7. **Thumbnail Gallery**: Browse miniature representations of all acquired images.
8. **Metadata Access**: Reveal the hardware and experimental parameters tagged to the images.

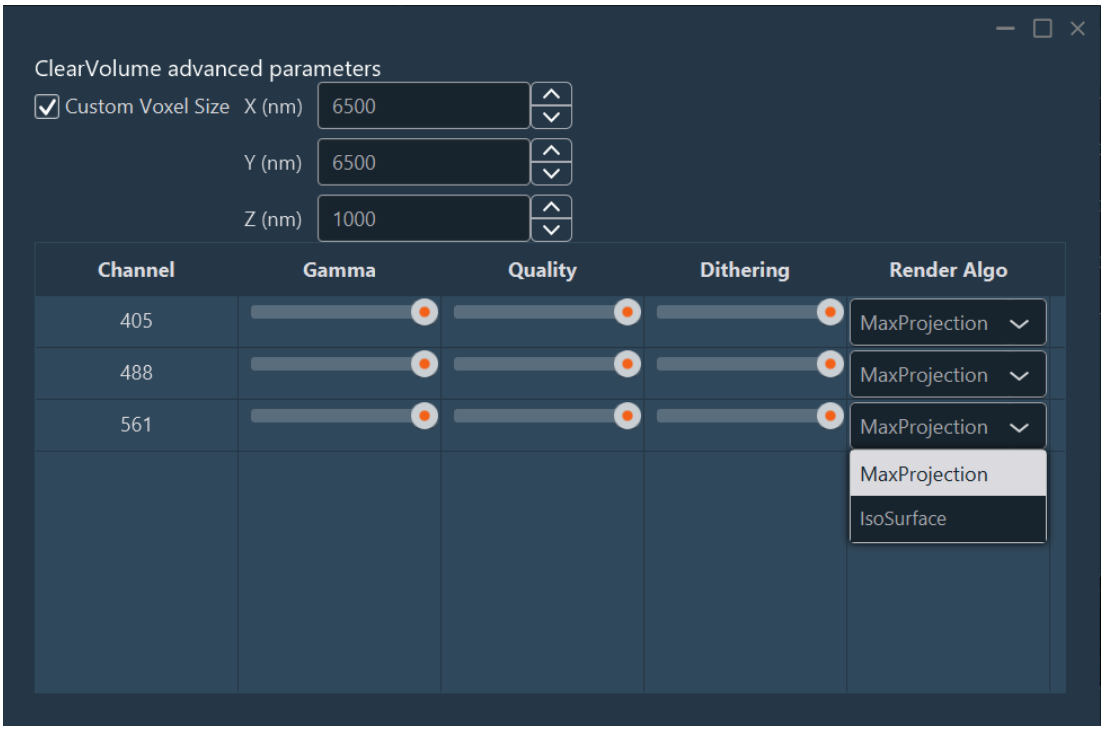
3D Viewer



1. Click the **3D** button to launch the volumetric viewer.
2. 3D visualization canvas.

All rights reserved.

3. Adjust rendering options and parameters:
    1. **Contrast**: Display the LUT and adjust the contrast bar below the 3D view.
    2. **Reset**: Restore the default 3D camera view.
    3. **Advanced**: Access detailed rendering parameters per channel:
        - **Voxel Size** (nm).
        - **Gamma**: Apply a non-linear gamma curve to adjust contrast.
        - **Quality**: Adjust rendering quality for navigation.
        - **Dithering**: Reduce banding/aliasing artifacts.
        - **Render Algorithms**:
            - **MaxProjection**: Displays only the voxels with the maximum intensity encountered along the viewing ray.
            - **IsoSurface**: Renders a 3D surface connecting points of equal intensity value.



Limitations

The 3D viewer accepts raw data only, and renders one position at a time:

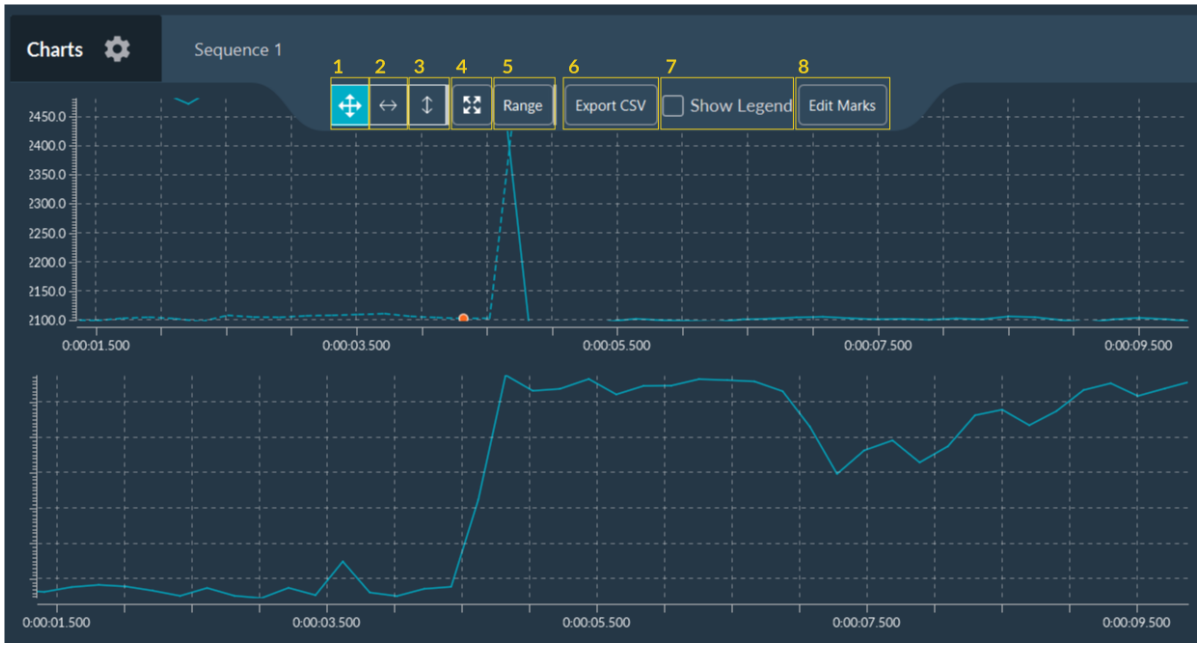
- Use the dimension filters to select the position to render.
- Processed outputs are not supported. Selecting a channel overlay (`CHANNEL_MULTICOLOR`) or a tiling (`SIMPLE_TILING` , `STITCH_TILING`) opens the viewer on an empty black volume, without an error message.

1.5.3 Interacting with Graphics

Interact with charts by adjusting their appearance, adding time markers, or exporting raw data. Hover over a graph to access its toolbars.

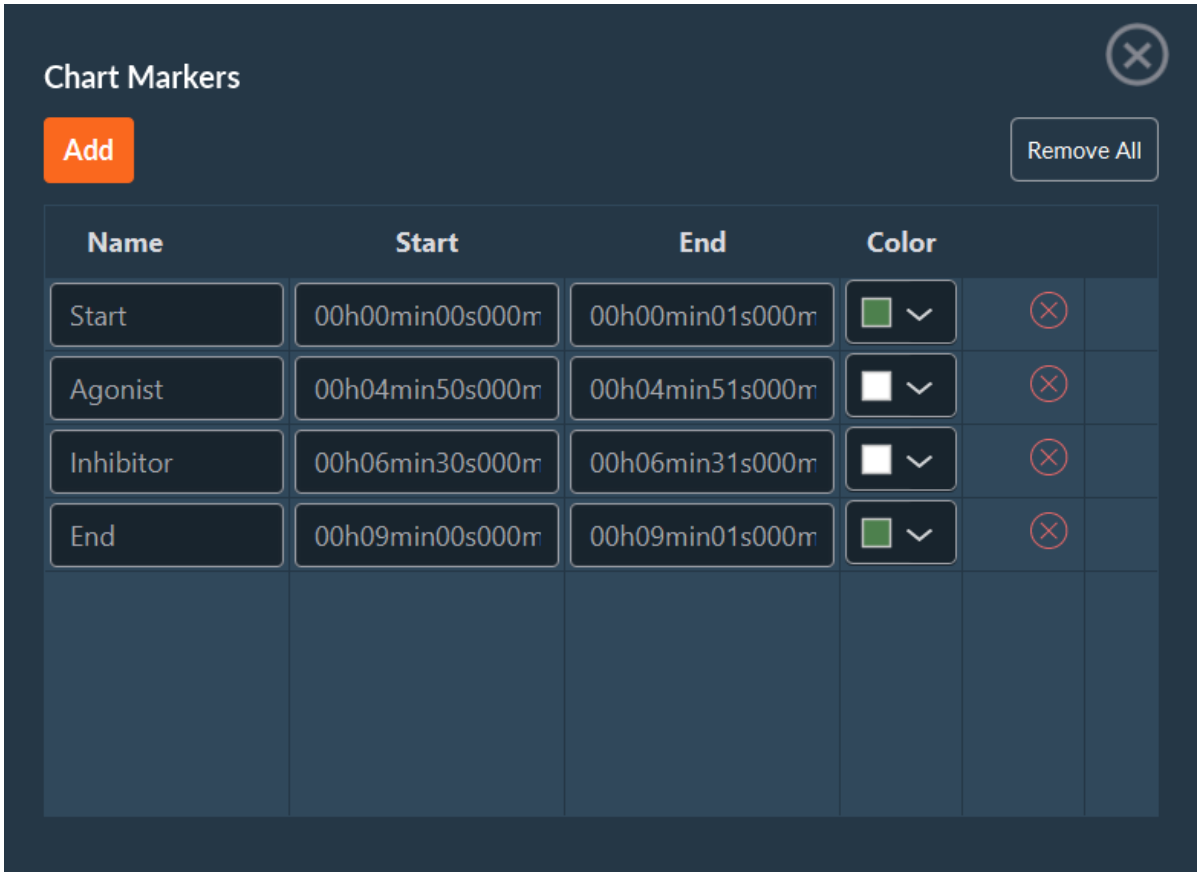
- **Pan**: Click and drag the mouse wheel (middle click) to pan across the graph.
- **Zoom**: Scroll the mouse wheel to zoom in and out. Click and drag to select a region to magnify.
- **Jump to Image**: Left-click a data point on the graph to view the corresponding image.
- **Reset View**: Right-click within the graph to restore the default axis scaling.

All rights reserved.



Graph Tools

1. Enable the XY zoom mode
2. Enable the X zoom mode
3. Enable the Y zoom mode
4. Reset zoom or enable auto-ranging.
5. Manually override the axis ranges.
6. Export the raw graph data to a `.csv` file.
7. Toggle curve legends.
8. **Add Markers**: Annotate the timeline with event markers. Markers can be exported to a `.csv` file.

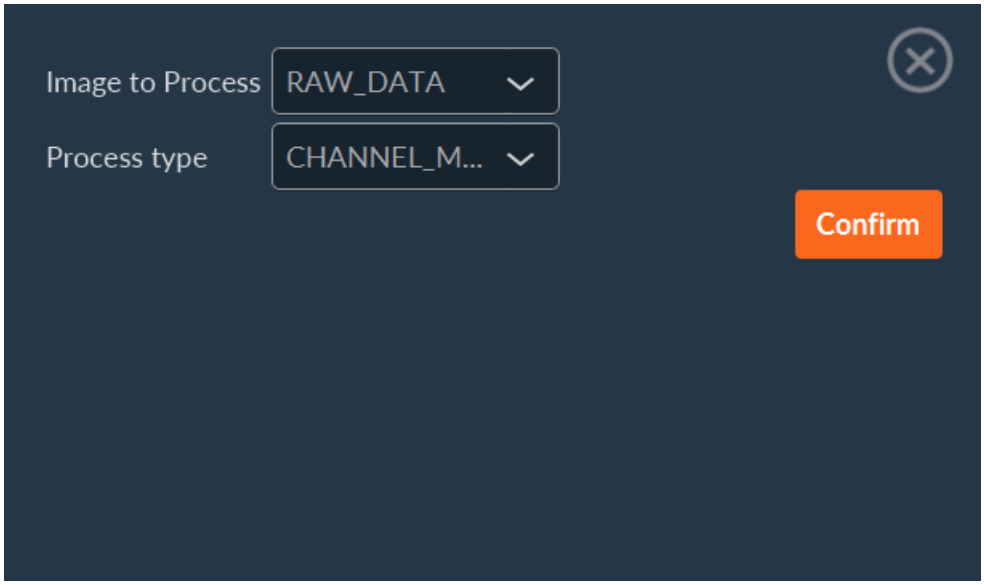


In this window you can add information about your experiment as markers. These events, which are fully customizable, can be associated with the acquisition itself (start, pause, end), external events (addition of an inhibitor, medium supplementation), or others. These markers can be saved in a .csv file and reused at any time.

All rights reserved.

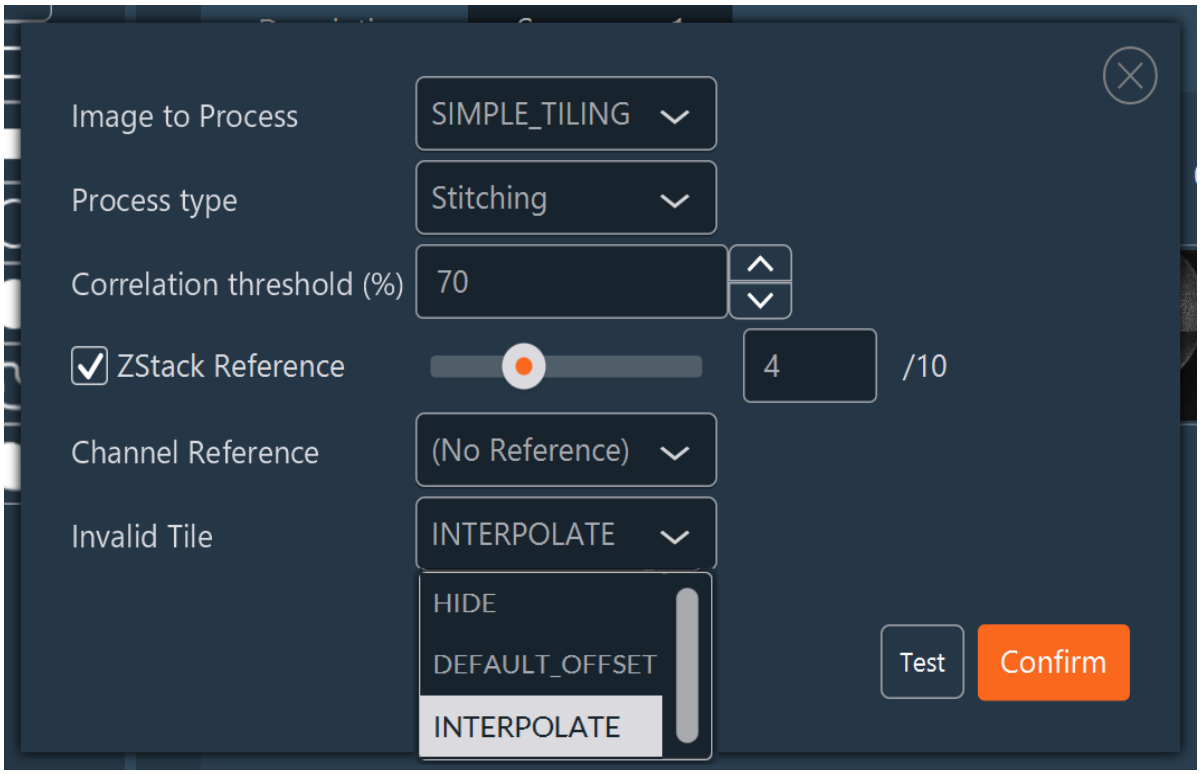
1.5.4 Image Processing

Apply post-acquisition data processing from the Visualization tab.



1. Select the target sequence from the image drop-down list.
2. Choose the desired data processor.
3. Click **Confirm** to execute.

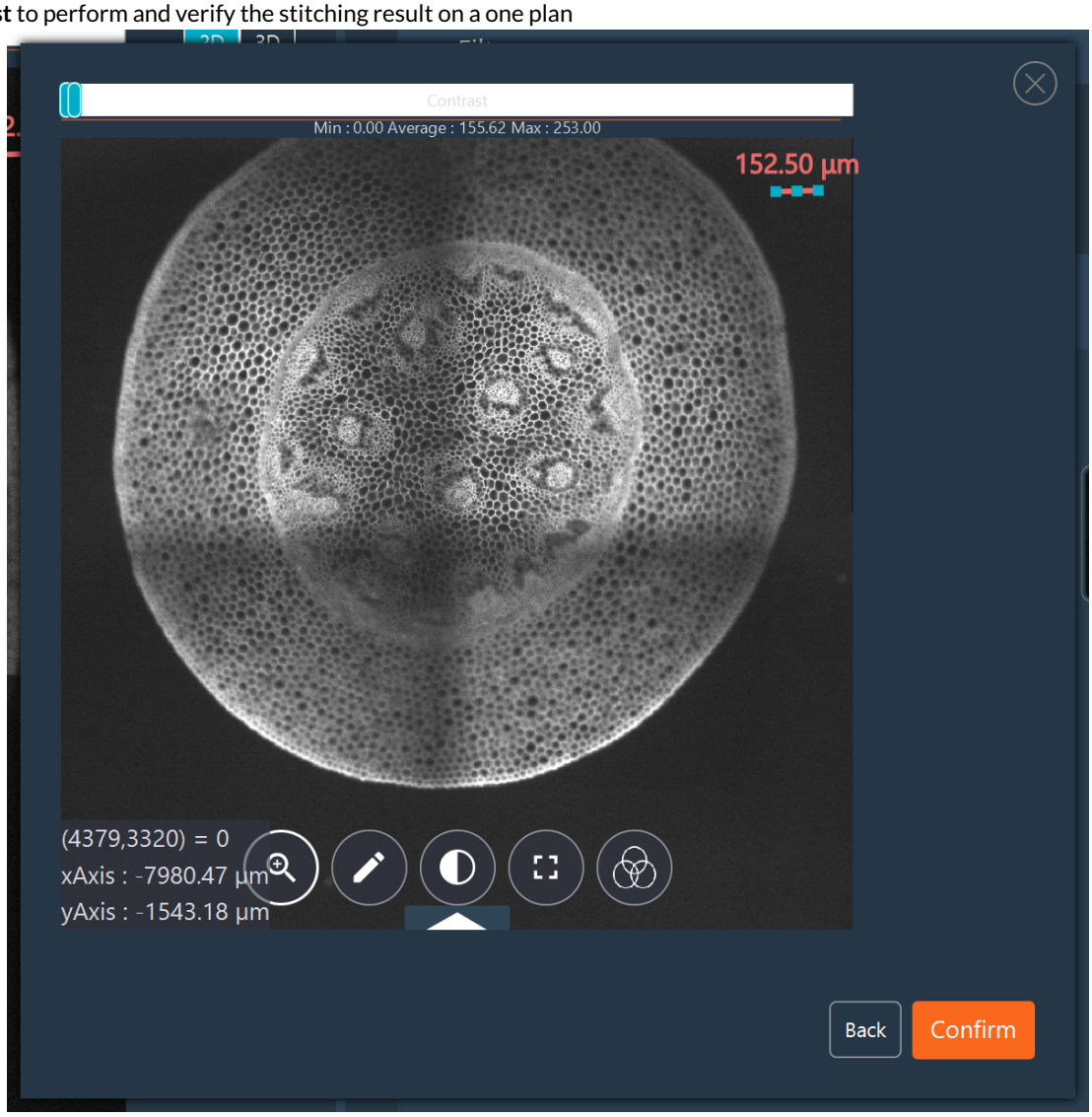
Example: Tiling Stitching



1. Select the **image to process**.
2. Select the `Stitching` **process type** (refer to the stitching algorithm for details on the registration and blending calculations).
3. Set the **correlation threshold (%)** for image stitching (the minimum correlation score, T_{corr}, required to validate the alignment between adjacent tiles; boundaries scoring below this threshold are flagged as invalid, triggering the configured fallback behavior).
4. Optional: Select a **Z-stack reference**, i.e. the focus plane used to calculate the stitching coordinates.
5. Optional: Select a **channel reference**, i.e. the channel used to calculate the stitching coordinates.

All rights reserved.

6. Define the fallback behavior for **invalid tiles** (uncorrelated):
    - `HIDE` : Discard the tile from the final image.
    - `DEFAULT_OFFSET` : Place the tile based solely on its recorded mechanical stage coordinates.
    - `INTERPOLATE` : Interpolate the tile's position based on the calculated offsets of surrounding valid tiles.
7. Click **Test** to perform and verify the stitching result on a one plan



8. Click **Confirm**.

1.5.5 Data Export

Export sequences as MP4 videos or multi-dimensional Tiff stacks via the Export menu (Toolbar Item 6).

Video Export

Use dimension filters to isolate the image sequence, then click **Video** from the Export dropdown.

All rights reserved.

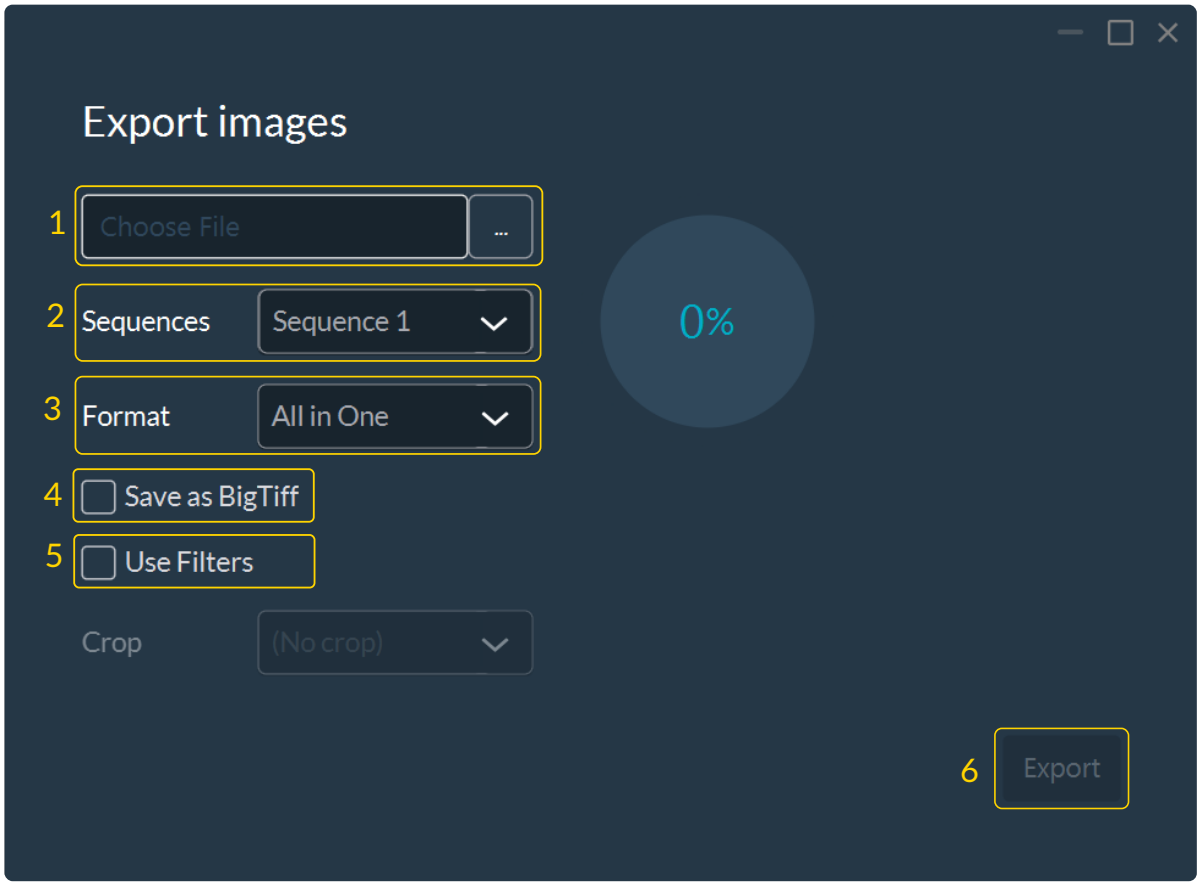


To export a video:

1. Select the destination file path.
2. Verify the sequence to export.
3. Select the video format/codec.
4. Click **Export**.

Stack Export

Use dimension filters to isolate images, then click **Stack** from the Export dropdown to generate a multidimensional TIFF.



To export a stack:

1. Select the destination file path.
2. Verify the sequence to export.
3. Select the data format.

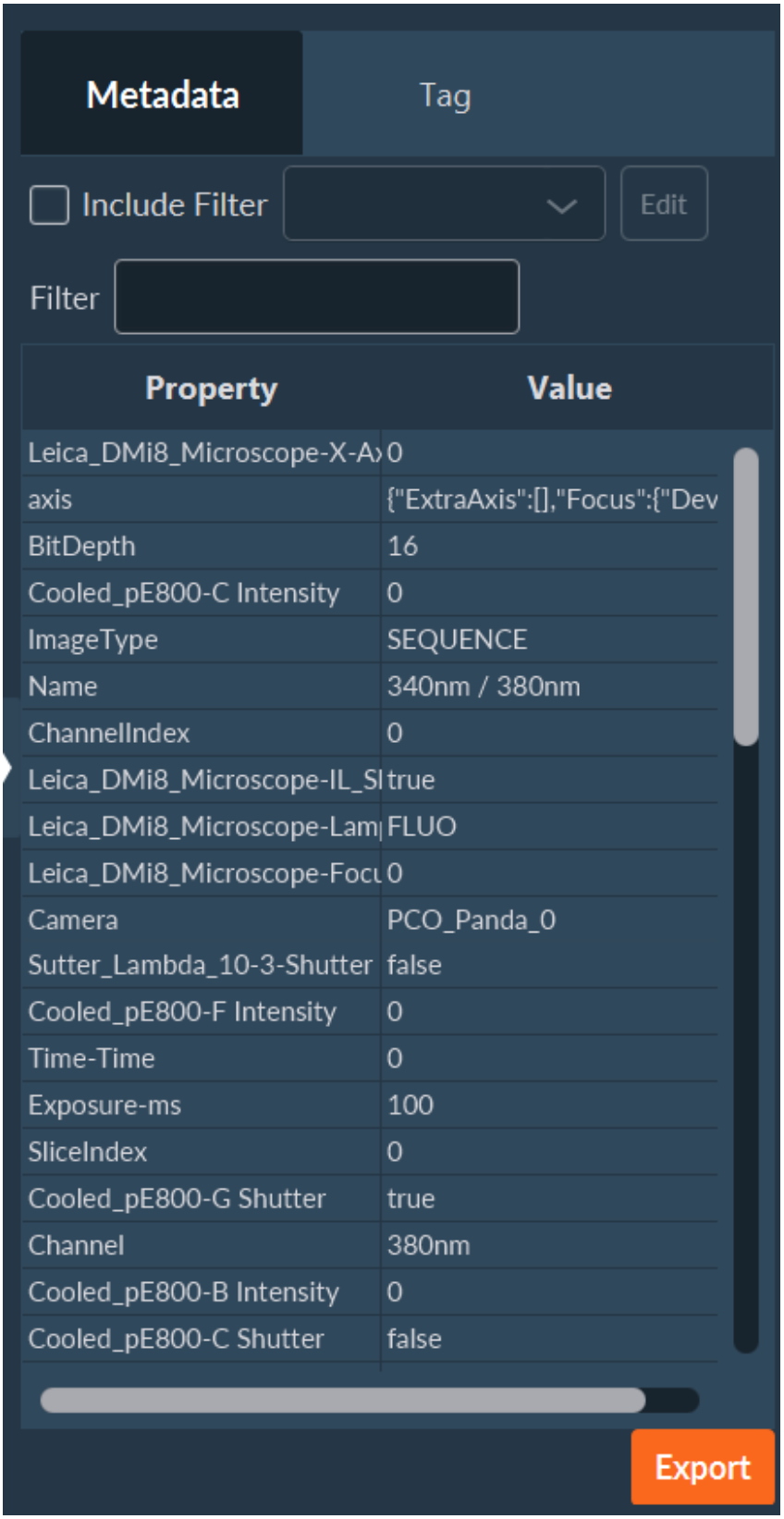
All rights reserved.

4. (Optional) Check **Save as BigTiff** if the expected file size exceeds 4GB.
5. (Optional) Check **Use Filters** to apply your visualization filters to the export.
6. Click **Export**.

Metadata Access

On the right side of the window there is a white triangle. You can click on it to access all the metadata. In this tab, you can access all the metadata, including the camera, light source or microscope settings; a search bar and some filters are available to facilitate the search for some specific parameters. This list can also be exported, if necessary, by clicking on the Export button located in the lower right part of the screen. All metadata are bio-format compatible.

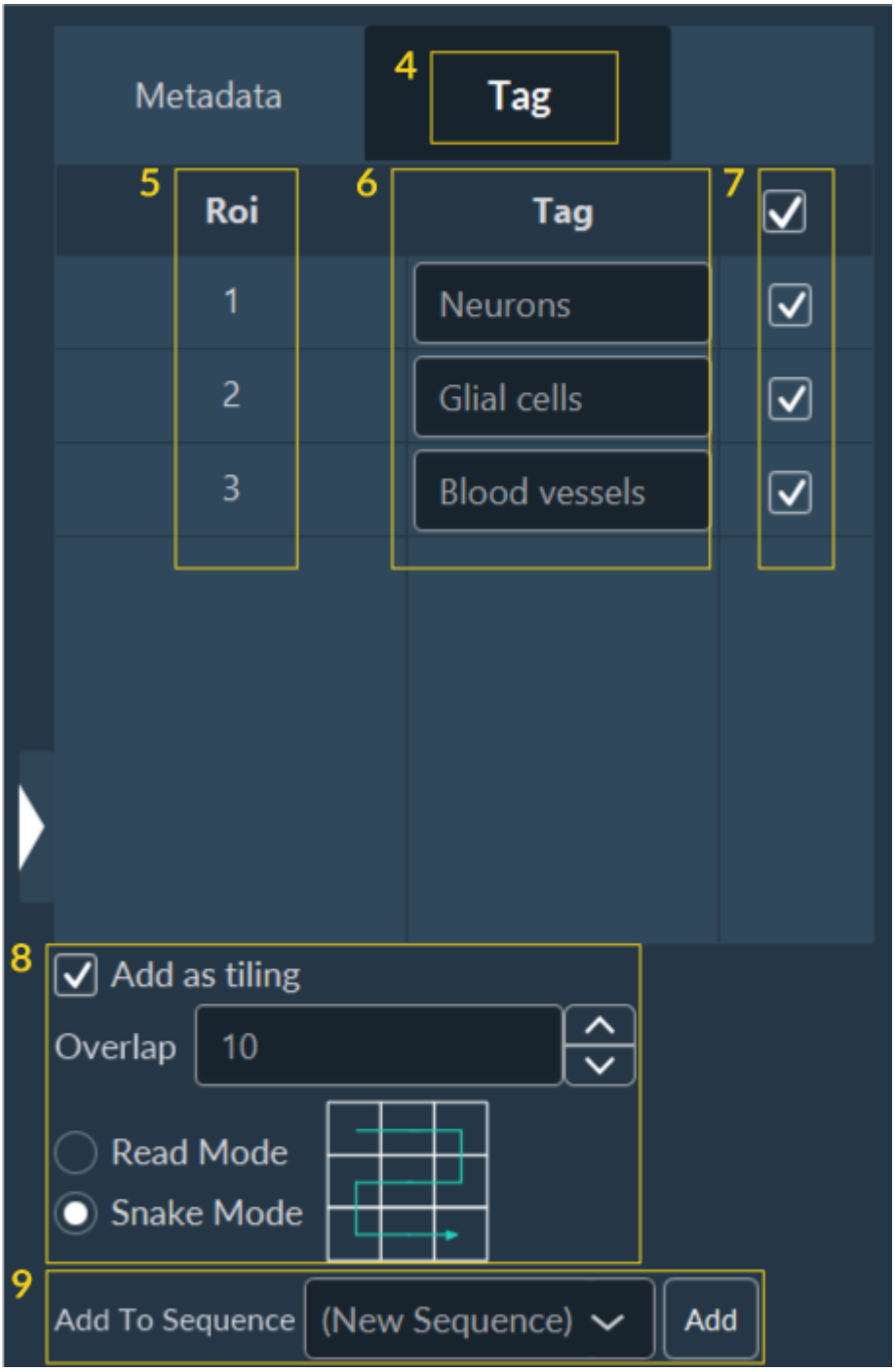
All rights reserved.



1.5.6 Semi-automated feedback microscopy feature

The software supports feedback microscopy workflows. For example, you can acquire a low-magnification overview scan, tag regions of interest, and export those coordinates into a new, high-magnification acquisition sequence.

All rights reserved.



To use this feature:

1. Acquire an overview image (e.g., a **Tiling** scan).
2. Use the ROI tools to draw regions around structures of interest.
3. In the **Visualization** tab, click the white triangle on the right edge to expand the side panel.
4. Navigate to the **Tag** sub-tab.
5. This lists all drawn ROIs. You can manage them directly from this list.
6. (Optional) Assign descriptive names/tags to each ROI.
7. Select the ROIs you wish to target.
8. If the new targets require their own sub-tilings, configure those spatial parameters here.
9. Choose to inject these new coordinates into a fresh sequence or append them to an existing one.

Smart Microscopy

Use this workflow for:

1. A low-magnification pre-scan followed by targeted high-resolution imaging.

All rights reserved.

2. A brightfield pre-scan, applying fluorescence imaging to identified structures to reduce global phototoxicity.

2026-09-08

All rights reserved.

1.6 Add-ons

1.6.1 Custom Python Scripting

Introduction to Python Scripting

The **Python Scripting** module bridges the gap between automated image acquisition and custom computer vision algorithms. By executing Python scripts after each image capture event, it enables smart microscopy: dynamic, adaptive workflows that use computational analysis to control microscope hardware. This module is built on two core components: a graphical interface and a development toolkit.

THE GRAPHICAL INTERFACE MODULE

> **Licensing Requirement**
>
> The **Custom Scripting** software license option is required to use this module.

Located natively within the Inscoper I.S. GUI, the module serves as the bridge between your custom scripts and the acquisition engine. It is found within the **multi-dimensional acquisition (MDA)** tab, immediately below the standard sequence settings.

- **Automated Settings UI**: Upon importing a Python script, the module parses the code to identify all functions (potential entry points). Parameters defined in your Python functions are mapped to graphical controls (e.g., `bool` arguments generate toggle switches, `str` become text fields, enumerations become dropdown lists).
    - The **Image Provider** function dictates which acquired images are passed to the Python processing function at any given time point.
    - The **Processing Function** executes the analysis (e.g., computer vision, intensity measurements) and returns the updated values for microscope hardware parameters.
- **Sequence Execution**: The system engine triggers these functions as callbacks following each image capture. The Image Provider is invoked first to assess if analysis is needed; if confirmed, the Processing Function runs, adjusting designated hardware parameters, or pushing figures to the interface.

THE DEVELOPMENT TOOLKIT (`inscoper-scripts`)

Provided as a local package, this toolkit serves as the foundation for custom script development. It contains a collection of Python scripts and utility sub-modules:

- **Application Scripts**: A suite of ready-to-use smart microscopy master applications, including modules for:
    - Mechanical tracking
    - Single-cell population screening/tracking
    - Autofocus
    - FRAP workflows
- **Structural Templates**: Simplified application examples that serve as starting points for generating custom scripts.
- **Utility Sub-modules**: A suite of helper libraries containing predefined image providers, ROI objects, and mathematical functions for tracking, measurement, segmentation, and filtering. These utilities simplify interacting with the core software environment, allowing you to display figures in the GUI, manipulate metadata dictionaries, and generate analytical plots.

2026-09-08

All rights reserved.

Calling Scripts in Inscoper I.S.

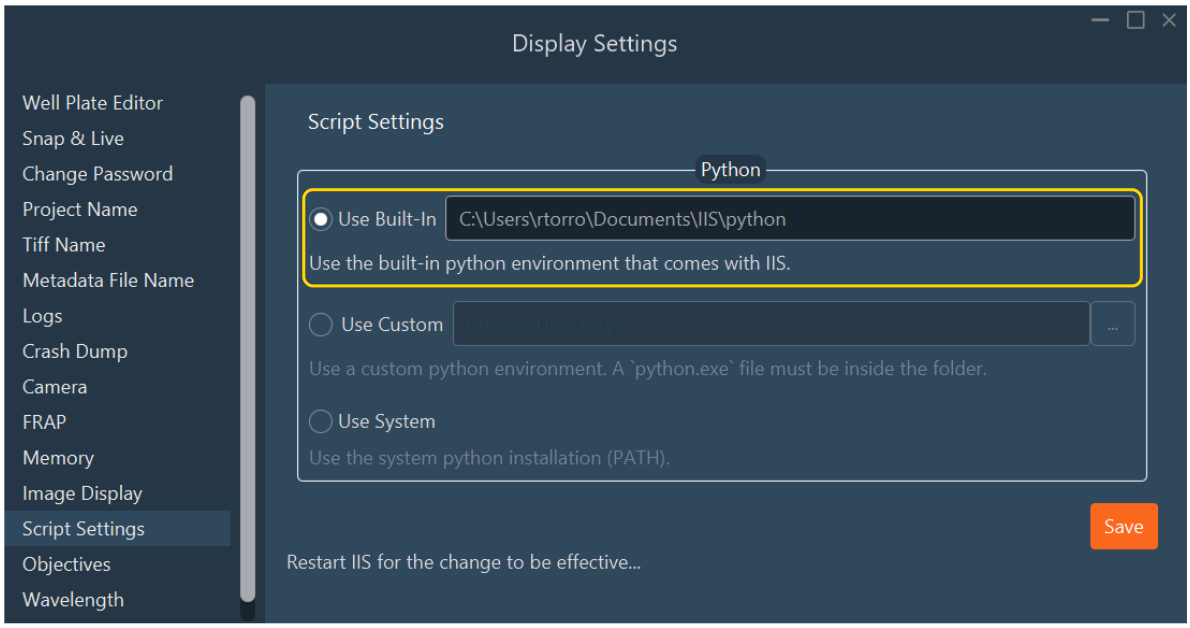
The Python Scripting module enables the execution of custom Python algorithms within an MDA sequence.

To ensure compatibility, scripts must adhere to the templates provided in the `inscoper-scripts` local package. You may use the included sample scripts for testing.

PYTHON ENVIRONMENT SETUP

Before running a script, select the Python environment the software uses.

1. Navigate to **Settings** > **Display Settings**.
2. Select **Script Settings** from the left-hand navigation pane.
3. Choose your preferred execution environment:
    - **Use Built-In (recommended)**: Uses the Python environment bundled with the software installation. It includes the standard (non-GPU) data science and image analysis packages.
    - **Use Custom**: Point the software to a specific, user-managed local Python executable (`python.exe`). Note: This advanced route explicitly requires the jep library (see warning below).
    - **Use System**: Uses the Python installation found in the PATH of the operating system.
4. **Restart Inscoper I.S.** to fully instantiate the environment changes.



Jep Package Required for Custom Environments

If you elect to use a custom environment (such as a specific Conda environment), the jep (Java Embedded Python) package is strictly required to bridge the software architectures.

Install it by activating your target environment and running:

```
pip install jep
```

Expanding the Built-In Environment

All rights reserved.

If your custom script requires specialized statistical or biological libraries not included in the standard distribution:

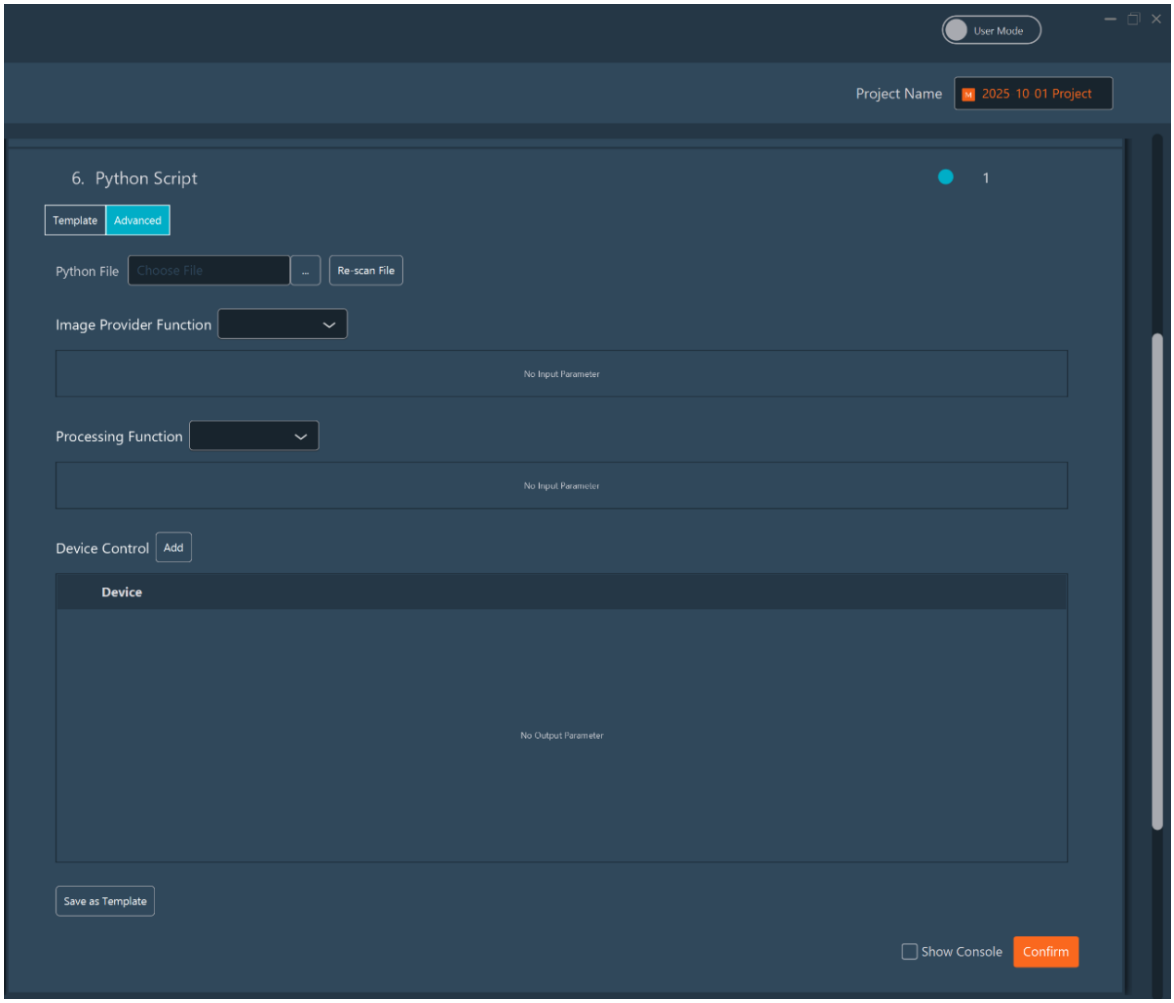
1. Open a command line terminal within the **I.I.S. installation directory**.
2. Navigate to the nested `python` subdirectory.
3. Execute the standard pip installation command:

```
./python.exe -m pip install <package_name>
```

IMPORTING AND MAPPING A SCRIPT APPLICATION

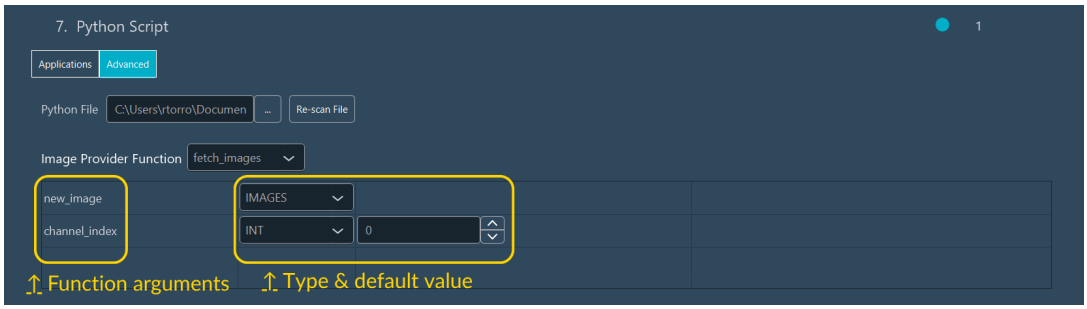
Once your Python interpreter is set, you can load your script into the MDA sequence definition.

1. Navigate to the central **Acquisition** tab and select the **MDA** (Multi-Dimensional Acquisition) sub-tab.
2. Expand the **Python Script** module panel and toggle the **Advanced** switch to enable this mode.
3. Click the **"..."** button adjacent to **Re-scan File**, browse to the `inscoper-scripts` local package, and select your specific `.py` script.



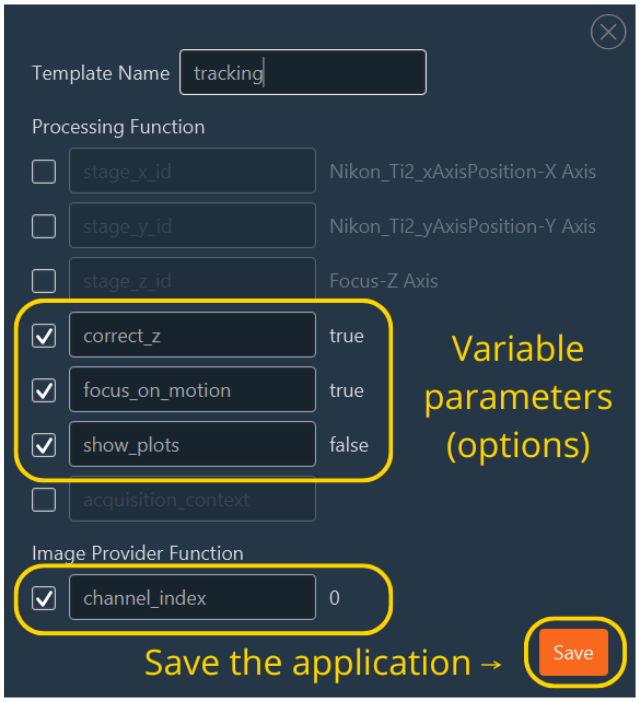
1. **Map Python functions to I.I.S. parameters.** Define which function handles image fetching (image provider) and which handles heavy-computations (processing), then map the functions' arguments to the proper type and value.
    1. **Step A (image provider)**: Select the script function responsible for fetching images (e.g., `fetch_images`).
        - **Map Arguments**: Map the script variable `new_image` to the `IMAGES` data type. This allows the software to pass the most recently acquired image to your script.
        - Map `channel_index` to `INT` and assign a default value (e.g., `0`). These channel indices correspond to their order within the Multi-channels dimension.

All rights reserved.



1. **Step B (processing function)**: Select the function responsible for data analysis (e.g., `correlation_tracking`).
    - **Map Arguments**: Map the `images` variable to the `IMAGES` type. This variable receives the images curated by the Image Provider function.
    - Map hardware identifiers (e.g., `stage_x_id`, `stage_y_id`, `stage_z_id`) to the `SUBDEVICEID` type and set default values matching your hardware.
    - Map logical switches (e.g., `correct_z`, `focus_on_motion`, `show_plots`) to `BOOL` and define initial states.
    - Map `acquisition_context` to `IIS_CONTEXT`. This provides access to internal software APIs, such as rendering `matplotlib` figures in the UI (see Interacting with Inscoper I.S.).
2. **Step C (device control)**: Declare any hardware sub-devices that will be updated by your processing function's outputs (it should match symmetrically with the return values of the processing function, e.g. if the processing function returns a tuple of new 'x', 'y', and 'z' values, then you should declare the X, Y, and Z sub-devices here). If your function only logs data without modifying hardware, leave this blank.

2. **Save as a Reusable Template**
    1. Once mapping is complete, click **Save as Application Template**.
    2. In the dialog box, select which parameters should remain editable by the user.
    3. Unselected parameters will be hardcoded to their default values and hidden from the UI.



Configuring the UI for Script Visualization

If your Python script is designed to output interactive visual graphs (via `matplotlib`) or text logs to the console, you must configure a data processor to receive this data.

1. Enable **Expert Mode** using the top-right interface toggle.
2. Within the **Data Processing and Charts** operational section, click **Switch to Data Processor**.

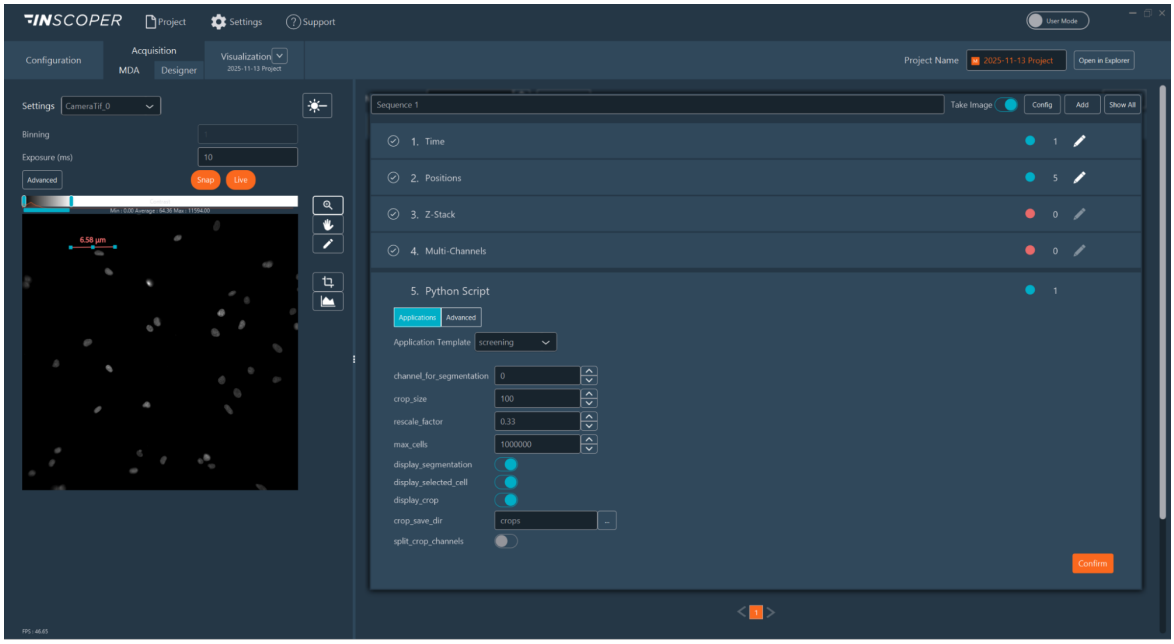
All rights reserved.

3. **Enable Console Logging**: Drag the **Feedback Console** module from the library into the right-hand panel workspace. This active console captures and displays all standard Python `print()` logging outputs.
4. **Enable Graphical Output**: If your script pushes a figure with `display_figure(..., processor_name='name')`:
    - Drag a **Feedback Chart** module into the right-hand panel workspace.
    - Rename this chart element in the interface so that it matches the `processor_name` string used in your script exactly. The figure does not appear otherwise.
5. Save this complete charting layout with a descriptive, logical name and click **Add**.
6. Click **Switch to Dimensions** to return to the standard sequence configuration window.

EXECUTING THE SCRIPTED EXPERIMENT

Executing a templated script is designed for operational simplicity.

1. Navigate to the **Application** tab.
2. Select your saved template from the dropdown list.
3. Adjust any exposed control parameters.
4. Verify that the **Data Processing and Charts** menu selection matches the layout required by the script.
5. Click **Start Acquisition** to initiate the sequence.



2026-09-08

All rights reserved.

Writing Custom Scripts

This module runs Python functions during acquisition. Scripts must follow the templates in the `inscoper-scripts` repository.

IMAGE PROVIDER FUNCTION

The **image provider** function is your script's entry point. The acquisition engine calls it **after every image capture**. It has two responsibilities:

1. **Gating**: Decide whether the processing function should run (e.g., wait until a Z-stack is complete).
2. **Image selection**: Decide which images to fetch from memory and pass to the processing function (e.g., the full Z-stack, or only the last 2D frame).

Signature

Arguments

- `new_image` (`images`): The first argument, typed `images` . It receives the **last image acquired** by the hardware, together with the metadata that identifies the multidimensional state of the microscope (time, position, channel, and Z-slice indices).
- **Optional arguments**: Define additional parameters (such as `channel_index` or `focus_option`) that the user configures in the interface.

Return Value

The image provider function must return a `List[Dict[str, int]]` .

- **Empty list `[]` or `None` : Skips** the processing function for the current iteration.
- **Non-empty list of dictionaries**: **Runs** the processing function. Each dictionary specifies one image to fetch from memory.
  - **Empty Dictionary `{}` :** Fetches the **current image** (indices are inferred from the `new_image` object).
  - **Specific Indices** (e.g., `{'focusIndex': 3}`): Fetches an image from a specific coordinate (e.g., Z-slice 3). Any dimension index not explicitly declared (e.g., `timeIndex` , `positionIndex`) is copied from the current `new_image` .

Dimension	Index key	Size key
Time (t)	`timeIndex`	`timeSize`
Position (XY)	`positionIndex`	`positionSize`
Channel (λ)	`channelIndex`	`channelSize`
Focus (Z)	`focusIndex`	`focusSize`
Lifetime (FLIM)	`flimIndex`	`flimSize`

Code Example: Fetching an Entire Z-stack

This example checks whether the current image is the last slice of a Z-stack. If so, it collects all prior slices so that the Imaging Software can dispatch this list of images to the processing function; if not the final slice, it does nothing and awaits the next capture. Note that the `positionIndex` and `timeIndex` are inferred from the `new_image` object.

```
from inscoper_scripts.utils import extract_metadata
import numpy as np
from typing import List, Dict

def fetch_zstack(new_image: "images", channel_index: int = 0) -> List[Dict[str, int]]:
    """
    Collect the full Z-stack for the current position in the target channel.
    """
    # 1. Read the metadata dictionary of the current image
    meta = extract_metadata(new_image)
```


All rights reserved.

```
    current_z = meta.get('focusIndex', 0)
    total_z = meta.get('focusSize', 1)

    # 2. Run only on the last Z slice
    if current_z != (total_z - 1):
        return [] # An empty list skips the processing function

    # 3. Select every slice, 0 to total_z-1
    # Only 'focusIndex' and 'channelIndex' are given;
    # 'timeIndex' and 'positionIndex' are inferred from the current image.
    image_selection = []
    for z in range(total_z):
        image_selection.append({
            'focusIndex': z,
            'channelIndex': channel_index
        })

    return image_selection
```

Code Example: Fetching the Current Image

This is the simplest case. A list containing one empty dictionary sends only the current image to the processing function.

```
def fetch_current_image(new_image: "images") -> List[Dict[str, int]]:
    return [{}]
```

Code Example: Fetching the Last Two Timepoints

This pattern retrieves the images from the previous (`t-1`) and current (`t`) timepoints. Use it for tracking algorithms.

```
def fetch_last_two_timepoints(new_image: "images") -> List[Dict[str, int]]:
    meta = extract_metadata(new_image)
    current_t = meta.get('timeIndex', 0)

    # At least two timepoints are required (index 0 and index 1)
    if current_t < 1:
        return []

    return [
        {'timeIndex': current_t - 1}, # Previous timepoint
        {'timeIndex': current_t}      # Current timepoint
    ]
```

PROCESSING FUNCTION

In the data processing function, the first argument must be typed `images`. It receives the list of image arrays returned by the image provider.

All rights reserved.

SUPPORTED ARGUMENT TYPES AND UI AUTOMATION

Python type	Generated widget	Description
`str`		A string parameter, rendered as a text input field.
`int`		An integer value, rendered as a spin box with an increment of 1.
`float`		A floating-point value, rendered as a spin box with an increment of 0.001.
`bool`		A boolean value, rendered as a toggle switch.
`Path` or `str` with `/` or `\\`		A directory path, rendered as a directory chooser.
`Literal["val1", "val2", "val3"]`		A value constrained to an enumeration, rendered as a single-selection dropdown list.
`List[Literal["val1", "val2", "val3"]]`		Several values constrained to an enumeration, rendered as a multi-selection dropdown list.
`images`		Represents one or more image arrays provided by the Imaging Software (the last acquired image for the image provider function, or a curated list of images for the processing function).
`SubDeviceId`		A sub-device name (used as a key to read its value in the metadata), rendered as a dropdown list of all available sub-devices.
`channel`		A channel name, rendered as a dropdown list of all available channels.
`IIS_CONTEXT`		A context object, allowing the script to interact with the software environment (variable transfer, method execution).

Set default values in the function signature, using standard Python syntax: e.g., `var1: float = 0.5` or `List[Literal["val1", "val2", "val3"]] = ["val2"]`.

INTERACTING WITH INSCOPER I.S.

The `IIS_CONTEXT` exposes methods allowing your script to interact with the software environment.

`getCameras()`

Retrieves a list of active cameras.

- **Returns**: `List[str]` - The camera names.

Example:

```
cameras = IIS_CONTEXT.getCameras()
# Returns, for example: ["Camera1", "Camera2"]
```

`getProjectPath()`

Retrieves the path of the currently active project directory.

- **Returns**: `str` - The active project path.

Example:

```
path = IIS_CONTEXT.getProjectPath()
# Returns, for example: "D:/Data/Project_001"
```


All rights reserved.

`displayImage(id, img)`

Sends an image array to a data processor in the visualization dashboard.

- **Parameters**:
  - `id` (`str`): An identifier matching the target data processor.
  - `img` (`NDArray`): The image data to display.

`setVariable(id, value)`

Stores a variable in the application. Use it to pass values between calls during the acquisition loop.

- **Parameters**:
  - `id` (`str`): The key identifying the stored variable.
  - `value` (`Object`): The value to store.

Example:

```
IIS_CONTEXT.setVariable("tracking_counter", 1)
```

`getVariable(id)`

Retrieves a variable previously stored with `setVariable` .

- **Parameters**:
  - `id` (`str`): The key of the variable.
- **Returns**: `object` - The stored value.

Example:

```
count = IIS_CONTEXT.getVariable("tracking_counter")
```

`fireOnDemand(cameraName, roi, channelName, powerMap, repetition)`

Triggers a FRAP photo-activation sequence on the given region of interest (ROI), using the specified channel and power settings.

- **Parameters**:
  - `cameraName` (`str`): The name of the camera to be used for the photo-activation sequence (this string must exactly match one of the returned values from the `getCameras()` execution).
  - `roi` (`Object`): The region of interest. Pass an ROI instance (e.g., `RectangleROI`) from the `inscoper_scripts.utils.rois` module, or a list of such instances. Given a list, the system photoactivates each ROI in turn.
  - `channelName` (`str`): The name of the channel to use.
  - `powerMap` (`Object`): The illumination power settings, as a Python dictionary (e.g., `{"MyLaserSubDevice": 100}`).
  - `repetition` (`Object`): The number of repetitions (e.g., `10` runs 10 consecutive cycles).

Example:

```
from inscoper_scripts.utils.rois import RectangleROI

# Define the parameters
cam = "Camera1"
my_roi = RectangleROI(0, 0, 512, 512, fill=False)
channel = "GFP_Activation"
power = {"488nm_Laser": 50.0}
reps = 1

# Trigger the sequence
IIS_CONTEXT.fireOnDemand(cam, my_roi, channel, power, reps)
```


All rights reserved.

2026-09-08

All rights reserved.

Python Script Library Reference

This page is the reference for the pre-built Python scripts in the `inscoper_scripts` library. For each script, it lists the entry points, the required data processors, the files written to disk, and the metadata injected into the acquisition.

SMART MICROSCOPY SCRIPTS

Screening

Single-cell screening (deep learning)

- **Script name**: `single_cell_screening.py`
- **Entry points**:
  - Image provider: `fetch_images`
  - Processing function: `detect_single_cells`
- **Required data processors**: `segmentation` , `crop`
- **Description**: Automatically screens and segments individual cells within an acquisition sequence. It uses a deep learning segmentation algorithm (Cellpose) to detect isolated single cells in a selected fluorescence channel, crops each cell, and writes the crops to disk. The interface displays the per-cell measurements.
- **Disk output**: `crops/*.json` , `crops/*.tif`
- **Injected metadata**: `NBR_OF_CELLS` , `MEAN_CELL_AREA` , `MEDIAN_CELL_AREA` , `ROI_POINTS`

Single-cell screening (thresholding)

- **Script name**: `single_cell_screening_no_ai.py`
- **Entry points**:
  - Image provider: `fetch_images`
  - Processing function: `detect_single_cells` , `run_detect_single_cells_async`
- **Required data processors**: `segmentation` , `crop`
- **Description**: Equivalent to `single_cell_screening.py` , but segments cells with classical image processing (intensity thresholding and watershed transforms) instead of deep learning. This is faster, but less reliable on confluent or cluttered fields.
- **Disk output**: `crops/*.json` , `crops/*.tif`
- **Injected metadata**: `NBR_OF_CELLS` , `MEAN_CELL_AREA` , `MEDIAN_CELL_AREA` , `ROI_POINTS`

Cell tracking

Cell population tracking

- **Script name**: `cell_population_tracking.py`
- **Entry points**:
  - Image provider: `fetch_images`
  - Processing function: `cell_population_tracking`
- **Required data processors**: `segmentation` , `tracks` , `timeseries`
- **Description**: Tracks a population of cells over time in a 2D or 3D sequence. It segments each frame (with Cellpose or thresholding), links cells between frames to build tracks, and computes running population metrics (mean and median velocity or intensity). It includes heuristics for cells that are temporarily lost and for global stage drift.
- **Disk output**: `tracks.csv` (if enabled)
- **Injected metadata**: `NBR_OF_CELLS`

Stage-based tracking

Correlation tracking of a target object

- **Script name**: `correlation_tracking_withZ.py`
- **Entry points**:
  - Image provider: `fetch_images`

All rights reserved.

    - Processing function: `correlation_tracking`
- **Required data processors**: `motion_difference` , `motion_mask`
- **Description**: Tracks an object in 3D using phase correlation. It computes the XYZ shift between the current and previous timepoints and updates the stage position to keep the object centered in the field of view. It also corrects the Z position by locating the best focus plane, and can estimate sub-pixel shifts.
- **Disk output**: `tracks.csv` (if an export path is set)
- **Injected metadata**: None. The computed shifts are written to the logs.

Photomanipulation

FRAP and photo-activation

- **Script name**: `roi_manipulation.py`
- **Entry points**:
    - Image provider: `fetch_images`
    - Processing function: `compute_roi`
- **Required data processors**: `image` , `roi`
- **Description**: Shows how to define and manipulate regions of interest (ROIs) on an incoming image array. It creates rectangles, circles, and lines, and can fire the laser on demand over the ROI geometry. Use it as a starting point for automated FRAP or photo-activation workflows.
- **Disk output**: None
- **Injected metadata**: None

Focus

Software autofocus

- **Script name**: `autofocus.py`
- **Entry points**:
    - Image provider: `fetch_images`
    - Processing function: `auto_focus`
- **Required data processors**: `focus` (implicit)
- **Description**: Determines the best Z position from a Z-stack acquired during the sequence. It measures image sharpness (contrast gradients) across the slices and returns the Z position of the sharpest focus plane. This provides software autofocus during a time-lapse without a hardware focus device.
- **Disk output**: None
- **Injected metadata**: None. The selected Z position is written to the logs.

Image correction

Background model

- **Script name**: `background_model.py`
- **Entry points**:
    - Image provider: `fetch_images`
    - Processing function: `correct_background`
- **Required data processors**: `background` , `normalized`
- **Description**: Corrects the background illumination. At the first timepoint t_0, it builds a background model (for example, the median intensity projection across several fields of view). At later timepoints, it applies that model to flat-field correct the incoming images. It can mask out cells and objects before building the model.
- **Disk output**: `background.tif` (the background model)
- **Injected metadata**: None

All rights reserved.

TEMPLATES

Demonstration workflow

- **Script name**: `demo.py`
- **Entry points**:
  - Image provider: `fetch_images`
  - Processing function: `find_most_eccentric_cell`
- **Required data processors**: `segmentation` , `crop`
- **Description**: A demonstration script that chains background subtraction, Cellpose segmentation, feature extraction (finding the most eccentric cell), and a stage move onto that cell. Use it as an example of combining several API features in one sequence.
- **Disk output**: `background.tif` , `crops/*.tif` , `crops/*.json`
- **Injected metadata**: `NBR_OF_CELLS`

Python environment check

- **Script name**: `test_python_env.py`
- **Entry points**:
  - Image provider: `fetch_images`
  - Processing function: `test_versions`
- **Required data processors**: None
- **Description**: A diagnostic script that verifies the Python environment. It prints the installed TensorFlow version to the console, which confirms that the deep learning dependencies are installed and reachable from the software.
- **Disk output**: None
- **Injected metadata**: None

TYPE HINTS FOR INTERFACE MAPPING

The type hints below declare your script parameters. The software uses them to generate the matching input widget in the interface.

- `"images"` : The list of buffered input images passed to the script.
- `"IIS_CONTEXT"` : The acquisition context object. It gives access to the microscope hardware state and to the execution methods (e.g., `fireOnDemand()`).
- `"SubDeviceId"` : The identifier of a hardware sub-device (e.g., "Stage X", "Laser 488nm"). The interface provides a dropdown list limited to the connected devices.
- `"channel"` : Represents a channel configuration object.
- `Literal["val1", "val2"]` : Creates a dropdown list containing only the given string options.
- `List[Literal[...]]` : Creates a multi-selection checkable list of the given options.
- `Path` : A file or directory path. The interface provides a file or directory picker.
- `bool` : Creates a checkbox or toggle switch.
- `int` , `float` , `Number` : Create numeric input fields.

> **Reference Examples**
>
> The `input_types.py` script shows one example of every type mapping.

SKIPPING THE PROCESSING FUNCTION

To skip processing for a given acquisition event (for example, to analyze only the last slice of a Z-stack, or only selected timepoints), return an empty list from `fetch_images` : `[]` .

When `fetch_images` returns `[]` , the software does not call the processing function (e.g., `detect_single_cells`) for that event.

All rights reserved.

Example (`_image_provider_functions.py`):

```python
def fetch_zstack(new_image, channel_index: int = 0) -> List[Dict]:
    # ... preliminary metadata extraction ...
    current_focus = meta_dict.get('focusIndex', 0)
    size_z = meta_dict.get('focusSize', 1)

    # Process only on the last Z slice
    if current_focus != (size_z - 1):
        return [] # An empty list skips the processing function

    # ... otherwise construct and return the required image list ...
```

This pattern restricts expensive operations, such as neural network inference, to the events where the required data (here, the complete Z-stack) is in memory.

2026-09-08

All rights reserved.

1.6.2 Photomanipulation (ScanFRAP)

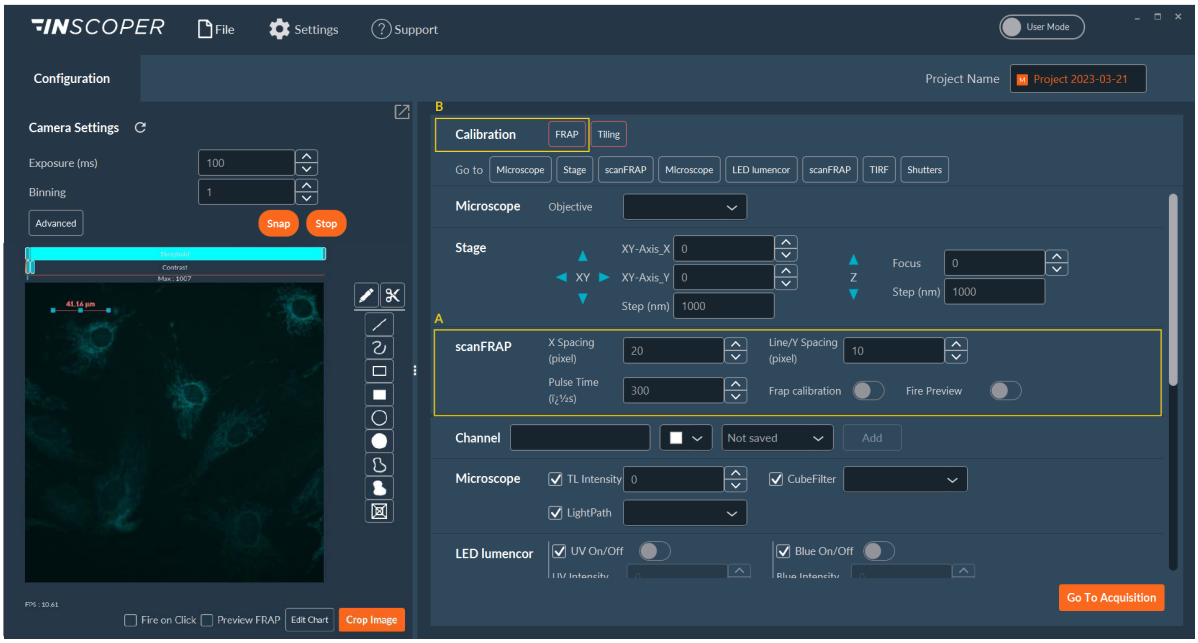
The **scanFRAP** add-on enables targeted laser-based photomanipulation (such as FRAP, photoactivation, or optogenetics) integrated into your imaging sequences.

Configuration & Calibration

GLOBAL CONFIGURATION

Section (A) of the **scanFRAP** Configuration tab controls the geometric point density during active photomanipulation sweeps.

- **X and Y spacing**: Defines the laser spot density used to fill solid geometric shapes (e.g., circles, rectangles).
- **Line/Y spacing**: Defines the laser spot density used exclusively for lines or unfilled shape boundaries.
- **Fire Preview**: Activates the manipulation laser within currently selected ROIs to allow safe, empirical testing of your power configurations.



> **Note**
>
> The **Fire Preview** function requires manually drawn ROIs and the correct manipulation channel to be actively selected.

All rights reserved.

FRAP CHANNEL

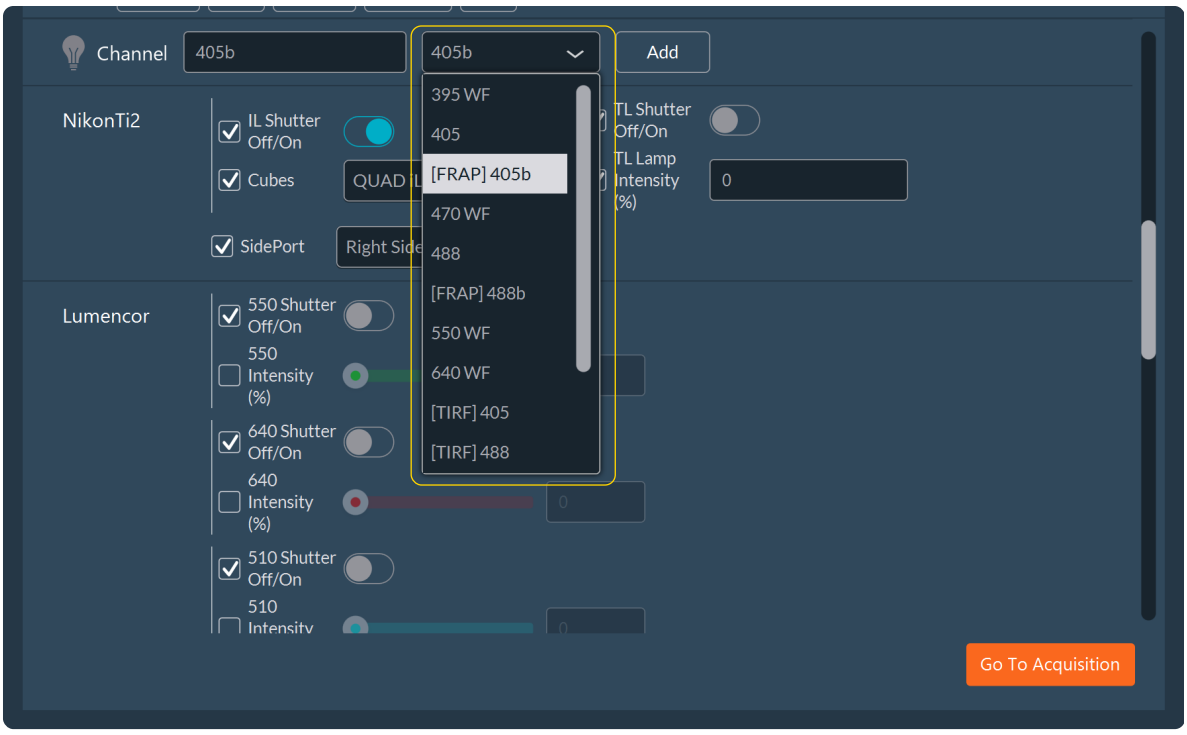


Photomanipulation requires a dedicated FRAP channel. To create one:

1. Select the laser line of interest.
2. Set the TIRF/FRAP shutter to FRAP and the imaging mode to FRAP.

Click **Add** to save the channel temporarily (for the current session), or **Save** to store it permanently in the database.

A channel is automatically identified as a FRAP channel when the MaxiILDA shutter is toggled on and the TIRF/FRAP mode is set to FRAP. It appears as "[FRAP] Channel Name" in the channel list of the Configuration tab.



Note

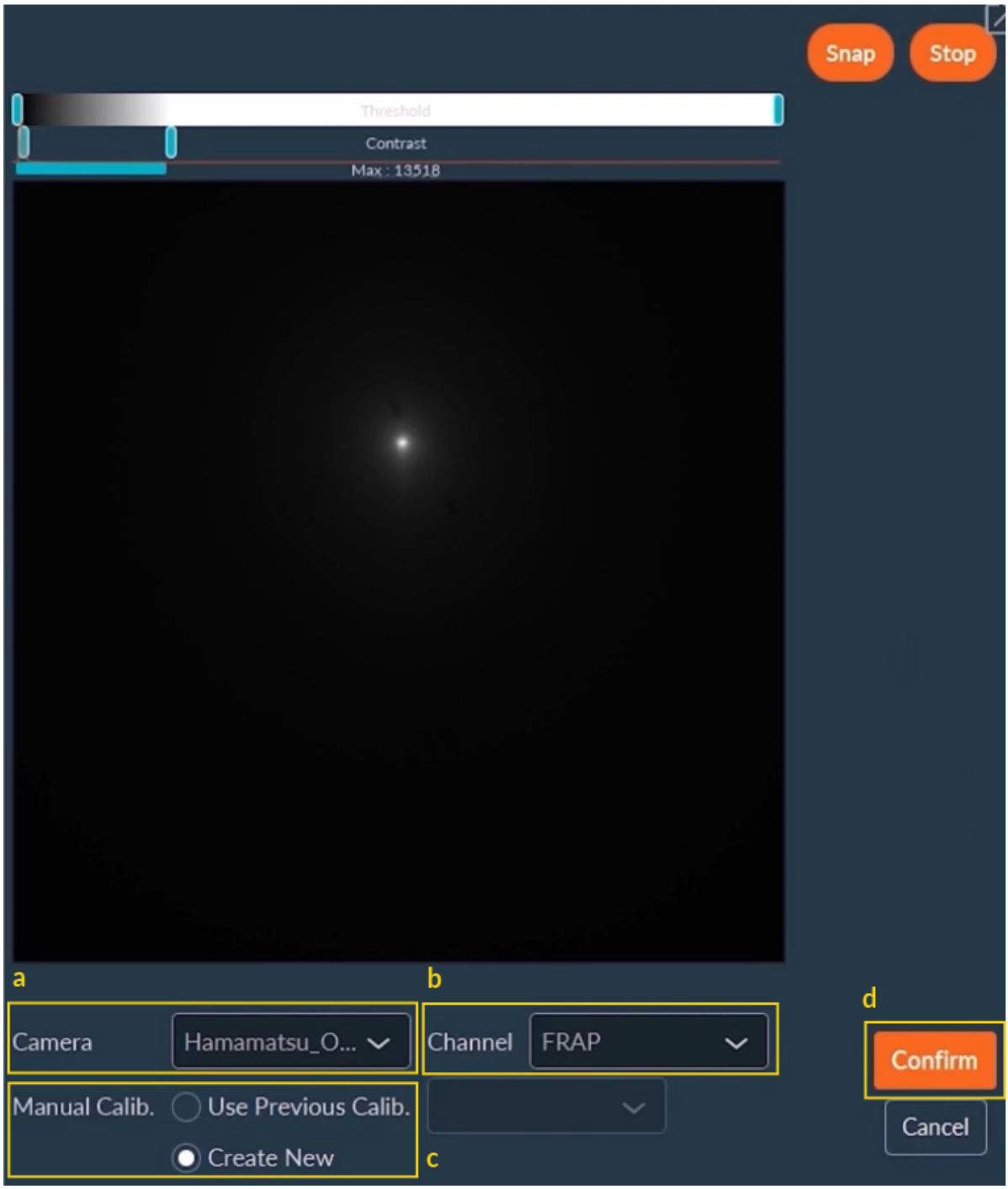
FRAP channels are reserved for photomanipulation applications and cannot be used for standard live imaging. They will not be found in the channel shortcut above the livestream.

All rights reserved.

SPATIAL CALIBRATION

scanFRAP requires spatial calibration to ensure targeting accuracy. Recalibrate the galvanometer mirrors before each experimental session.

Navigate to the generic calibration menu (B) and click **FRAP**.



- **a**. Select the currently active imaging camera.
- **b**. Select the pre-configured laser channel designated for photomanipulation.
- **c**. Choose an existing archived calibration profile or initiate a new one.
- **d**. Click **Confirm** to route the optical settings.

Tip

Perform the spatial calibration on a fluorescent reference slide (e.g., plastic or glass). It provides a strong, uniform signal and avoids photobleaching the biological sample.

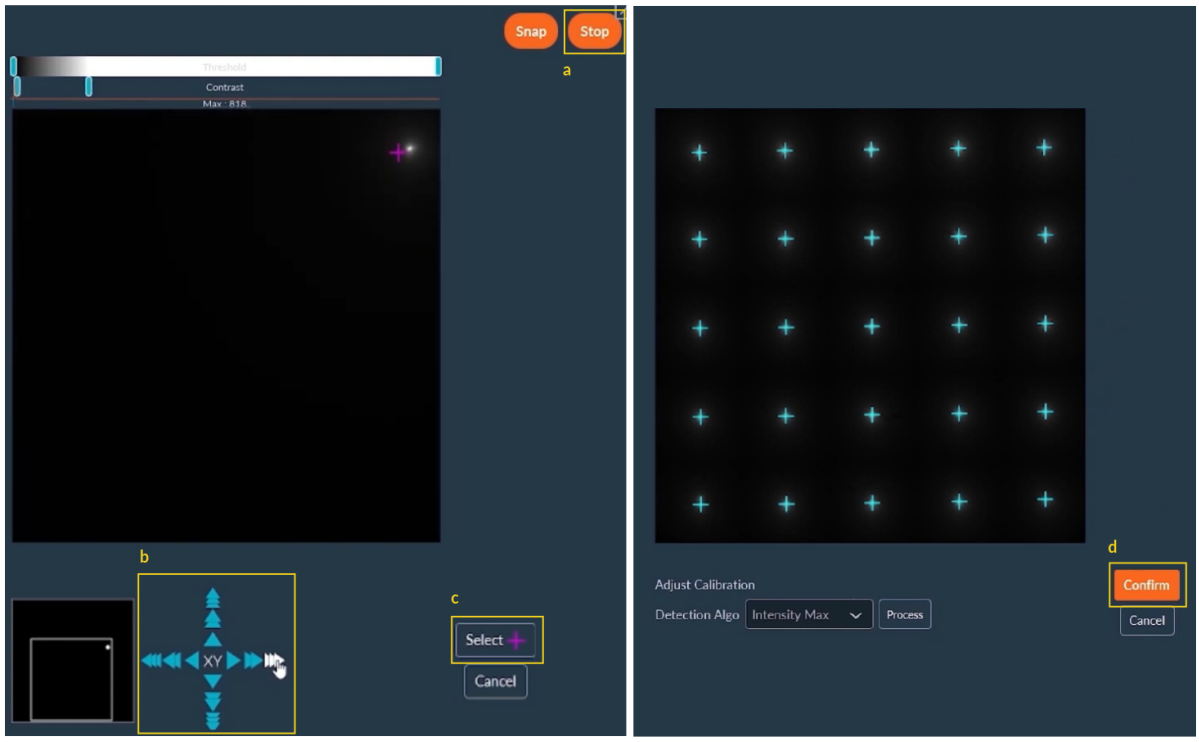
Warning

All rights reserved.

Maintain optimal Z-focus throughout this process. Laser targeting accuracy depends on a sharp focal plane.

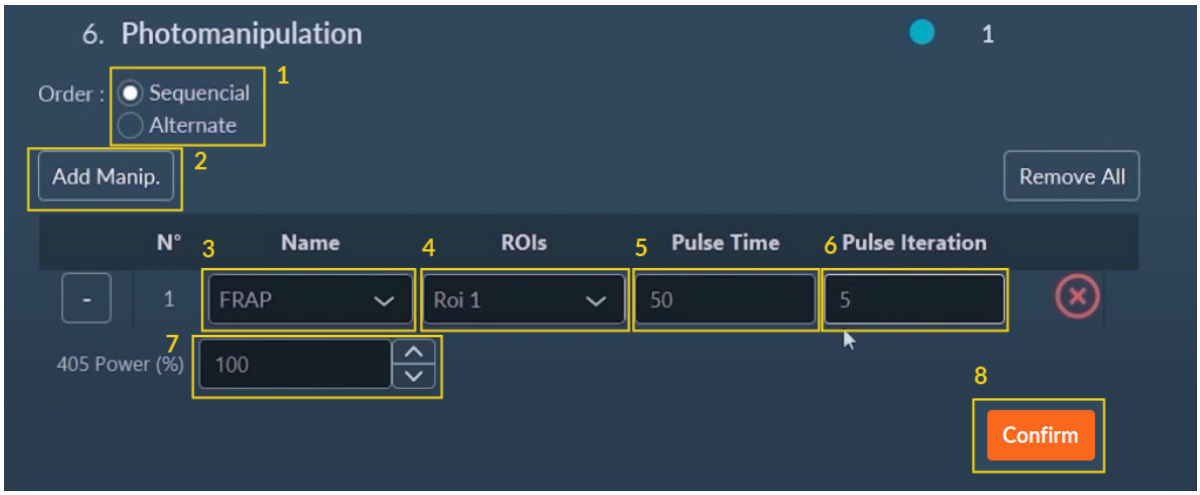
New calibration

- **a**. Activate **Live** mode to visualize the reference slide.
- **b**. Use the virtual joystick to align the laser pointer with the software's virtual crosshair.
- **c**. Click **Select**. Repeat this alignment process three times at different XY coordinates across the field of view.
- **d**. The software will generate a verification pattern. Validate its geometric regularity (e.g., straight lines, sharp corners), then click **Confirm** to accept the calibration matrix.



Sequence Acquisition

scanFRAP procedures are incorporated into your sequence via the **Photomanipulation dimension**.



To configure the **Photomanipulation** dimension:

1. Select the scanning mode: Scan assigned ROIs one at a time (**Sequential**) or interleave the scanning paths between them (**Alternate**).
2. Click **Add Manip.** to instantiate a new photomanipulation event block.
3. Select the pre-configured laser channel assigned to drive the photomanipulation.
4. Select and activate all target ROIs on the sample preview.

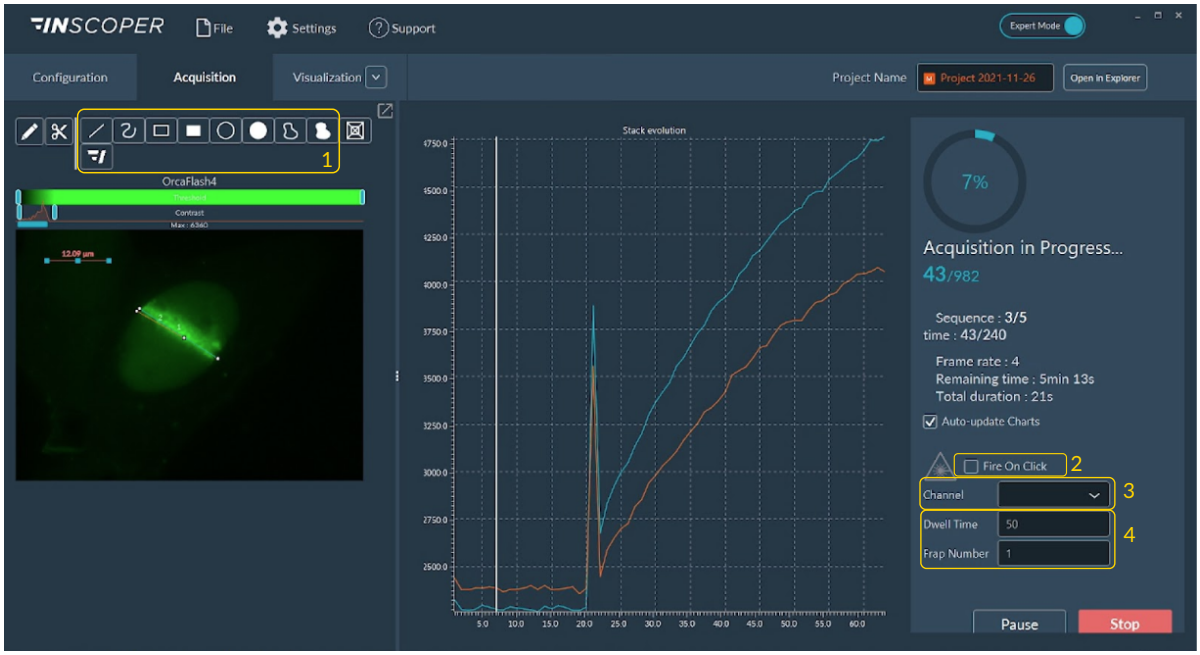
All rights reserved.

5. Adjust the total **pulse time** (the time required for the galvanometers to move from the previous coordinates plus the programmed exposure dwell time).
6. Define the total number of consecutive exposure iterations.
7. Adjust the laser output power (percentage).
8. Click **Confirm** to save the sequence parameters.

> **Note**
>
> Before initiating a complex photomanipulation experiment, empirically ensure that the scanFRAP galvo module is acutely calibrated.

Live Acquisition Overrides

During an actively running time-lapse acquisition sequence, you can monitor experimental progress in real-time, reviewing both live images and plotted raw intensity data for each designated ROI. The unique **Fire On Click** feature allows you to selectively photomanipulate entirely new, spontaneous areas on the fly without interrupting the sequence timeline.



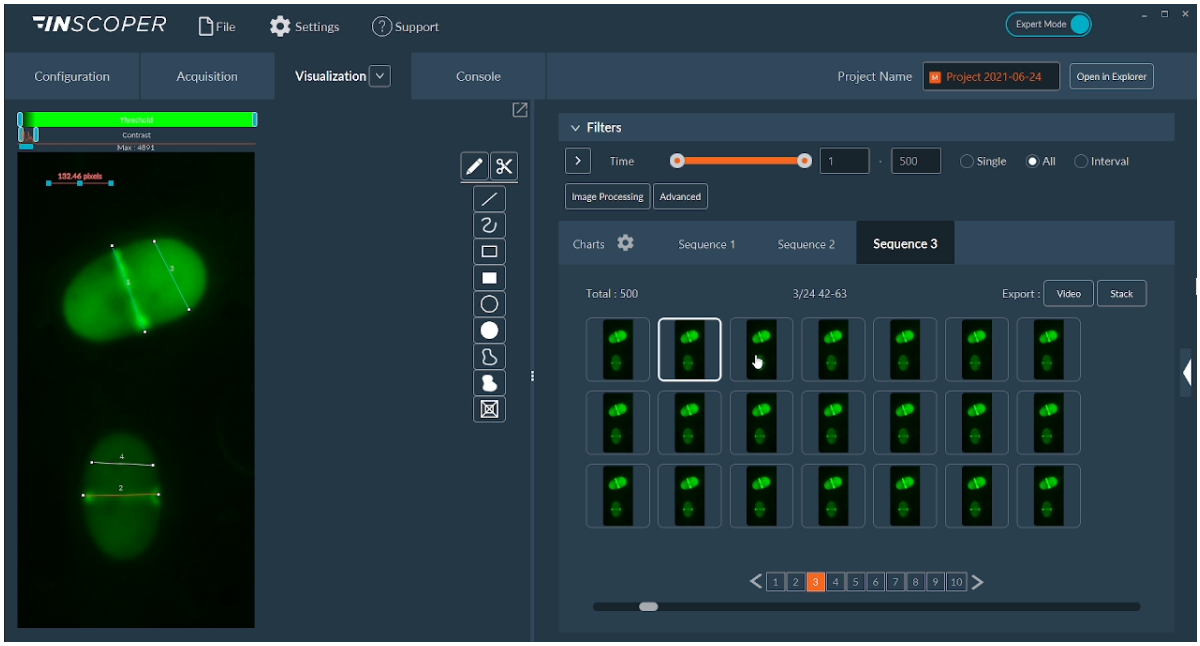
To deploy **Fire On Click**:

1. Draw a new target ROI on the live, incoming image stream.
2. Check the **Fire On Click** enablement box.
3. Verify the accurately calibrated manipulation channel is actively selected.
4. Adjust the temporal **Dwell Time** (pulse duration per pixel) and the **FRAP number** (total physical iterations).
5. Click directly on the newly active ROI in the viewer to immediately demand and execute the photomanipulation sweep.

Data Visualization

The **Visualization** tab displays the sequence imaging results alongside graphs tracking the fluorescence intensity over time within all defined ROIs.

All rights reserved.



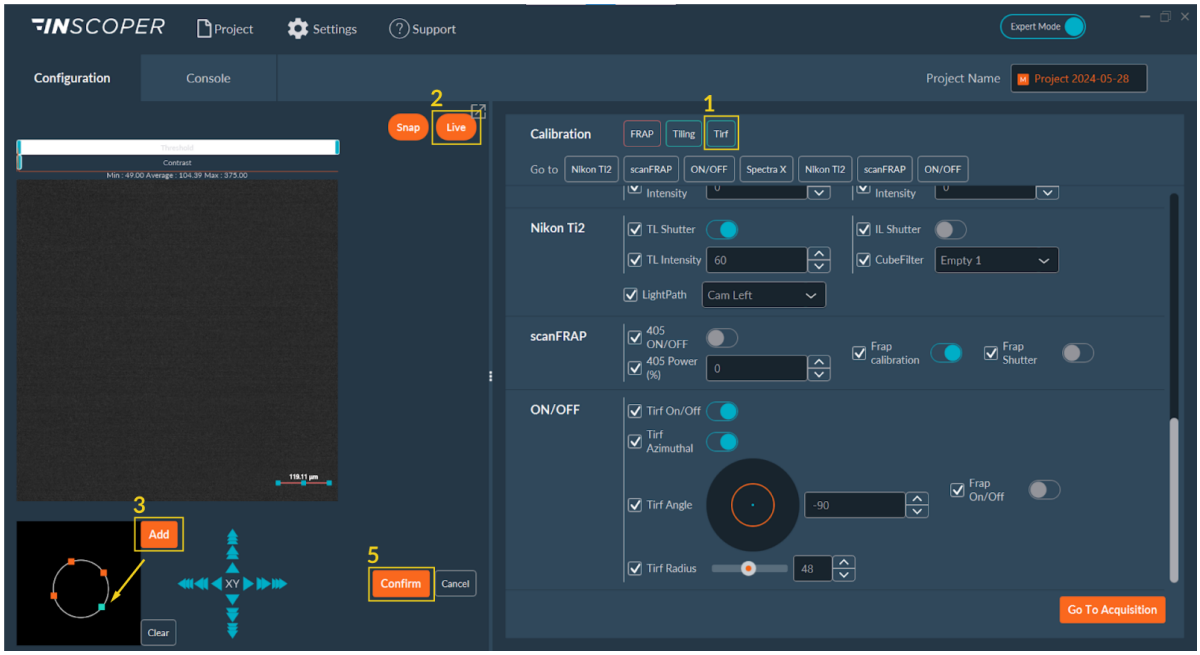
2026-09-08

All rights reserved.

1.6.3 TIRF

Total Internal Reflection Fluorescence

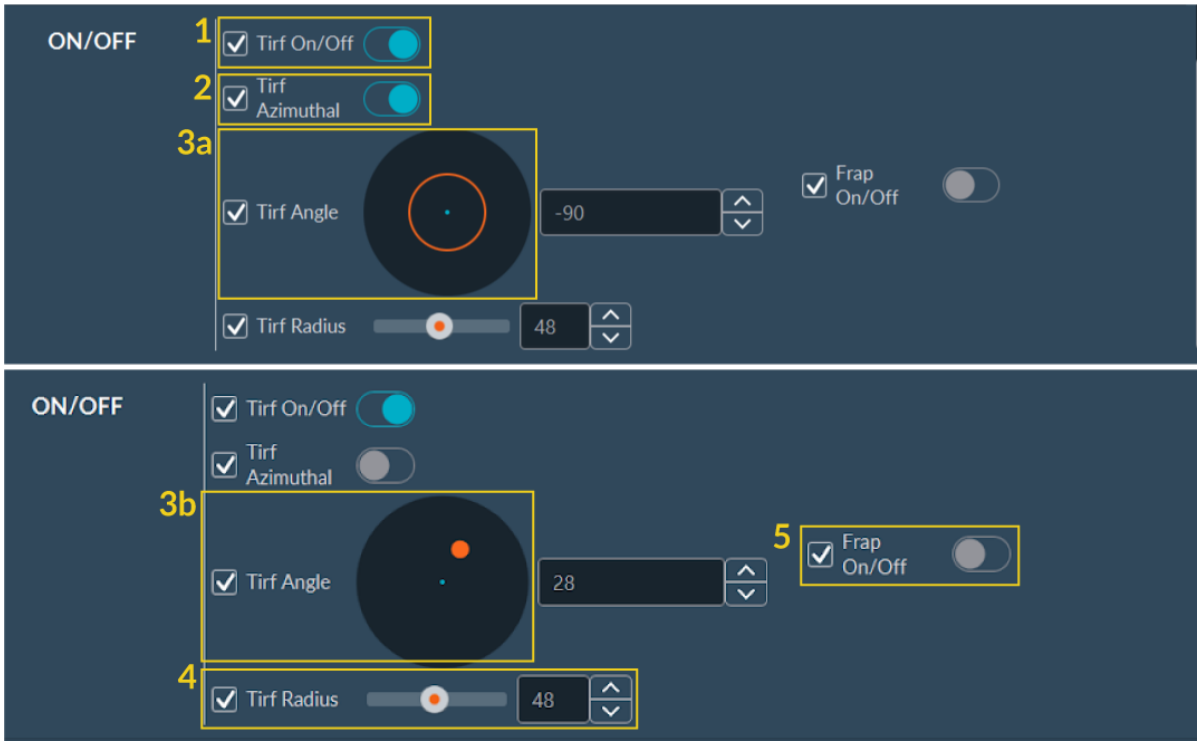
Optical Calibration



1. Navigate to the Calibration section and click **TIRF**.
2. Enable the **Live** camera feed to observe the sample.
3. Click **Add** to insert a calibration point into the schematic field of view. This blue control point corresponds to the TIRF laser position. Drag this point radially to adjust the laser incidence angle until you achieve critical angle TIRF illumination (evidenced by the loss of out-of-focus background fluorescence).
4. Repeat the coordinate placement process to define additional calibration positions if needed for different sample topologies or wavelengths.
5. Click **Confirm** to save the spatial calibration matrix.
6. Repeat this angular calibration protocol for each discrete laser wavelength intended for TIRF imaging, as critical angles vary with excitation wavelength.

All rights reserved.

Configuration



1. **Activate TIRF**: Toggle the switch to enable or disable TIRF mode hardware control.
2. **TIRF Azimuthal**: Toggle **TIRF Azimuthal** to enable circular illumination, minimizing interference fringes and uneven illumination.
3. **Option**:
    - a. Activated: The interface displays a diagram of the circular laser scanning pattern at the currently set TIRF angle.
    - b. Deactivated: The interface displays a single laser position (orange dot). Specify its coordinates using the **TIRF Angle** parameter field.
4. **TIRF Radius**: Adjust the spatial radius to set the laser incidence angle at the objective lens aperture. This control determines the illumination regime: Widefield (Epi), Highly Inclined and Laminated Optical Sheet (HILO), or TIRF.
5. **FRAP**: Toggle the corresponding switch to activate the targeted photobleaching (FRAP) function.

CREATING A TIRF CHANNEL

Save the spatial parameters (**Azimuthal mode**, **Angle**, and **Radius**) to the designated TIRF channel preset in the Configuration tab to properly execute automated TIRF imaging.

Sequence Acquisition

Select the saved TIRF channel to perform a TIRF acquisition during the multi-dimensional sequence.

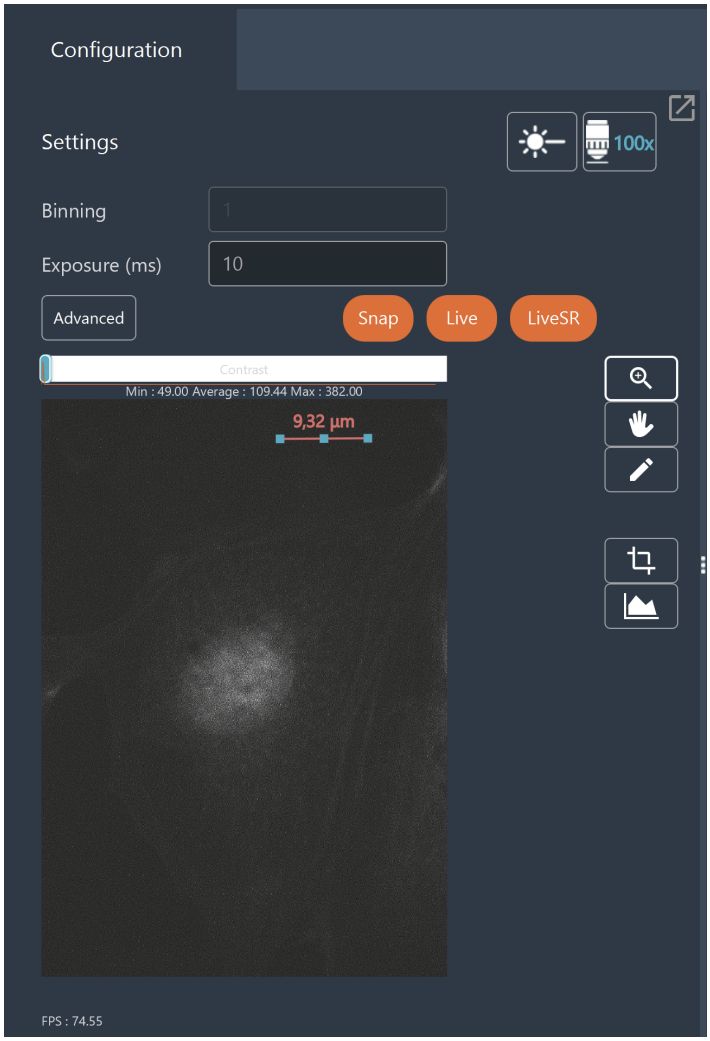
2026-09-08

All rights reserved.

1.6.4 LiveSR

LiveSR Real-Time Preview

In addition to the standard Live and SNAP buttons, use the LiveSR button to preview LiveSR processing in real time.



Configuring LiveSR Parameters

In the acquisition tab :

- Configure your acquisition sequence.
- Select the **DataProcessor** that will perform the LiveSR processing.
- Click the **Edit Parameters** button, to configure the following settings:
  - **Online**:
    - If Online is not checked, a smoothing step is applied at the end of the algorithm.
    - If it is checked, this step is skipped for faster process. In Live (LiveSR), the smoothing step is skipped.

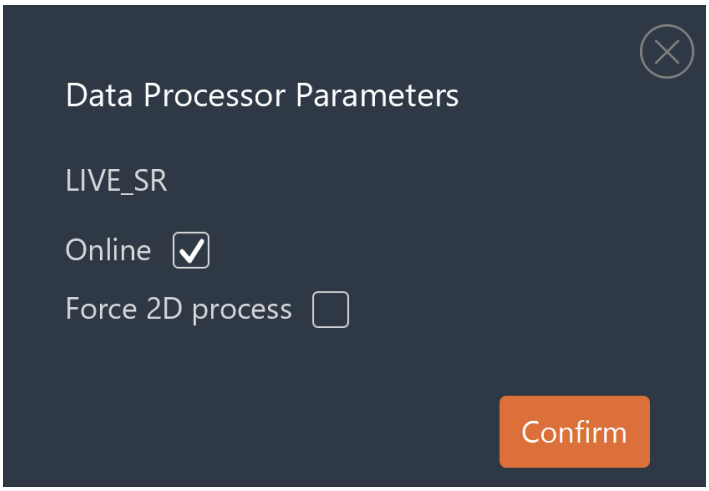
> **Note**
>
> The smoothing is a low-pass filter to remove any noise that may be introduced by deconvolution

  - **Force 2D process**:
    - By default, the system applies 2D processing for single frames and automatically switches to 3D processing during Z-stack acquisitions.
    - Check **Force 2D process** to enforce 2D processing across all frames, even during a Z-stack acquisition.

All rights reserved.

> **Note**
>
> - **Maximum Z-Step**: The Z-step size must be **less than 0.4 µm** when running 3D processing. Exceeding this threshold will trigger an error message.
> - **Bypassing the Limit**: To acquire a Z-stack with a step size greater than 0.4 µm, you must enable the **Force 2D process** option.

- Click **Confirm** to apply your parameters.



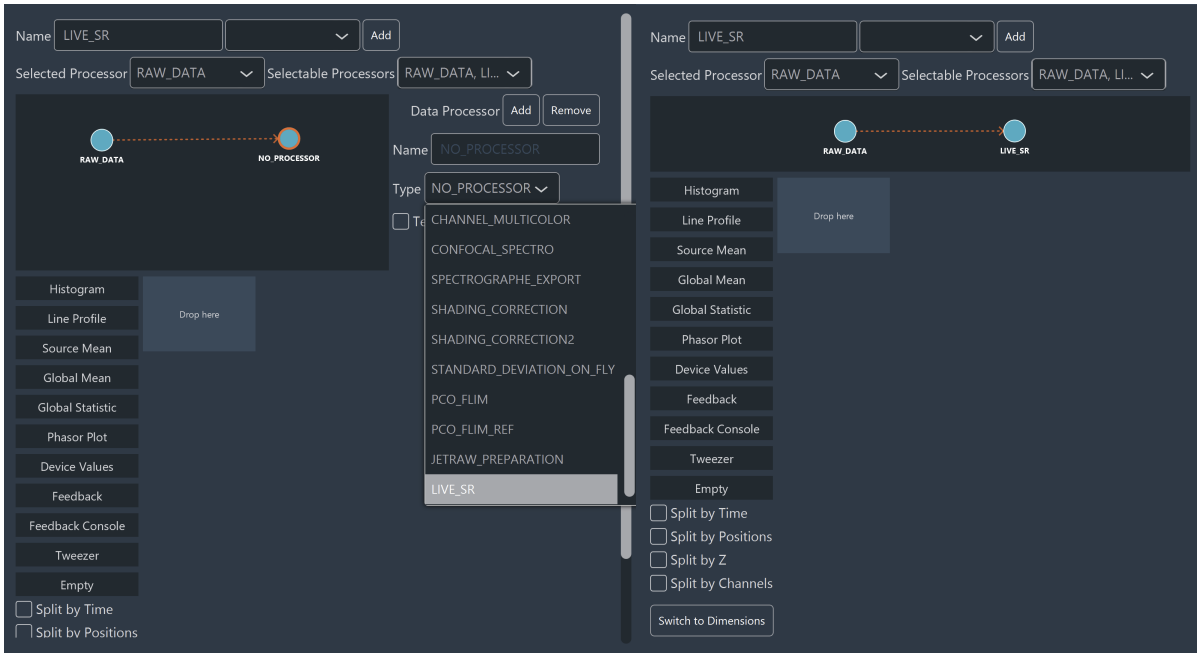
- Click **Start Acquisition** to run the experiment.

Creating a LiveSR DataProcessor

To construct a custom LiveSR processing pipeline:

- Enable **Expert Mode**.
- Click **Switch to DataProcessor**.
- Add Processing Nodes:
  - Right-click the **RawData node** and select **New Node**.
  - Double-click the new node to open its properties.
  - In the right-hand panel, select the **LIVE_SR process**.
- Configure Display Settings:
  - **Selected Processor**: Select the process from the drop-down menu that will be displayed in real time **during acquisition**.
  - **Selectable Processors**: Select the process(es) that will be accessible in the **Visualization tab** after acquisition completes.
- Enter a name for the custom DataProcessor, then click **Add** to save it.

All rights reserved.



2026-09-08

All rights reserved.

1.6.5 liveRATIO

The **liveRATIO** add-on enables real-time ratiometric imaging, allowing you to quantify the ratio between two emission wavelengths over time (e.g., FRET, calcium imaging). The `Ratiometric imaging` data processor drives this function.



Hardware Configuration

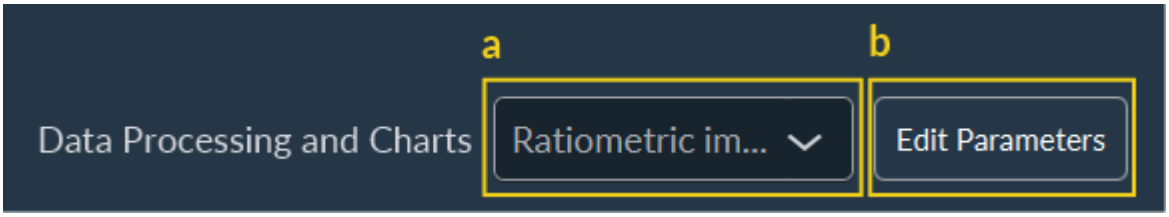
For physical hardware and baseline channel setup, please refer to the standard Configuration section.

Sequence Acquisition

1. WAVELENGTH SETUP

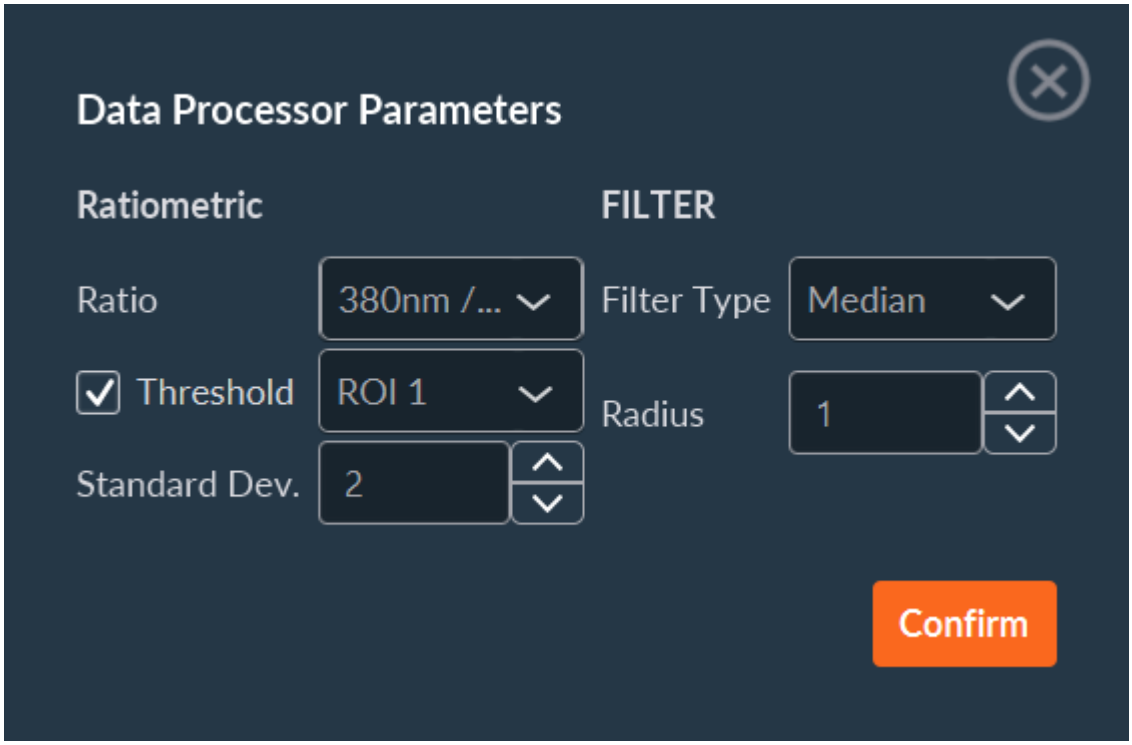
1. Navigate to the **Multi-Channels** dimension within the **Acquisition** tab.
2. Configure two or more wavelengths in the sequence.

2. DATA PROCESSING CONFIGURATION



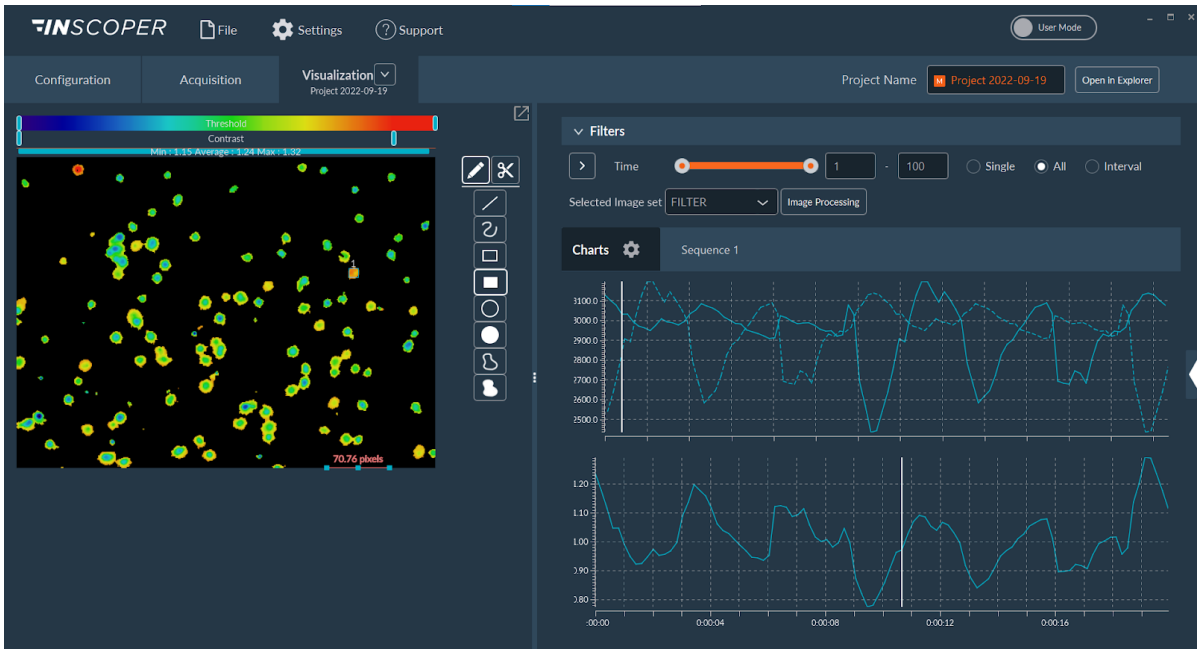
- a. Choose **Ratiometric imaging_Despeckle** from the **Data processing and Chart** drop-down menu.
- b. Click **Edit Parameters** to open the module's settings.
- c. Specify which channels form the **numerator** and **denominator** for your calculated ratio.
- d. (Optional) Enable background noise removal:
  - If enabled, draw a Region of Interest (ROI) in an area devoid of signal. This ROI serves as the background reference.
  - Specify the standard deviation multiplier for the background subtraction threshold calculation.
- e. Select the spatial filtering method (median or average filter).
- f. Define the radius (in pixels) for the spatial filter.
- g. Click **Confirm** to save your processing parameters.

All rights reserved.



Data Visualization

The **Visualization** tab displays both the raw sensor intensity data and the calculated ratiometric data for any drawn ROIs. These data plots update in real-time whenever an ROI is created, adjusted, or removed.



2026-09-08

All rights reserved.

1.6.6 Confocal MAICO

The Confocal MAICO provides a solution for laser scanning confocal microscopy, designed for integration with microscopes from any manufacturer. It is used for routine fluorescence imaging, functioning as a dedicated benchtop system or as a primary instrument in a core facility.

System Startup

Before launching the Inscoper Software, you must **turn on and calibrate the confocal MEMS unit**:

1. Turn the physical Power key to **On**. The system LED transitions from solid orange to a blinking green light.
2. Wait 5 to 10 minutes for the unit to reach thermal stabilization. The **Laser Calibration** button in the startup interface turns green once the system is ready.
3. Press the **Laser Calibration** button to initiate the automated hardware calibration.

> **Important**
>
> You must open the microscope's optical shutter for this step to succeed.

4. The green calibration light turns off upon successful completion.
5. You may now safely launch the main Inscoper Software.

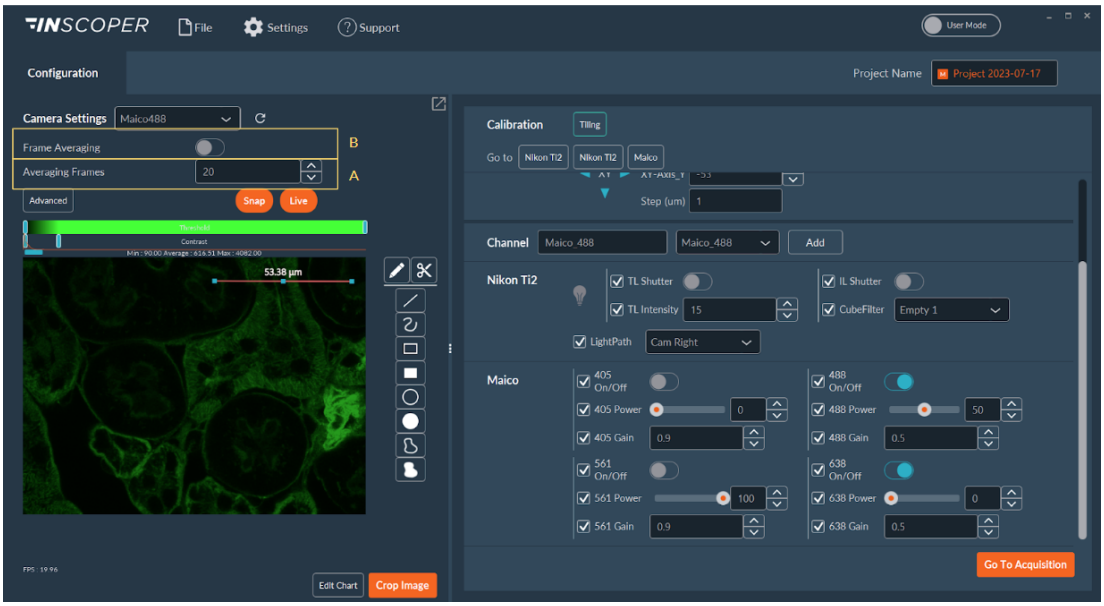
Channel Configuration

The **Configuration** tab provides direct access to the laser and detector hardware parameters, allowing you to finely tune laser excitation power and detector amplification (gain).

- Click **Add** to apply parameter changes temporarily for the current live experiment session.
- Switch to **Expert Mode** and click **Save** to create a new, reusable channel preset or to store your current settings permanently to the database.

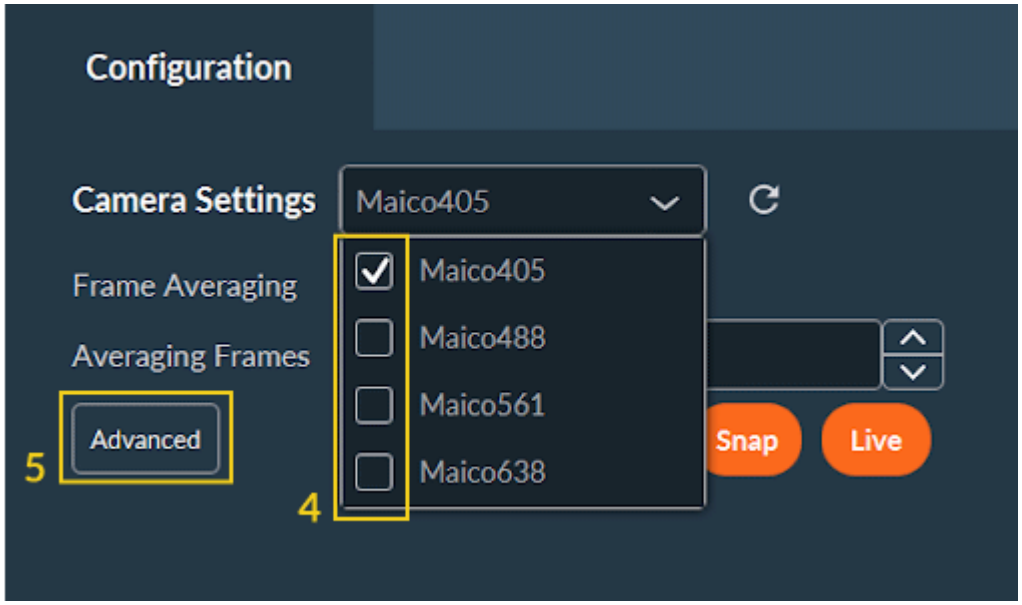
SETTING UP A MAICO CHANNEL

1. For multicolor imaging sequences, create a **single channel preset that encompasses all required laser lines**, even if you perform sequential (line-by-line) acquisition later.
2. Adjust the excitation intensity (% power) and detector gain for each wavelength.
3. **Frame Averaging**: Enable frame averaging across all active detectors to improve the image Signal-to-Noise Ratio (SNR):
    1. Set the number of consecutive frames to average.
    2. Toggle the averaging feature on.

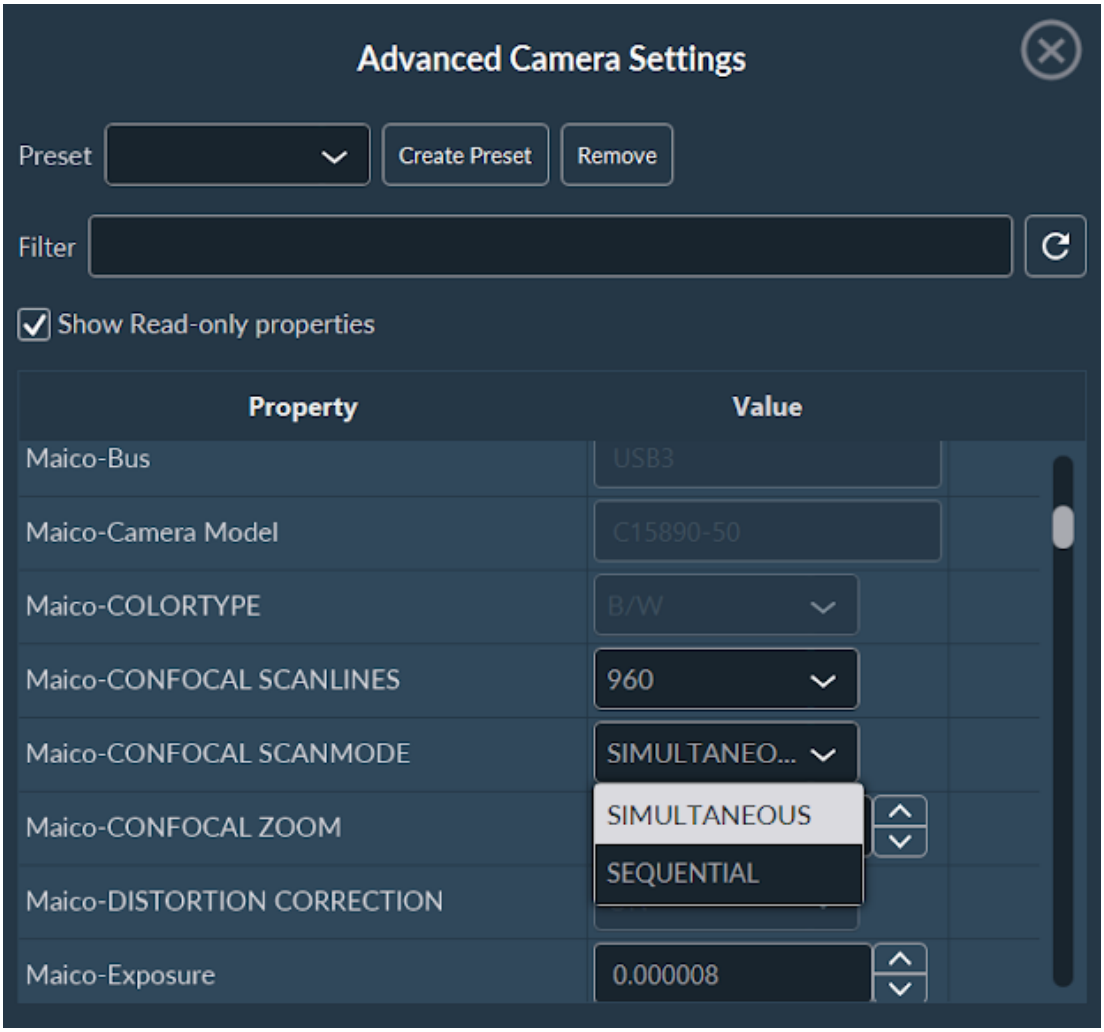


4. **Detection View**: Select the detection channels you want to monitor by checking their respective boxes in the interface.

All rights reserved.



5. **Advanced Options**: Access detailed scanning hardware settings by clicking **Advanced**:
    - **Scan line**: Select the number of vertical scan lines (960, 480, or 240). Lower line counts significantly increase the frame rate but reduce spatial resolution.
    - **Scan Mode**: Choose between Sequential (minimizes spectral bleed-through) or Simultaneous (maximizes acquisition speed) excitation.
    - **Zoom**: Choose between optical Zoom 1 or 2.



6. Click **Go to Acquisition** to transition to the sequence builder.

Sequence Acquisition

Configure your experimental sequences in the **Acquisition** tab, following the standard procedures outlined in the global User Guide.

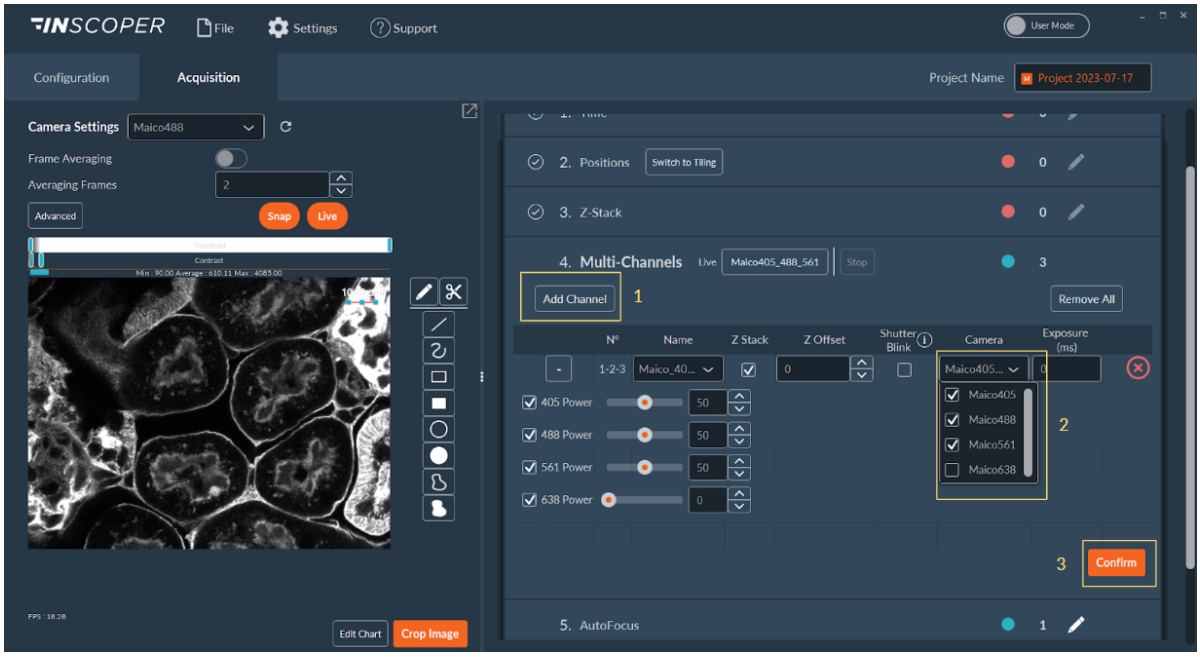
All rights reserved.

Warning

For MAICO confocal acquisitions, the only procedural deviation from standard widefield imaging occurs within the **Multi-Channels** dimension setup.

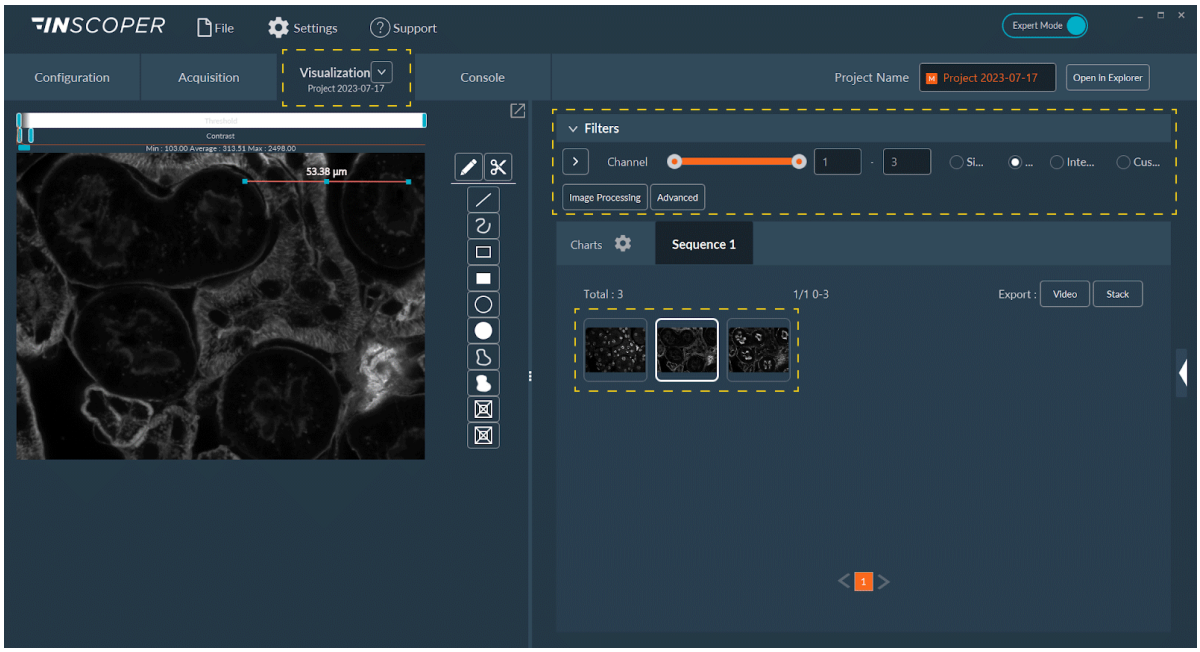
THE MULTI-CHANNELS DIMENSION

1. Click **Add Channel** to insert your previously configured multi-laser MAICO channel preset into the sequence.
2. Check the box corresponding to your desired confocal detection mode.
3. Click **Confirm** to route the configuration.



Data Visualization

Immediately following the acquisition, the **Visualization Tab** becomes active. You can browse, filter, and review your confocal image stacks by their assigned dimensions (e.g., Z, T, C).



Tip

If your system includes an active Microvolution license, you can perform advanced 3D deconvolution directly within Inscoper I.S. by clicking the **Image Processing** button.

All rights reserved.

2026-09-08

All rights reserved.

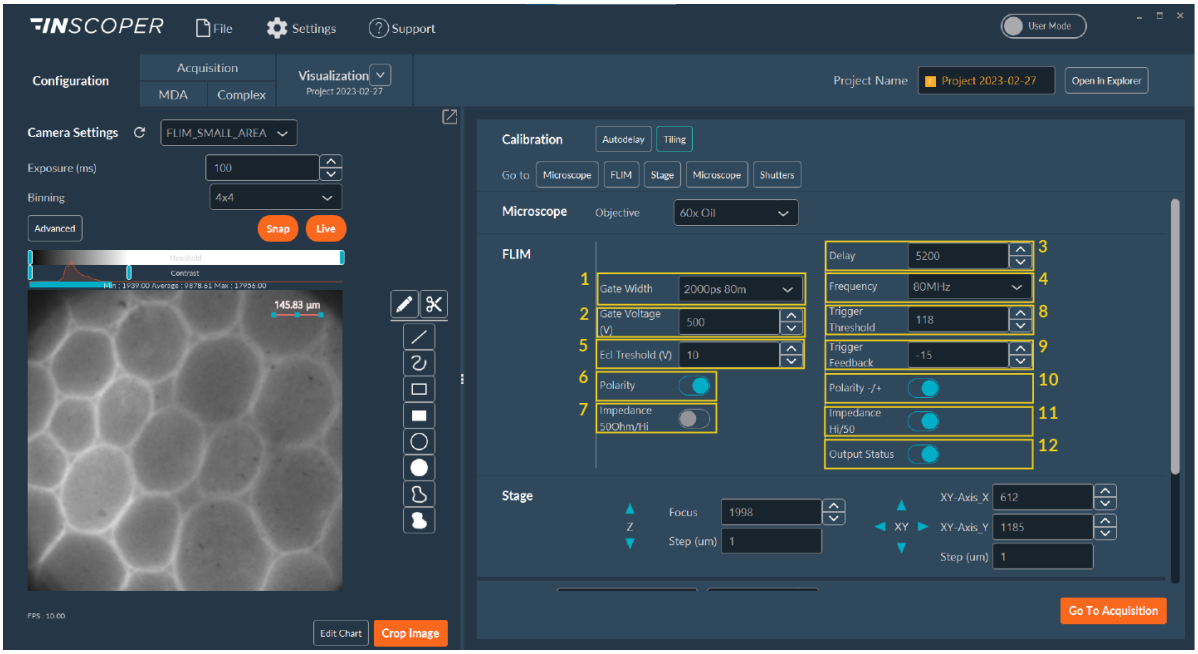
1.6.7 Time Gated Domain FLIM (fastFLIM)

fastFLIM is a hardware and software solution designed to measure mean fluorescence lifetime using a widefield, camera-based microscope architecture. fastFLIM measurements combine with the other imaging dimensions, including Multi-positions, Tiling mosaics, Multi-channel overlays, and Z-stacks.

Unlike traditional point-scanning FLIM (e.g., TCSPC), which is typically slow, **fastFLIM** is a **camera-based FLIM technique** that employs a time-domain approach combined with a time-gated image intensifier. This captures the lifetime of the entire field of view simultaneously (or rapidly via spinning disk), providing significantly higher acquisition speeds and minimal phototoxicity. This makes it highly effective for observing rapid biological dynamics and biosensors in living cells.

Hardware Configuration

The Configuration tab gives access to the FLIM timing and gating parameters:



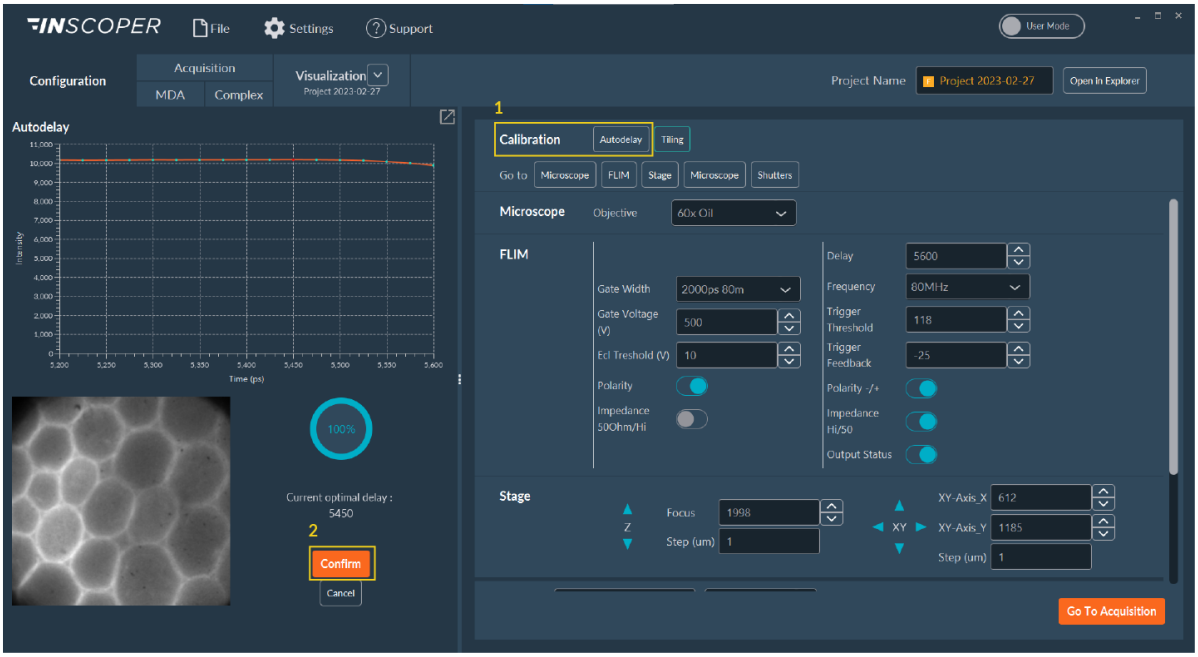
1. **Gate Width**: Specifies the absolute duration of the temporal detection window (gate).
2. **Gate Voltage**: Sets the multi-channel Plate (MCP) potential voltage, which directly controls the internal signal amplification (gain) of the intensifier.
3. **Delay**: The temporal offset (in picoseconds) applied relative to the pulsed laser synchronization signal.
4. **Frequency**: The operational repetition rate of the central delay generator.
5. **ECL Threshold**: Sets the electrical noise floor threshold for the input synchronization signal routed to the intensifier.
6. **Polarity**: Configures the detector logic to correctly identify the leading or trailing edge of the delay generator's synchronization pulse.
7. **Impedance**: Adjusts electrical impedance matching to ensure artifact-free signal detection.
8. **Trigger Threshold**: Sets the electrical noise floor threshold for the input synchronization signal routed to the delay generator.
9. **Trigger Feedback**: (Specific to legacy delay generator models). Displays a diagnostic value verifying lock synchronization between the delay generator and the pulsed laser. Optimal operational values hover near 0, safely within a [-100, +100] operational range.
10. **Polarity (-/+)**: Configures the delay generator logic to correctly detect the laser's physical synchronization pulse.
11. **Impedance (Hi/50Ω)**: Adjusts electrical impedance matching specifically for the laser synchronization line.
12. **Output Status**: Manual override for the optical shutter control.

All rights reserved.

> **Note**
>
> The visibility and editability of specific parameters adapt dynamically based on the specific intensifier and gating hardware model recognized by the system.

AUTODELAY PULSE CALIBRATION

> **Important**
>
> Before initiating a FLIM experimental sequence, execute an **Autodelay** calibration. This step synchronizes the pulsed laser excitation with the initial temporal gate opening, ensuring accurate photon capture and reliable lifetime calculations.



1. Click **Autodelay** to launch the automated hardware calibration protocol.
2. Wait for the software algorithm to calculate the optimal temporal offset. Once displayed, click **Confirm**. This calibrated picosecond value will automatically populate the master **Delay** field in the Configuration tab.

Sequence Acquisition

With the hardware configured and calibrated, proceed to the Acquisition tab to build the imaging sequence.

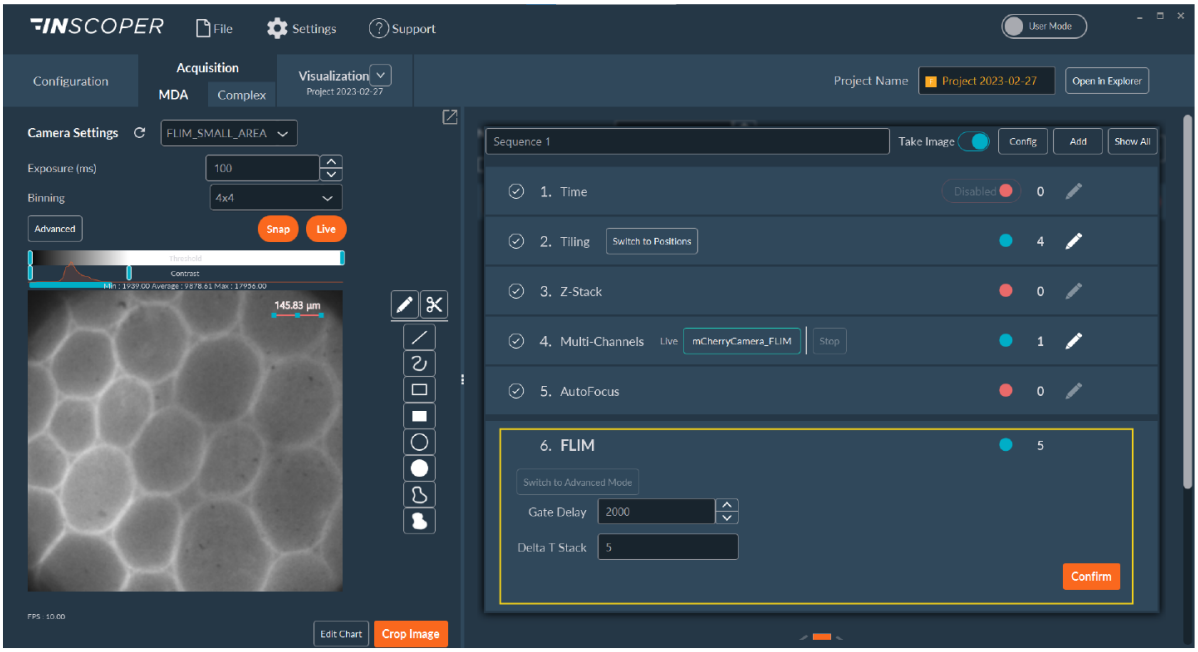
Configure the imaging channels in the Multi-channel dimension. Define both the widefield fluorescence channels and the FLIM measurement channels.

In the **FLIM dimension** panel, the available controls depend on the number of active FLIM channels:

SIMPLE MODE

This mode activates automatically when only **a single FLIM channel** is present in the sequence.

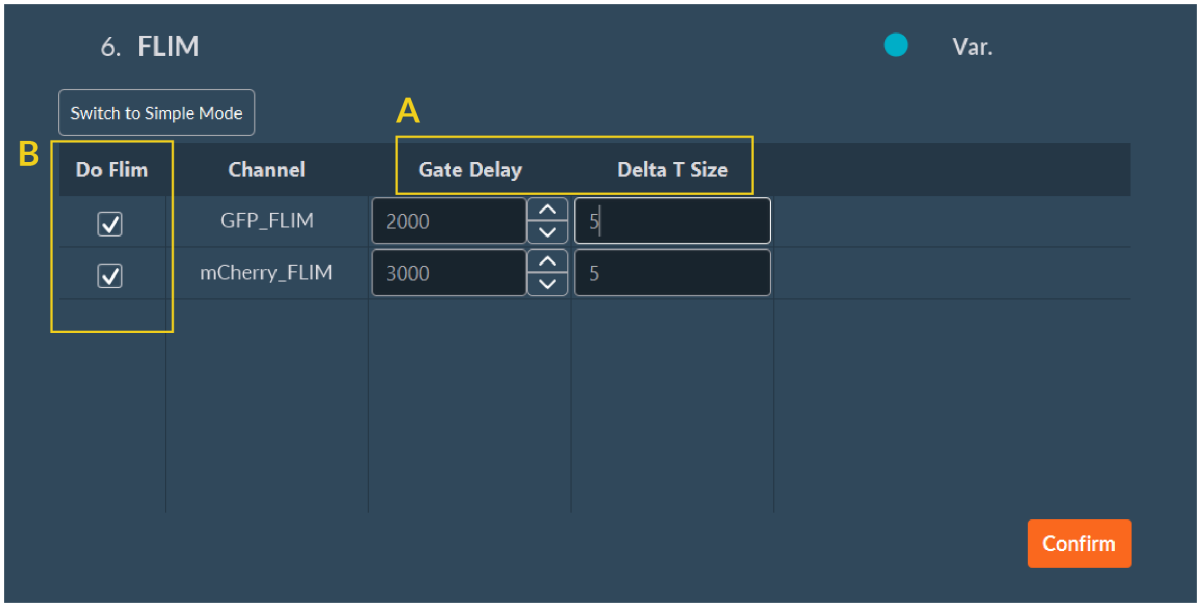
All rights reserved.



- **Gate Delay**: The temporal shift between successive measurement gates along the decay curve.
- **Delta T stack**: The number of time points to acquire for the complete FLIM decay curve.

ADVANCED MODE

This mode activates automatically when **multiple FLIM channels** are configured in the sequence. It exposes per-channel timing parameters.

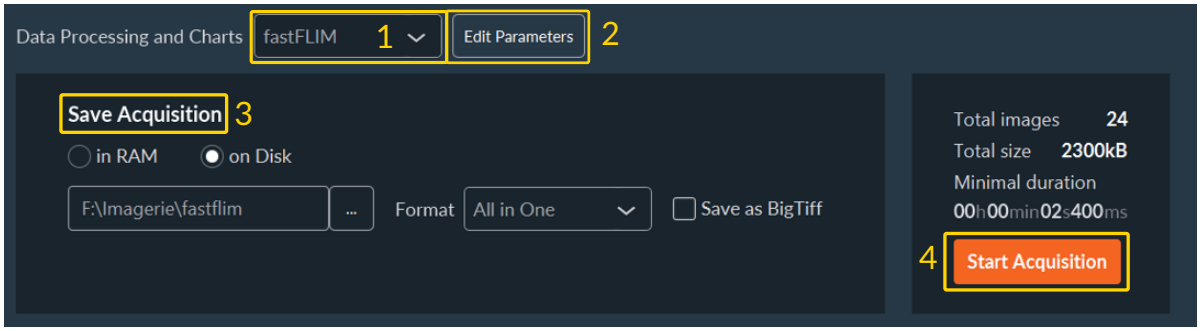


- **A**: Define an independent **Gate Delay** and **Delta T Size** for **each channel**.
- **B**: Enable or disable the FLIM measurement per channel with the **Do Flim** checkboxes.

FINALIZING ACQUISITION

Once the FLIM timing parameters are set, click **Confirm** on the dimension module. To complete the data routing:

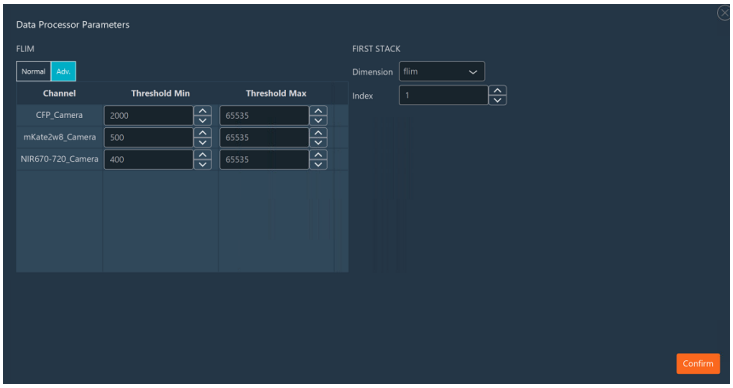
All rights reserved.



1. Navigate to the Save menu. Select **fastFLIM** from the Data Processing and Charts computation drop-down menu.
2. Click **Edit Parameters** to define the absolute minimum and maximum **Intensity Thresholds** (photon counts) required for valid pixel-wise FLIM calculation.
    - **Normal** : Define the absolute minimum and maximum **Intensity Thresholds** (photon counts) required for valid pixel-wise FLIM calculation.



    - **Advanced** : Define by channel different minimum and maximum **Intensity Thresholds** (photon counts) required for valid pixel-wise FLIM calculation.

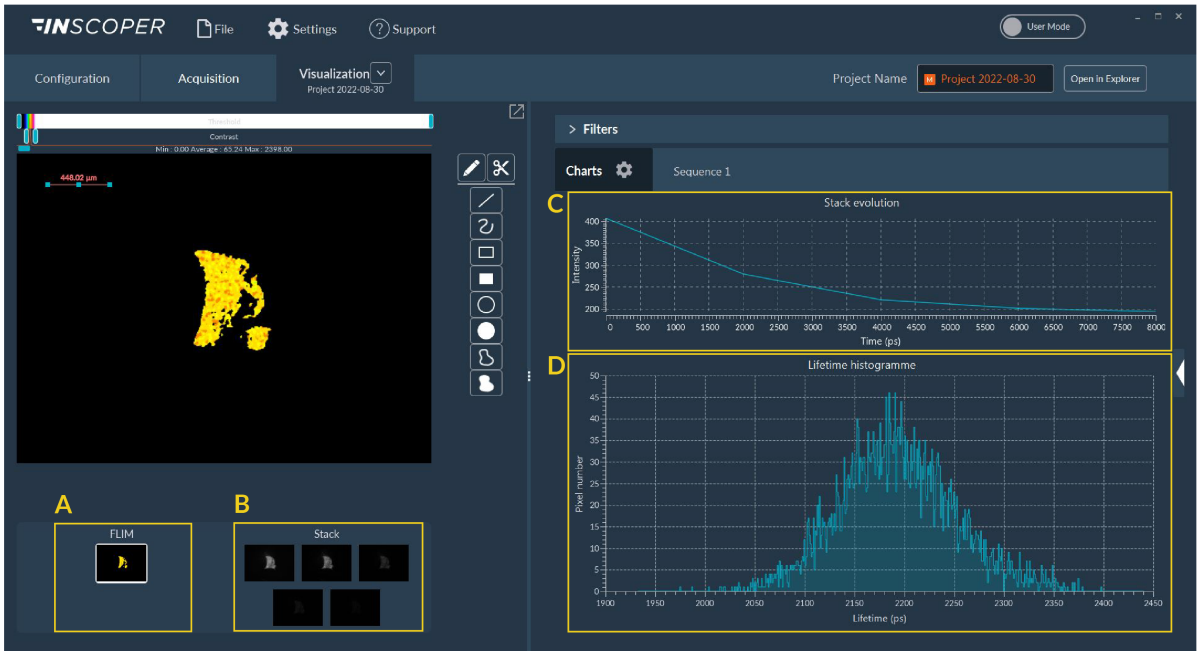


    - Click **Confirm**. (Pixels below this threshold are excluded from the decay fit).
3. Specify your absolute destination save folder path.
4. Click **Start Acquisition** to initiate the sequence.

Data Visualization

The Visualization tab provides FLIM analytical results computed at the end of the acquisition. It displays calculated FLIM maps alongside raw sensor intensity, integrated lifetime distributions, and point-wise intensity decay curves.

All rights reserved.



- **A**: The pseudo-colored, computed FLIM image map.
- **B**: The time-gated intensity images used for the decay calculation.
- **C**: The plotted intensity decay curve for the selected pixel or ROI.
- **D**: The global pixel lifetime distribution histogram.

All generated multi-dimensional images and statistical charts are saved automatically to your specified destination folder upon compute completion.

2026-09-08

All rights reserved.

1.7 How-to guides

1.7.1 Use the Nikon Ti2 Water Dispenser

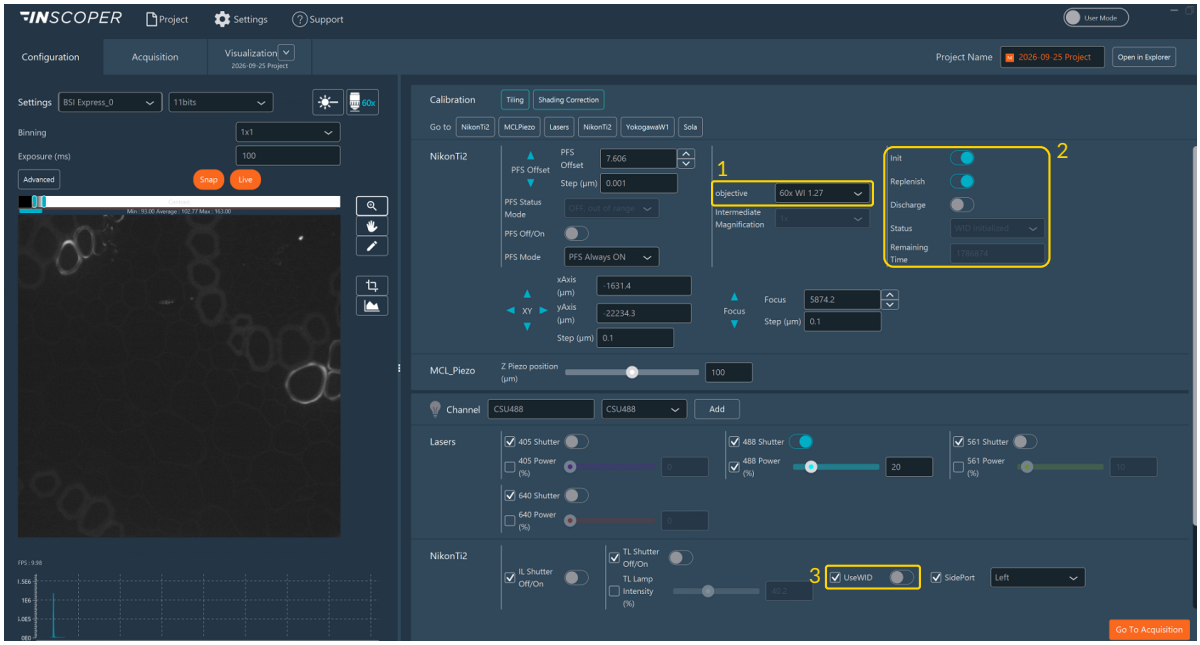
The **Water Immersion Dispenser** (WID) of the Nikon Ti2 delivers water between a water-immersion objective and the sample, and removes it afterwards. Immersion water evaporates during imaging, so a long acquisition requires the water to be refreshed at regular intervals. This page explains how to fill the circuit, refresh it during an acquisition, and empty it at the end of the session.

Before you start

- The WID must be declared in the Configurator, as the Water dispenser module of the Nikon Ti2e. If its controls are missing from the **Configuration** tab, contact the instrument administrator.
- A water-immersion objective compatible with the WID must be mounted on the objective turret. The WID refuses to start with any other objective.
- Protect the objective, the sample, and the surrounding area from accidental water.

The WID controls

The WID controls are displayed in the **NikonTi2** panel of the **Configuration** tab.



1. **objective**: the objective in the light path, here `60x WI 1.27`.
2. **WID controls**: the **Init**, **Replenish**, and **Discharge** switches, which start the three operations; the **Status** of the WID; and the **Remaining Time** before the next recommended refresh.
3. **UseWID**: a virtual parameter that exists only to be saved in a channel. A channel with **UseWID** switched on refreshes the water each time the acquisition runs it.

A switch that cannot be used in the current state of the WID is disabled. This is a safety restriction, not a communication failure. The Reference section at the bottom of this page lists every control, status value, and duration.

Fill the circuit

1. Select the water-immersion objective in **objective** (1).
2. Stop any acquisition and any movement of the stage or of the focus.
3. Switch **Init** (2) on.
4. Wait until **Status** shows WID Initialized and **Remaining Time** starts counting down. **Init** stays on while the WID is initialized.

Pumping takes about 3 minutes. The objective then moves to a service position while the surplus water is removed, and returns to its focus position.

All rights reserved.

Do not start **Init** again while the WID is initialized. Use **Replenish** or **Discharge** instead.

Refresh the water

Switch **Replenish** (2) on. A refresh removes air bubbles, delivers fresh water, and removes the surplus. During the cycle, the objective leaves its focus position and returns to it at the end, so do not acquire images until **Status** shows WID Initialized again.

Remaining Time restarts at 30 minutes after each successful refresh. It is a reminder computed by Inscoper, not a measurement of the water left:

- When it reaches `0`, no refresh starts automatically. The WID stays initialized until the next **Replenish**, whether started by hand or by the acquisition.
- At `0`, water can still remain on the objective. Refreshing every 30 minutes at most is the recommended default, but Inscoper I.S. does not enforce it: after checking the water on the objective, a longer interval, for example 45 minutes, can be used.

Refresh the water during an acquisition

Fill the circuit before starting the acquisition: an acquisition only refreshes a circuit that is already initialized. Then choose the approach from the duration of the acquisition, since one refresh is recommended for about 30 minutes of imaging:

Duration of the acquisition	Approach
Shorter than 30 minutes	Switch **Replenish** (2) on, wait for WID Initialized, and start a single imaging sequence.
Longer than 30 minutes	Run a multi-sequence acquisition in which a dedicated sequence refreshes the water at regular intervals, as described below.

CREATE THE WATER DISPENSER CHANNEL

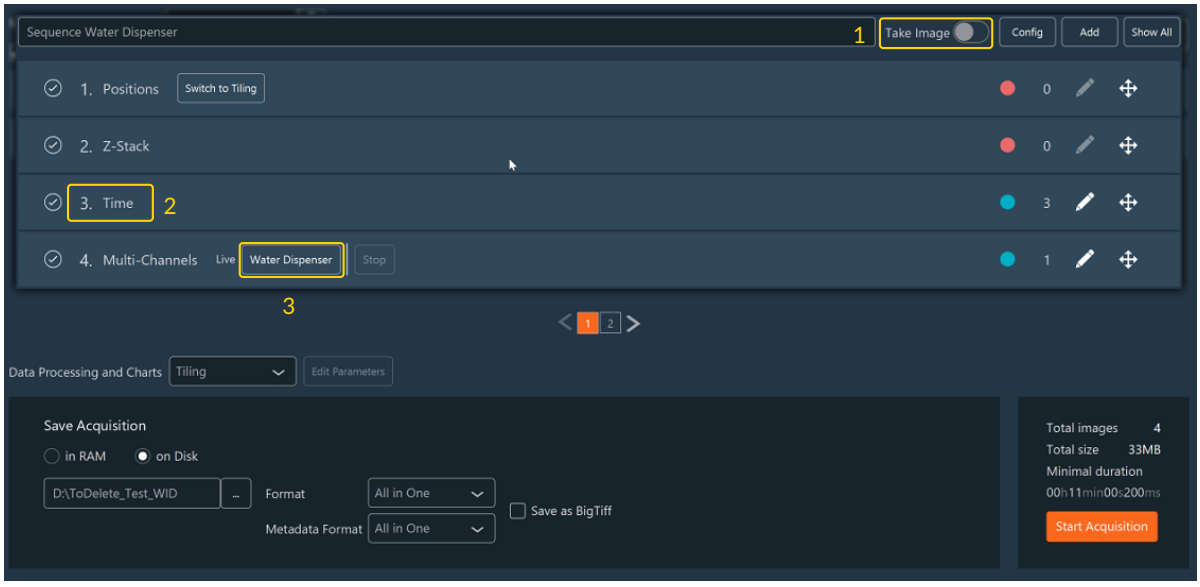
In the **Configuration** tab, create a channel dedicated to the water, as described in Creating and Loading Channels:

1. Check the box beside **UseWID** (3), and switch the parameter on.
2. Name the channel, for example `Water Dispenser`.
3. Save the channel.

UseWID must be switched off in every imaging channel. An imaging channel with **UseWID** switched on would start a refresh, and move the objective, each time it is acquired.

BUILD THE WATER DISPENSER SEQUENCE

In the **Acquisition** tab, add a sequence dedicated to the water, and name it, for example `Sequence Water Dispenser`.

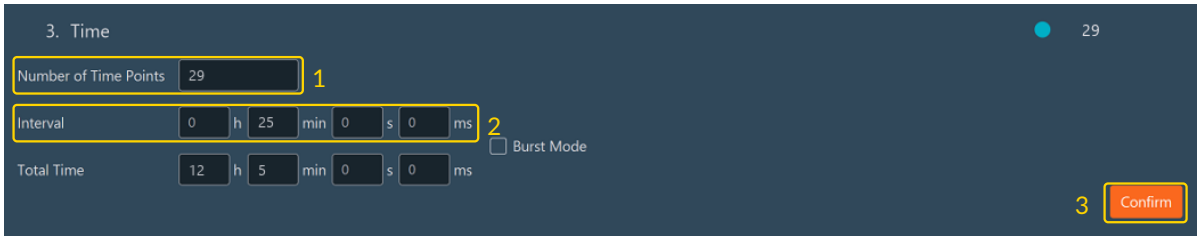


1. Disable **Take Image** (1). The sequence refreshes the water and saves no image.

All rights reserved.

2. Activate the **Time** (2) dimension. Its time points set when the water is refreshed.
3. In the **Multi-Channels** dimension, select the **Water Dispenser** (3) channel as the only channel of the sequence.

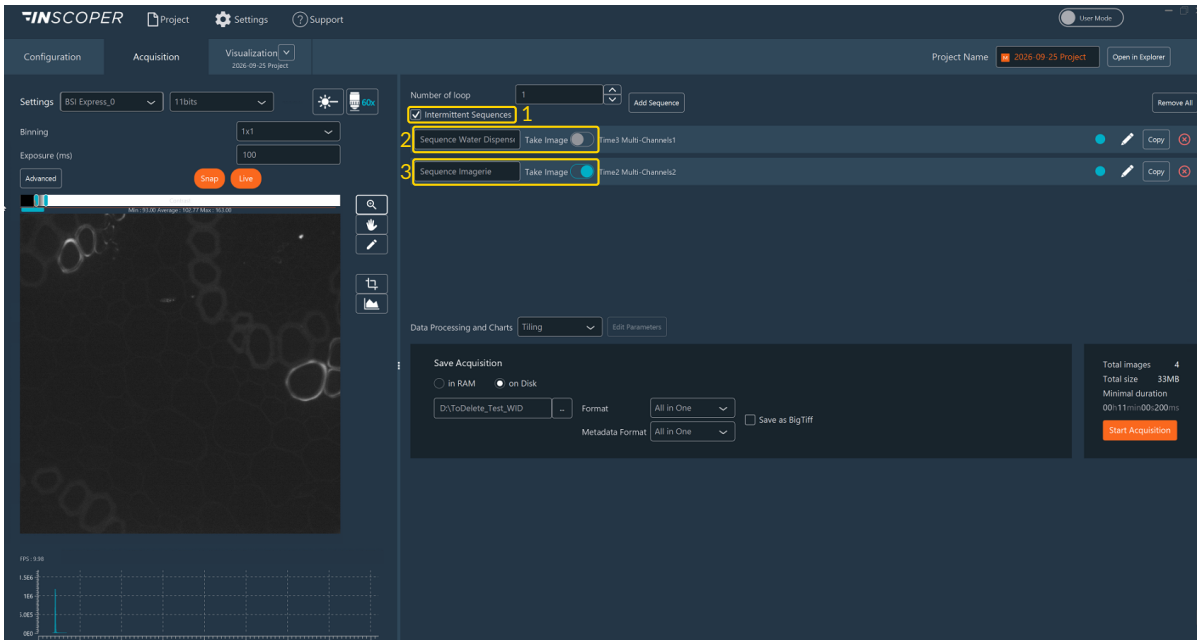
Then open the **Time** dimension to set the refresh interval.



1. Enter the **Number of Time Points** (1), which is the number of refreshes.
2. Enter the **Interval** (2) between two refreshes. An interval shorter than 30 minutes keeps **Remaining Time** above `0`; a longer interval is accepted, as described in Refresh the water.
3. Click **Confirm** (3).

The figure shows a 12-hour experiment: 29 refreshes, 25 minutes apart, for a **Total Time** of 12 h 5 min. Choose the number of time points so that the refreshes cover the whole duration of the imaging sequence.

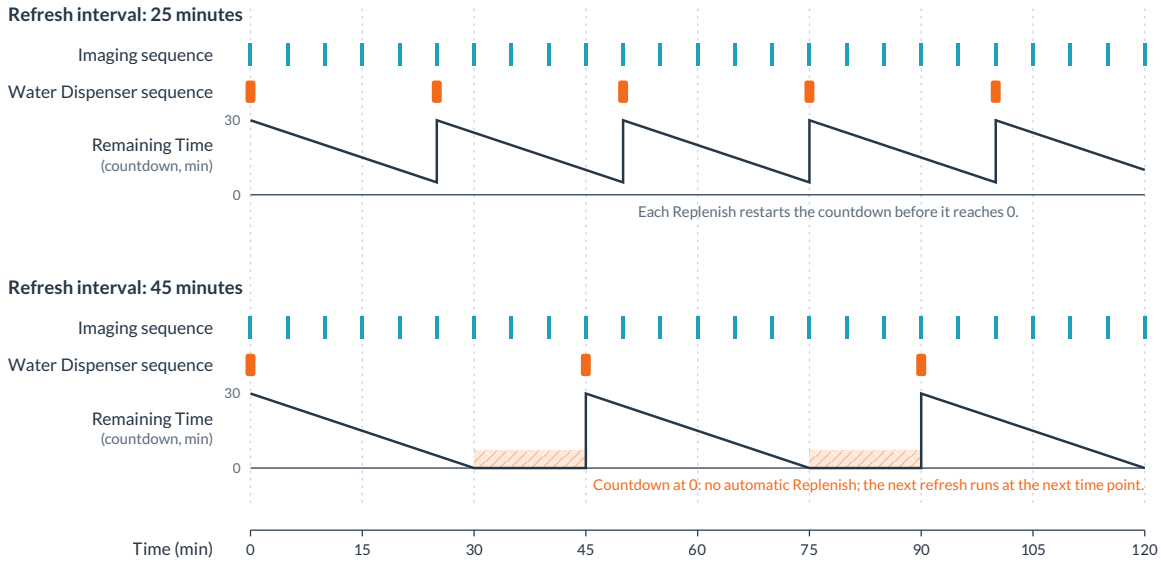
RUN IT WITH THE IMAGING SEQUENCE



1. Check **Intermittent Sequences** (1), so that the Water Dispenser sequence (2) runs at its own interval during the imaging sequence (3).
2. Check that **Take Image** is disabled for the Water Dispenser sequence (2) and enabled for the imaging sequence (3).
3. Click **Start Acquisition**.

Inscoper I.S. merges the time points of both sequences into a single timeline, ordered by time. At each time point of the Water Dispenser sequence, a **Replenish** runs between two time points of the imaging sequence. The acquisition waits for the end of the refresh, including the return of the objective to its focus position (see Durations). An imaging time point that falls during a refresh is acquired once the refresh is complete.

All rights reserved.



With a 25-minute interval, each refresh restarts **Remaining Time** before it reaches `0`. With a 45-minute interval, **Remaining Time** stays at `0` for 15 minutes before each refresh (hatched areas), and no refresh starts during that time. The 5-minute imaging interval is an example.

Empty the circuit

Empty the circuit at the end of the session, before changing objective, and before storage:

1. Stop the acquisition and every movement of the microscope.
2. Switch **Discharge** (2) on.
3. Wait until **Status** shows WID not initialized and **Remaining Time** shows `0`. The **Init** and **Replenish** switches return to off, and a new **Init** is allowed.
4. Clean the objective according to the Nikon procedure approved by the laboratory.

If Inscoper I.S. is closed while the WID is still initialized, the WID is discharged automatically. Let this discharge complete before switching off the microscope.

Interaction rules and safety

- Only one WID operation may run at a time.
- **Init** is allowed only when the WID is not initialized.
- **Replenish** is allowed only after a successful **Init**, and may be repeated.
- **Discharge** is allowed only while the WID is initialized.
- Run a **Discharge** operation before changing objective.
- Do not change objective, move the focus or the XY stage, start an acquisition, or disconnect the USB cable during an operation.
- Do not move the objective by hand during a WID cycle.
- Keep the objective and the sample area clean and dry outside the intended water path.

If an operation fails

1. Wait for the final error status, and check that the pump has stopped. Once pumping has started, the controller stops the pump automatically on failure.
2. Check for leaks and for the safety of the objective.
3. If the WID remains initialized, run **Discharge** before changing objective.
4. If the state is uncertain, do not start a new **Init**. Inspect the diagnostic log or contact the instrument administrator.

All rights reserved.

> **Warning**
>
> These functions support the Nikon Ti2 water-immersion workflow. They do not replace the safety instructions of the microscope manufacturer or the fluid-handling procedure of the laboratory.

Reference

CONTROLS

Control	Meaning
Init	Starts a complete initialization and fills the circuit.
Replenish	Refreshes the water and removes air bubbles while the WID stays initialized.
Discharge	Removes the water and returns the WID to the not-initialized state.
Status	Current phase or state of the WID.
Remaining Time	Time, in milliseconds, until the next recommended refresh. `0` means that no countdown is running; it does not start a refresh.
UseWID	Virtual parameter saved in a channel. When switched on, acquiring the channel starts a refresh.

STATUS VALUES

Displayed status	Meaning
WID not initialized	The circuit has not been filled, or has been emptied.
WID Initialized	The circuit is filled and ready for imaging.
Pumping Water	The dispenser is filling the circuit.
Absorbing surplus water	Excess water is being removed.
Removing Air bubbles	The refresh cycle is removing air from the circuit.
Moving to replenish position	The objective and the focus mechanism are moving to a service position.
Replenishing Water	A short water refresh is in progress.
Discharging Water	Water is being removed from the tube.

DURATIONS

Operation	Approximate duration
Init	3 minutes of pumping, plus positioning and removal of the surplus water
Replenish	0.5 s of air removal, 10 s of water delivery, and 5.3 s of surplus removal, plus movements and confirmations
Discharge	90 seconds, plus positioning and confirmation
Countdown after **Init** or **Replenish**	1,800,000 ms (30 minutes)

All rights reserved.

The actual durations vary with the focus position, the mechanics of the stand, the USB communication, and the timing of the state changes reported by the microscope.

2026-09-28

All rights reserved.

1.8 Notes

1.8.1 Opening OME-TIFF Files

Inscoper I.S. writes each acquisition as an `.ome.tiff` image file accompanied by a `metadata.companion` file. This section explains how to open both in common image analysis software.

- The `.ome.tiff` file contains the image data only.
- The `metadata.companion` file opens the images together with their associated metadata.

ImageJ

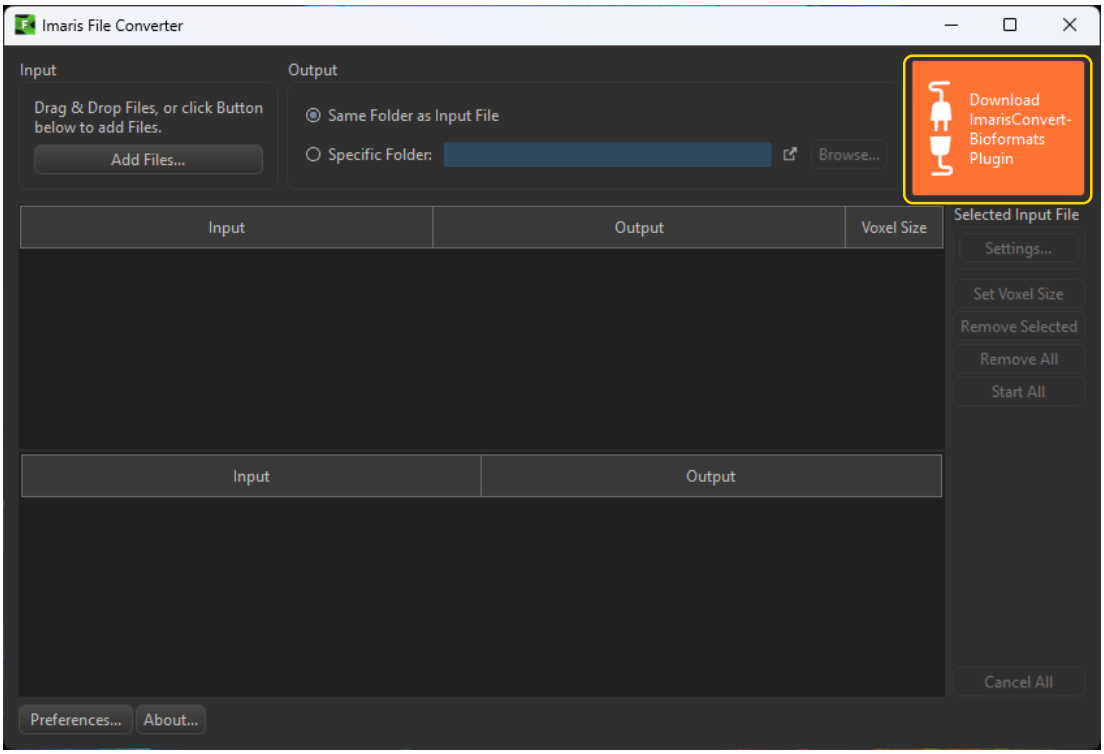
Download and install the Bio-Formats plugin in ImageJ before opening `.ome.tiff` and `metadata.companion` files. No further configuration is required.

Imaris

Imaris does not read `.ome.tiff` and `metadata.companion` files directly. Convert them first with the **Imaris File Converter**.

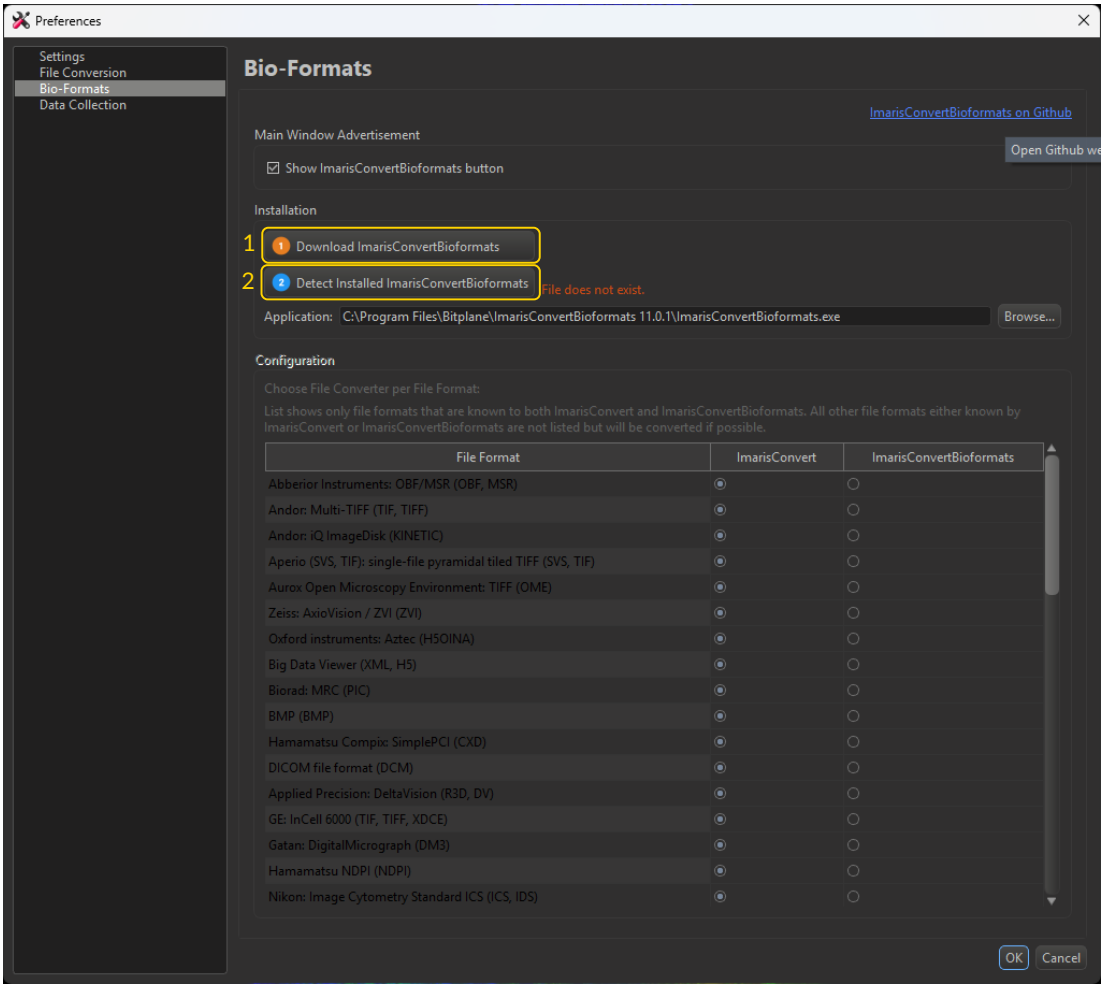
INSTALL THE BIO-FORMATS CONVERTER PLUGIN

1. Download and install the Imaris File Converter.
2. Open the Imaris File Converter and click **Download ImarisConvertBioformats Plugin**. The **Preferences... > Bio-Formats** panel opens.



3. In the **Installation** section, click **Download ImarisConvertBioformats** (1), then click **Detect Installed ImarisConvertBioformats** (2). The panel then reports the Bio-Formats JAR version and the plugin path.

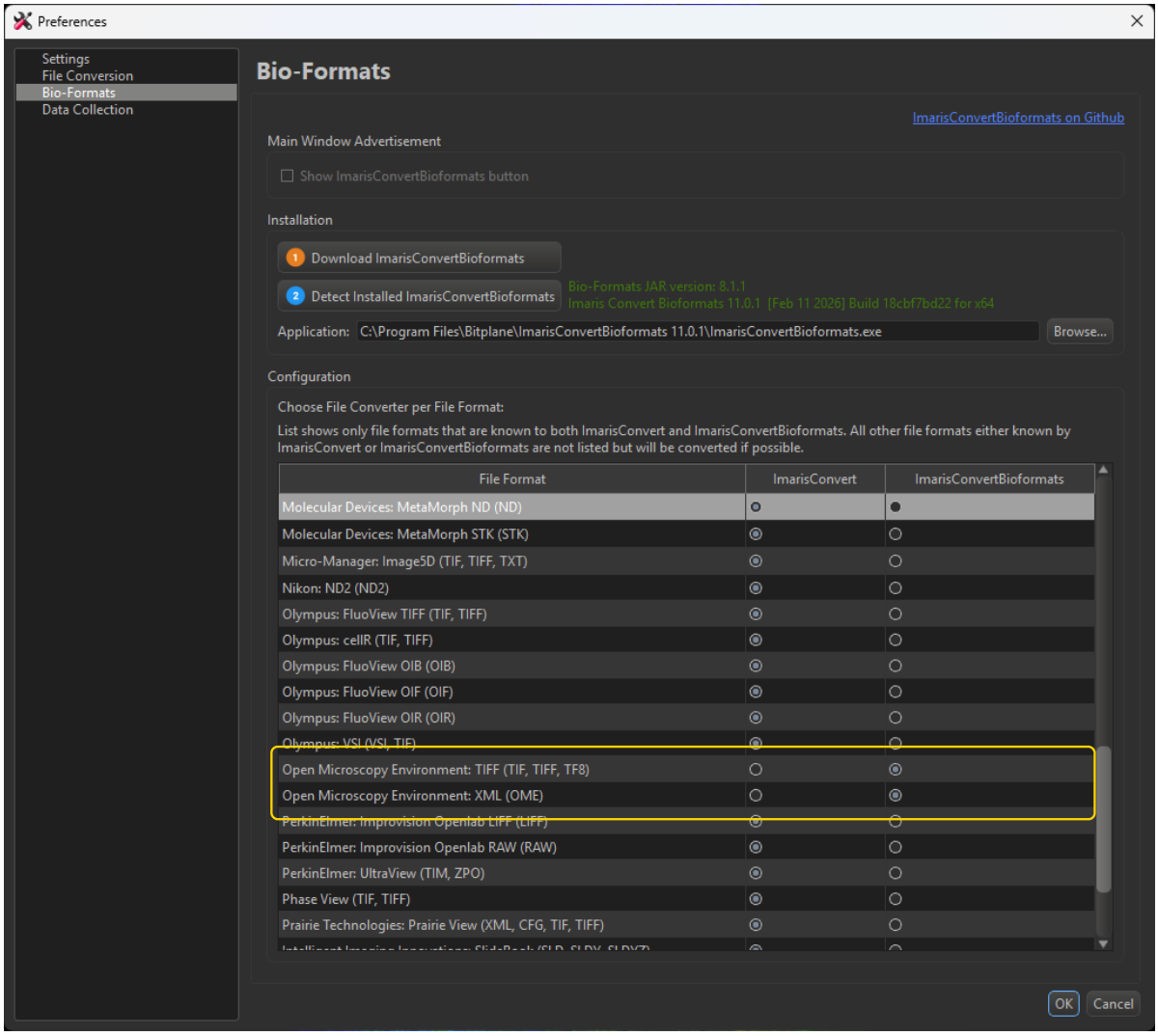
All rights reserved.



ASSIGN THE CONVERTER TO THE OME FILE FORMATS

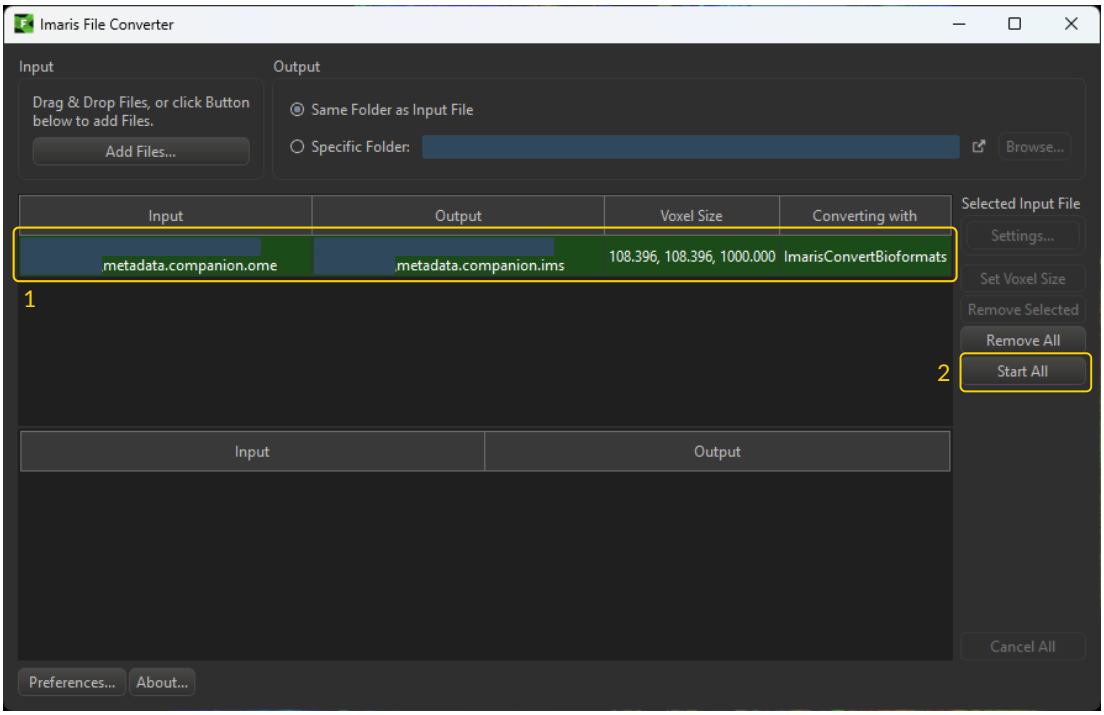
In the **Configuration** list of the same **Bio-Formats** panel, set both **Open Microscopy Environment: TIFF (TIF, TIFF, TF8)** and **Open Microscopy Environment: XML (OME)** to **ImarisConvertBioformats** instead of **ImarisConvert**, then click **OK**.

All rights reserved.



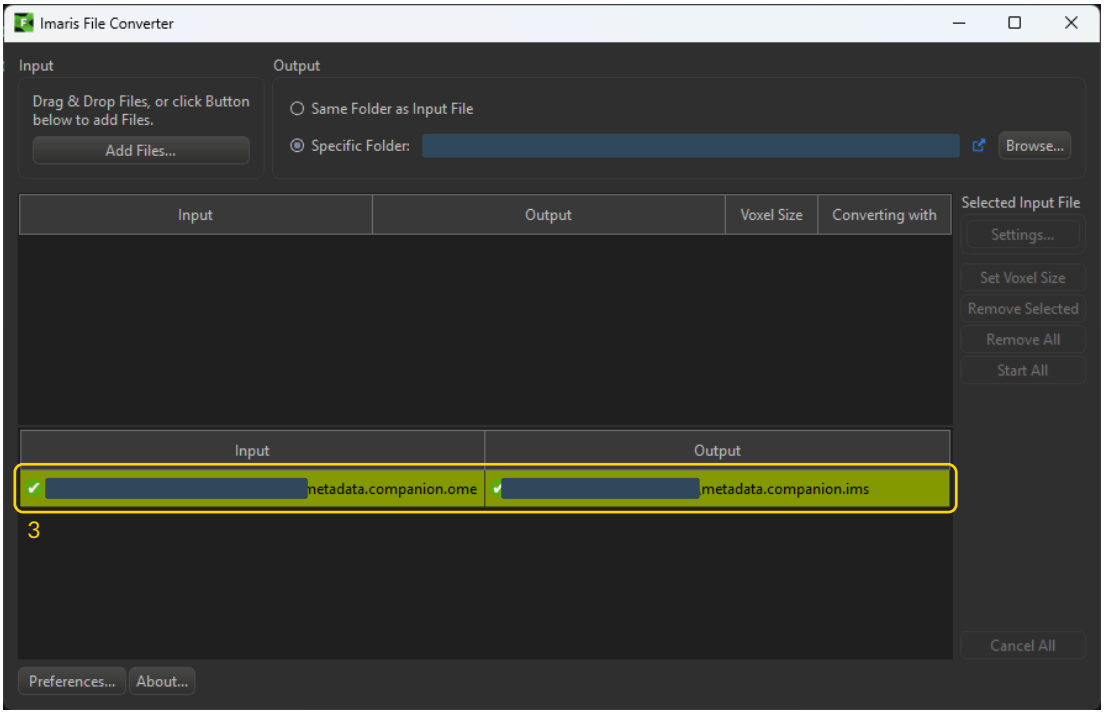
CONVERT THE FILE

1. Drag and drop the file into the input list, or click **Add Files...** (1).
2. Click **Start All** (2).



3. When the conversion completes, the file moves to the lower list and is ready to use (3).

All rights reserved.



Open the converted `.ims` file in **Imaris** or **Imaris Viewer**.

Other compatible software

These files are also compatible with other image analysis software, including Huygens, Aivia, and arivis Pro.

2026-09-10

All rights reserved.

1.8.2 Stitching Images in napari

A tiling acquisition writes one image file per stage position. Assembling those tiles into a single mosaic is called stitching. This page explains how to do it with napari and the napari-stitcher plugin, both of which are free and open source.

napari does not read Inscoper tiling metadata on its own. Dragging the acquisition folder into the viewer opens the tiles stacked on top of each other, with no positions. The loader script below reads the acquisition metadata and places every tile where the stage recorded it, so the plugin starts from an already correct mosaic and only refines it.

The script loads and places the tiles. Stitching and saving are done by you, in the napari window.

Download the loader script

File	Description
stitch_napari.py	Loads an Inscoper tiling acquisition into napari at its stage positions

Requirements

With the default saving settings, Inscoper I.S. writes a tiling acquisition in the form the script expects, and nothing has to be renamed, moved, or converted beforehand:

```
<project folder>/
├── <project name>.cbf              acquisition settings, not used here
├── <project name>.json             acquisition settings, not used here
└── images/
    └── RAW_DATA/                   this is the folder that matters
        ├── image_Pos0.ome.tif      one image file per stage position
        ├── image_Pos1.ome.tif
        ├── ...
        ├── image_Pos15.ome.tif
        ├── metadata.companion.ome  OME companion file, not used here
        └── metadata.json           required: tile positions and pixel size
```

Only two entries are required: one image file per stage position, and the `metadata.json` beside them. The rest is ignored. The number of image files is the number of tiles, and each file holds every plane recorded at that position: time, z, and channels all live inside the file.

SAVING SETTINGS THAT MATTER

Several saving settings are configurable in I.S., and some of them produce a layout the script cannot read. Check these before the acquisition, or re-save afterwards.

Setting	Where	Required value
Image saving format	Saving Images and Metadata	**All in one file**. **One TIFF per image** writes every plane as its own file, which the script refuses
Channel bundling	Display Settings, **Tiff Name** tab	Keep the default, which bundles all channels of a position into one file. The override that saves each channel separately is refused for the same reason
Metadata file name	Display Settings, **Metadata File Name** tab	Must remain `metadata.json`. The script looks for that name

Two settings are free:

- **TIFF file names.** The script takes the file names from `metadata.json` and never interprets them, so any naming convention built in the **Tiff Name** tab works. `image_Pos0.ome.tif` above is only the default.
- **.bigTiff.** Read like any other TIFF, and recommended for large mosaics as elsewhere.

All rights reserved.

If the acquisition was saved as one TIFF per image, or with one file per channel, the script stops with a message naming two of the files that share a stage position. Re-save the acquisition with **All in one file** rather than trying to load it.

Install napari

Install napari and the plugins in a dedicated environment, so that they do not interfere with other Python tools on the computer. The commands below use Miniconda, and are the same on Windows, macOS, and Linux.

```
conda create -n stitch python=3.11
conda activate stitch
pip install "napari[all]" napari-stitcher multiview-stitcher tifffile dask zarr
```

Check the installation by starting napari on its own:

```
napari
```

The viewer window opens. Close it before continuing.

Load the mosaic

Save `stitch_napari.py` anywhere you like, then run it on your acquisition:

```
conda activate stitch
python stitch_napari.py "<path to your acquisition folder>"
```

The path can be the `RAW_DATA` folder itself or any folder above it. The script searches downward for the acquisition metadata. Called without a path, it searches below the current folder.

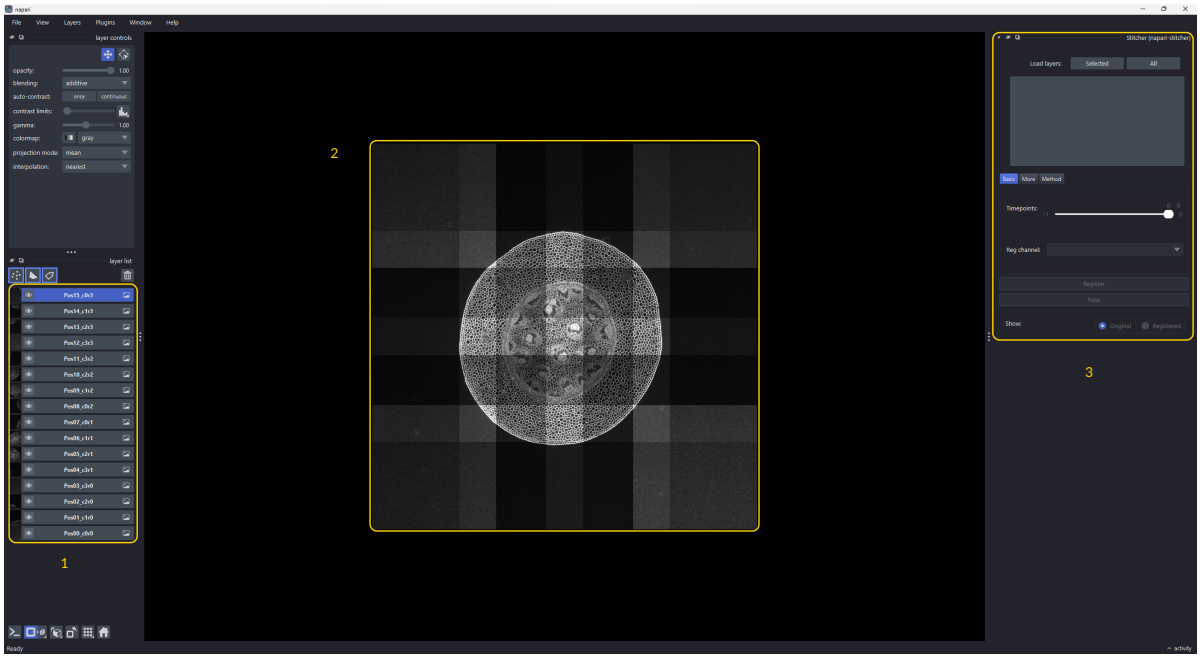
Before the viewer opens, the script prints a summary of what it found:

```
RAW_DATA: ...\images\RAW_DATA
  tiles       : 16  (SnakeMode)
  axes/shape  : YX (2304, 2304)
  pixel size  : 0.6492 um
  tile size   : 1495.8 x 1495.8 um
  overlap     : 30% x / 30% y
  stage axes  : NikonTi2-xAxis (x) / NikonTi2-yAxis (y)
  stage->image: [[-0.0007 -1. ] [-1. 0.0007]]
```

Read this summary before looking at the image. The number of tiles, the pixel size, and the overlap must match what you requested at acquisition. The last two lines report which stage axes were used and which orientation the system calibration produced, which is the information to check if the mosaic comes out wrong.

napari then opens on the loaded mosaic.

All rights reserved.



Layer list (1). One layer per tile, named `Pos<NN>_c<column>r<row>`: the acquisition order first, then the column and row in the mosaic. The two do not agree in snake mode, which is why `Pos04` sits at column 3, row 1 rather than at column 0, row 1.

Mosaic (2). The tiles, already placed from the acquisition metadata. The layers use additive blending, so the overlapping strips are drawn twice and appear brighter. That bright cross-hatch is the quickest way to confirm the placement: it must form a regular lattice. A staircase, a mirrored layout, or a diagonal smear means the placement is wrong, and registering it wastes time. See Troubleshooting.

Stitcher panel (3). Docked and empty. The script stops here.

LOADING PART OF A DATASET

For a large time-lapse or z-stack, load a single plane first to check the mosaic before committing to a full stitch.

Option	Effect
`--t N` or `--t A:B`	Load only these timepoints
`--z N` or `--z A:B`	Load only these z slices
`--c N`	Load only this channel. Required on multi-channel data, see below
`--source index`	Ignore the stage positions and rebuild an ideal grid from the row, column, and overlap

A single index removes the axis, and a range keeps it. `--t 0 --z 5` turns a time-lapse z-stack into plain two-dimensional tiles, while `--z 4:7` keeps a three-slice stack.

MULTI-CHANNEL ACQUISITIONS

napari-stitcher stitches one channel at a time, and it names the axes of an image from their count alone. A channel axis left in place is therefore mistaken for a z or a time axis, and the mosaic is registered and fused as if the channels were slices.

Select a single channel with `--c`, which removes the axis entirely:

```
python stitch_napari.py "<path to your acquisition folder>" --c 0
```

Then stitch the channels one after another, in separate runs. Channel numbering starts at 0, in the order the channels appear in the acquisition.

Tile placement itself does not depend on the channel, so every channel is fused on the same grid. To keep the results aligned exactly, note the registered positions from the first channel rather than re-registering each one independently.

All rights reserved.

If you omit `--c` on multi-channel data, the script loads it anyway but prints a warning:

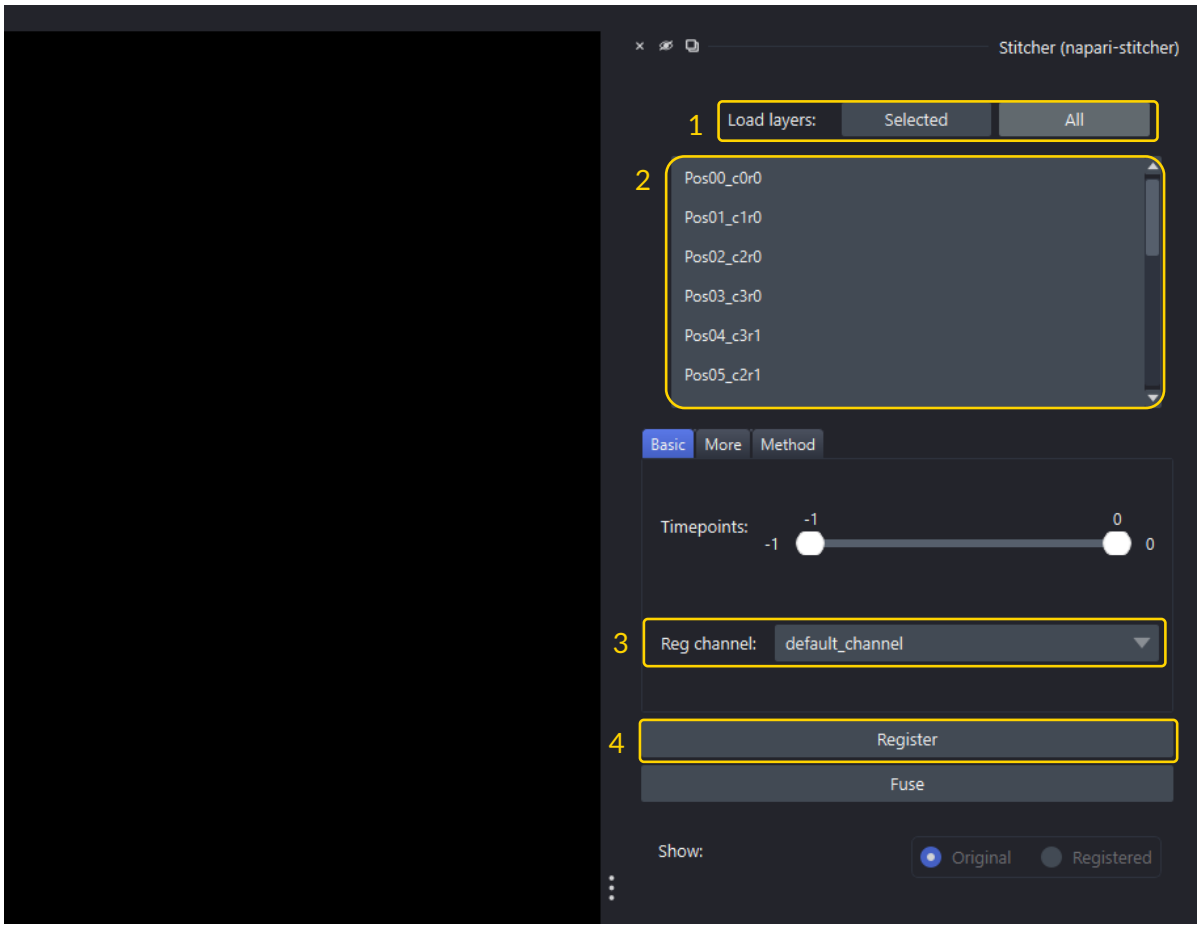
```
! the channel axis is still present (2 channels). napari-stitcher
  names axes from their count alone, so it will be mistaken for
  another axis. Pass a single index to --c to pick one channel.
```

`--source index` is a diagnostic option. If the stage-placed mosaic looks wrong, compare it against the ideal grid: if the ideal grid is correct and the stage-placed one is not, the problem lies in the system calibration rather than in the acquisition.

Stitch the mosaic

The script docks the Stitcher panel automatically. If it does not appear, open it from `Plugins > napari-stitcher > Stitcher`.

STEP 1: LOAD THE TILES INTO THE PLUGIN



1. Click **All (1)**, next to `Load layers`. Every image layer is loaded into the plugin with the positions the script set. To stitch only part of the mosaic, select those layers in the layer list first, then click **Selected**.
2. Check the **tile list (2)**. It must fill with the tile names. If it stays empty, nothing was loaded: the plugin refuses layers that still carry a channel axis. Right-click such a layer in the layer list and choose `Split Stack`, then load again.
3. Confirm **Reg channel (3)**. It shows `default_channel`, and there is nothing to choose: the script loads one channel at a time, so the channel was already selected on the command line with `--c`.
4. **Register (4)** is now enabled.

STEP 2: REGISTER

Registration measures the real shift between neighboring tiles from the image content, and refines the positions the script read from the metadata.

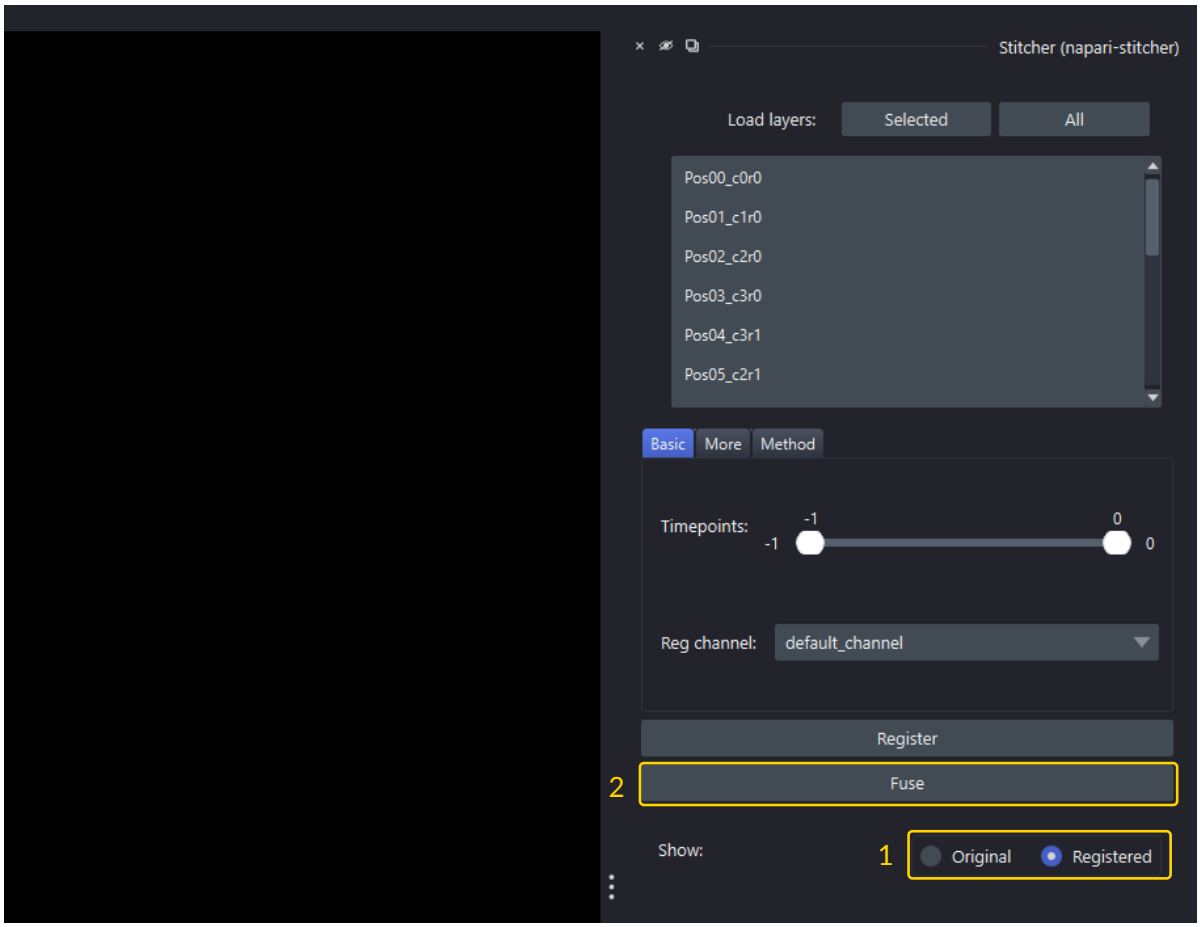
Two settings are worth reviewing before you click:

- On the `Method` tab, keep `Phase correlation`. It is the correct choice for a mosaic in which the tiles differ only by a translation, which is the normal case. `ITK Elastix` is needed only when the tiles also differ by a rotation.
- On the `More` tab, raising the registration binning to 2 or 4 shortens the computation considerably. The transform is still applied at full resolution, so the fused image loses no detail.

All rights reserved.

The `Timepoints` slider restricts what is registered and fused. On a time-lapse, start with a single timepoint.

Click **Register** and wait. A 16-tile single-plane mosaic takes roughly 30 seconds. The tiles then move to their refined positions.



1. Use **Show: Original / Registered (1)** to switch the viewer between the metadata placement and the registered one. Alternating between the two is the most direct way to judge what registration changed.
2. **Fuse (2)** is now enabled.

On a calibrated system, registration changes very little

The metadata placement is already accurate to about half a pixel. Registration is still worth running: it is inexpensive, it prevents small errors from accumulating over large mosaics, and it is the only step that detects a position the stage failed to reach.

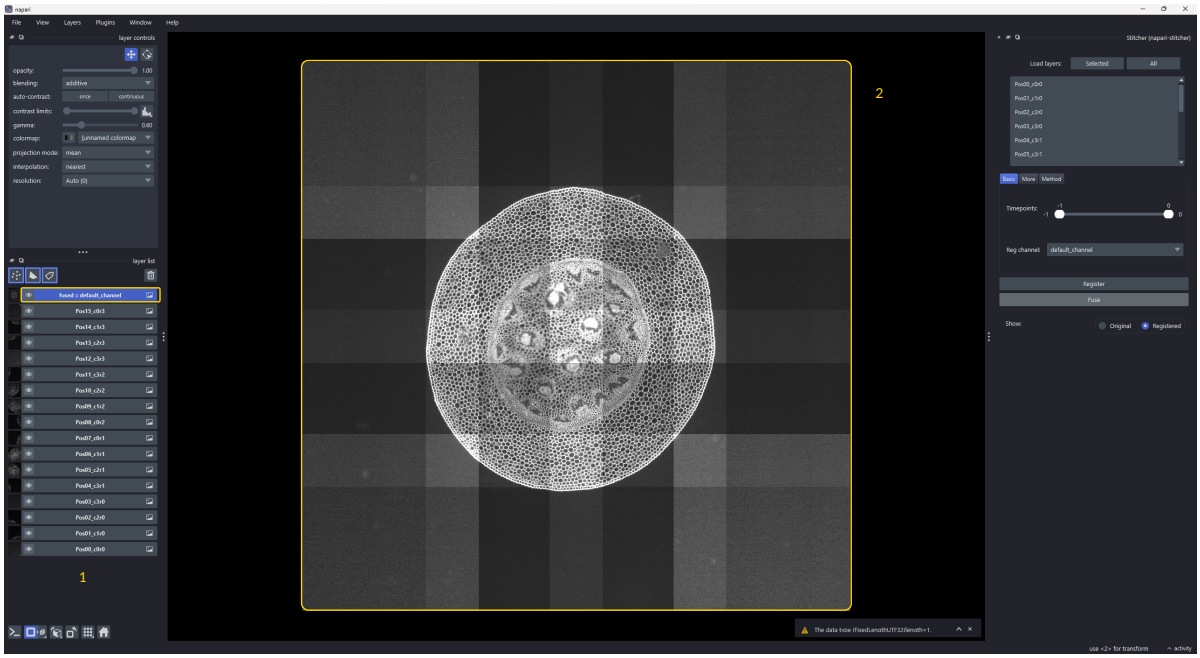
Mosaics acquired without overlap

Registration compares the overlapping strips between neighboring tiles. With no overlap it has nothing to compare and fails with `zero-size array to reduction operation minimum`. Skip this step and fuse directly on the metadata positions.

STEP 3: FUSE

Click **Fuse**.

All rights reserved.



1. A new layer, `fused :: default_channel` **(1)**, appears at the top of the layer list. It holds the assembled mosaic.
2. The **fused mosaic (2)** is displayed. It may still look tiled, as it does here, because the individual tile layers are still visible underneath it with additive blending. Alt-click the eye icon of the fused layer to hide every other layer and see the fused result on its own. Alt-click it a second time to restore the others.

Fusion covers every timepoint and every z slice allowed through by the `Timepoints` setting. On a four-dimensional dataset this is where the computation time goes, so restrict the selection before clicking **Fuse**.

STEP 4: SAVE

Select the fused layer, then choose `File > Save selected layer`.

Choose the format by size. A single-plane mosaic is written as TIFF without difficulty. For anything with a real time or z extent, save to `.zarr` instead: a five-timepoint, eleven-slice fusion of a 4 x 4 mosaic is about 5.6 GB, and a TIFF writer holds the whole array in memory before writing it.

Troubleshooting

The script stops with `N files share stage position ...`. The acquisition was saved as one file per plane, or as one file per channel, rather than one file per stage position. Re-save it with the **All in one file** image format and the default channel bundling. See Saving settings that matter.

The script stops with `No RAW_DATA/metadata.json found under ...`. Either the path holds no acquisition, or the metadata file was renamed in the **Metadata File Name** tab of the Display Settings. The script looks for `metadata.json`.

The script prints `no MosaicCalibration in metadata -- stage axes used as-is`. The acquisition carries no calibration data, so the script falls back to using the stage axes directly. This is rarely correct on a real stage. The mosaic may still be usable if the plugin can register it, but the system should be recalibrated.

The tile list stays empty after clicking `Load layers`. The plugin refuses layers that still carry a channel axis. Right-click the layer in the layer list, choose `Split Stack`, then load again.

The fused layer still looks tiled. The individual tile layers are still visible underneath it. Alt-click the eye icon of the fused layer to hide them.

2026-09-15

All rights reserved.

1.8.3 Software Autofocus

The software autofocus module operates in two phases: **Focus Evaluation** (calculating focus scores for each Z-plane) and **Focus Execution** (selecting the optimal plane and applying the mechanical offset).

Focus Evaluation Metrics

The software evaluates image focus by analyzing pixel intensity statistics using one of the selected algorithms below.

MAXIMUM INTENSITY METRIC

Conceptual Overview

The **Maximum** algorithm is designed for samples with sparse, high-contrast fluorescent signals on dark backgrounds. It identifies focus by finding the Z-plane that produces the single brightest pixel, assuming the peak fluorescence is reached at the exact focal plane.

Mathematical Formulation

The algorithm calculates the peak pixel intensity across the entire acquired sensor image:

$$F_{\text{Maximum}} = \max_{x \in [0,W), y \in [0,H)} I(x, y)$$

where:

- $I(x, y)$ represents the pixel intensity at coordinates (x, y).
- W and H represent the total width and height of the acquired sensor image in pixels, respectively.

SHARPNESS METRIC

Conceptual Overview

The **Sharpness** algorithm is designed for dense morphological structures (such as brightfield or phase contrast). It identifies focus by detecting high-frequency contrast gradients, which are strongest when the features are in sharp focus. To minimize background noise and ignore peripheral out-of-focus elements, the algorithm isolates a central region, filters noise, and computes local pixel gradients.

Mathematical Formulation

1. **ROI Definition:** To focus on the center of the field of view and ignore peripheral optical aberrations or sample tilt, the algorithm crops the center 20% of the sensor image. The cropped region width W_c and height H_c (in pixels) are defined as:

   $$W_c = 0.2 \cdot W, \quad H_c = 0.2 \cdot H$$

   The crop offsets O_w and O_h (in pixels) from the top-left corner are:

   $$O_w = 0.4 \cdot W, \quad O_h = 0.4 \cdot H$$

2. **Noise Reduction:** A spatial 3×3 median filter is applied to the cropped region to remove single-pixel sensor noise that might skew gradient calculations. The filtered intensity $M(i, j)$ for each pixel (i, j) within the cropped area is:

   $$M(i, j) = \text{median}\{I(O_w + i + \delta_x, O_h + j + \delta_y)\}$$

   where $\delta_x, \delta_y \in \{-1, 0, 1\}$.

   > **Median Filter Edge Behavior**
   >
   > When evaluating pixels located on the outer border of the 20% cropped ROI, the 3×3 median window reads adjacent pixels from the surrounding **original uncropped sensor image** I. Because the cropped ROI is centered, the required border pixels are always valid and accessible within the surrounding sensor space.

3. **Edge Detection:** The algorithm applies a diagonal gradient kernel K to detect high-frequency transitions:

   $$K = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

All rights reserved.

The local gradient response $S(k,l)$ is computed as the Frobenius inner product (element-wise product sum, denoted as :) of the kernel K and the local 3×3 neighborhood matrix $\mathbf{M}_{3\times3}(k,l)$ centered at (k,l):

$$S(k,l) = K : \mathbf{M}_{3\times3}(k,l) = \sum_{u=-1}^{1}\sum_{v=-1}^{1} K(u+2, v+2) \cdot M(k+u, l+v)$$

where the local neighborhood matrix is:

$$\mathbf{M}_{3\times3}(k,l) = \begin{bmatrix} M(k-1,l-1) & M(k,l-1) & M(k+1,l-1) \\ M(k-1,l) & M(k,l) & M(k+1,l) \\ M(k-1,l+1) & M(k,l+1) & M(k+1,l+1) \end{bmatrix}$$

Expanding this sum yields the exact pixel intensity equation:

$$S(k,l) = -2M(k-1,l-1) - M(k,l-1) - M(k-1,l) + M(k+1,l) + M(k,l+1) + 2M(k+1,l+1)$$

Gradient Kernel Edge Behavior

To prevent the 3×3 gradient kernel from reading outside the boundaries of the median-filtered crop matrix, the algorithm **excludes the outermost one-pixel border of the cropped ROI entirely** from the edge detection step. The loop limits are strictly bound to $k \in [1, W_c - 2]$ and $l \in [1, H_c - 2]$ during summation.

4. **Sharpness Score Computation:** The final sharpness score $F_{\text{Sharpness}}$ is the sum of the squared gradient responses over the cropped area:

$$F_{\text{Sharpness}} = \sum_{k=1}^{W_c-2}\sum_{l=1}^{H_c-2} [S(k,l)]^2$$

Autofocus Execution and Offset Correction

Once the software calculates the focus scores (F_i) for all N acquired Z-planes (where N is the value specified in the **Focus Number** field), it determines and applies the optimal focus position through the following steps:

BEST FOCAL PLANE SELECTION

The software identifies the index i_{best} of the Z-plane that maximizes the focus score:

$$i_{\text{best}} = \arg\max_{i \in [0,N)} F_i$$

Z-POSITION UPDATE

The system retrieves the physical Z-stage coordinate $Z(i_{\text{best}})$ associated with the best-scoring plane.

MECHANICAL OFFSET CORRECTION

In the subsequent image acquisition steps, the system drives the focus device to this optimal position:

$$Z_{\text{optimal}} = Z(i_{\text{best}})$$

If a destination device is selected under **Apply offset to**, the software calculates the relative physical offset:

$$\Delta Z = Z(i_{\text{best}}) - Z_{\text{nominal}}$$

The software then applies this offset correction to the selected destination device, ensuring the sample remains in focus throughout the multi-position or time-lapse sequence.

2026-09-08

All rights reserved.

1.8.4 Data Processor Algorithms

The Data Processor module executes various algorithms to transform raw images during or after acquisition. This document provides the conceptual overview and the exact mathematical formulation for each available processing operation.

No Processor

OVERVIEW

The `NO_PROCESSOR` operation acts as a pass-through node in the processing pipeline. It forwards the input image stream to the next node or to storage without performing any modifications.

MATHEMATICAL FORMULATION

For an input image $I_{\text{in}}(x, y)$ at pixel coordinate (x, y), the output image $I_{\text{out}}(x, y)$ is identical to the input:

$$I_{\text{out}}(x, y) = I_{\text{in}}(x, y)$$

Simple Tiling

OVERVIEW

The `SIMPLE_TILING` processor combines a sequence of individual tile images acquired during a grid-based acquisition into a single large mosaic image. It maps each tile to its coordinates on a global grid using nominal offsets without performing any alignment optimization or edge blending.

MATHEMATICAL FORMULATION

Let the input tile image at grid column index $i \in [0, N_x)$ and grid row index $j \in [0, N_y)$ be denoted as $I_{\text{in},i,j}$. Let W_{tile} and H_{tile} represent the width and height of a standard tile (in pixels), respectively, and let O_x and O_y represent the fractional grid overlaps along the X and Y dimensions (in the range $[0, 1)$).

1. **Mosaic Dimensions:** The total width W_{out} and height H_{out} of the output mosaic image (in pixels) are defined as:

$$W_{\text{out}} = (N_x - 1) \cdot W_{\text{tile}} \cdot (1 - O_x) + W_{\text{last}}$$
$$H_{\text{out}} = (N_y - 1) \cdot H_{\text{tile}} \cdot (1 - O_y) + H_{\text{last}}$$

where W_{last} and H_{last} represent the actual width and height of the boundary tiles.

1. **Tile Offset Calculation:** The top-left offset coordinate $(x_{\text{offset}}(i), y_{\text{offset}}(j))$ for the tile at column index i and row index j is calculated as:

$$x_{\text{offset}}(i) = i \cdot W_{\text{tile}} \cdot (1 - O_x)$$
$$y_{\text{offset}}(j) = j \cdot H_{\text{tile}} \cdot (1 - O_y)$$

1. **Mosaic Synthesis:** Each pixel (x', y') in the output mosaic I_{out} corresponding to tile $I_{\text{in},i,j}$ at local pixel coordinates (x, y) is copied directly:

$$I_{\text{out}}(x_{\text{offset}}(i) + x, y_{\text{offset}}(j) + y) = I_{\text{in},i,j}(x, y)$$

where $x \in [0, W_{\text{tile}})$ and $y \in [0, H_{\text{tile}})$. If tiles overlap, pixels are mapped directly as they are written to the mosaic canvas, with no blending applied.

All rights reserved.

Stitch Tiling

OVERVIEW

The `STITCH_TILING` processor aligns and blends adjacent tiles in a mosaic to create a seamless composite image. It uses correlation-based image registration to compute optimal relative offsets between adjacent tiles, followed by a distance-based blending algorithm inside the overlapping regions to smooth intensity transitions.

MATHEMATICAL FORMULATION

The processor performs its operations in two sequential phases: registration (offset optimization) and blending (image merging).

Phase 1: Registration and Offset Optimization

Relative displacement vectors (offsets) between adjacent tiles are calculated by maximizing the normalized cross-correlation across their expected overlap regions. If a boundary transition lacks sufficient texture or contains high alignment errors, the registration algorithm handles the transition using one of three alternative modes:

1. **Displacement Interpolation**: The processor calculates the mean relative East-West displacement (WE_x, WE_y) and North-South displacement (NS_x, NS_y) using only tile boundaries whose correlation score exceeds a correlation threshold T_{corr}. For any boundary transition where the correlation score falls below T_{corr}, or where the calculated relative offset deviates from the average displacement by more than 10 pixels, the processor overrides the offset using the respective mean displacement:

$$\text{Offset}_{E-W} = (WE_x, WE_y)$$

$$\text{Offset}_{N-S} = (NS_x, NS_y)$$

1. **Tile Exclusion**: Tiles lacking any neighbor boundary correlation above the threshold T_{corr} are excluded from the global offset calculation and from the final stitched mosaic.
2. **Stage Coordinate Fallback**: The processor retains the default stage coordinates provided by the microscope stage controller.

Once the relative offsets between neighbors are established, a global coordinate system is constructed by setting the top-left tile as the global reference origin $(0, 0)$ and recursively adding the relative offset vectors to propagate global coordinates to all remaining tiles.

Phase 2: Blending and Merging

Let S represent the set of input tiles that overlap at a given global mosaic coordinate (x, y). The blended output intensity $I_{\text{out}}(x, y)$ is computed as follows:

- **Single-Tile Regions ($|S| = 1$):** If only one input tile $k \in S$ covers the coordinate, the pixel value is written directly:

$$I_{\text{out}}(x, y) = I_{\text{in},k}(x - x_{\text{offset},k}, y - y_{\text{offset},k})$$

- **Double-Tile Overlap Regions ($|S| = 2$ with Horizontal or Vertical Alignment):** When exactly two tiles overlap and their offsets align strictly along a single axis, the pixel value is linearly blended using a normalized distance ratio $r \in [0, 1]$:

$$I_{\text{out}}(x, y) = (1 - r) \cdot I_{\text{in},1}(x - x_{\text{offset},1}, y - y_{\text{offset},1}) + r \cdot I_{\text{in},2}(x - x_{\text{offset},2}, y - y_{\text{offset},2})$$

where r represents the relative distance of the pixel coordinate (x, y) from the boundary of the overlap region.

- **Multi-Tile Overlap Regions ($|S| > 1$):** For corner overlaps or complex overlapping configurations, the processor calculates a weighted average based on the shortest distance from the pixel to the boundary of each overlapping tile. The weight $w_k(x, y)$ for input tile $k \in S$ is defined as:

$$w_k(x, y) = \max\left(\min\left(x - x_{\text{offset},k}, x_{\text{offset},k} + W_{\text{tile}} - x, y - y_{\text{offset},k}, y_{\text{offset},k} + H_{\text{tile}} - y\right), 1\right)$$

The blended output pixel intensity is:

$$I_{\text{out}}(x, y) = \frac{\sum_{k \in S} w_k(x, y) \cdot I_{\text{in},k}(x - x_{\text{offset},k}, y - y_{\text{offset},k})}{\sum_{k \in S} w_k(x, y)}$$

All rights reserved.

Standard Deviation on Fly

OVERVIEW

The `STANDARD_DEVIATION_ON_FLY` processor computes the pixel-wise standard deviation across a chosen acquisition dimension (e.g., over time or Z-planes) as new images are acquired. To minimize memory usage and processing overhead, it implements Welford's online variance calculation algorithm, updating the variance incrementally for each pixel without storing the full stack in memory.

MATHEMATICAL FORMULATION

Let $I_{\text{in},k}(x, y)$ represent the intensity of the pixel at coordinate (x, y) for the k-th input image added to the stack. At step k (for $k \geq 1$):

1. **Mean Update:** The running mean $\mu_k(x, y)$ is updated as:

$$\mu_k(x, y) = \mu_{k-1}(x, y) + \frac{I_{\text{in},k}(x, y) - \mu_{k-1}(x, y)}{k}$$

with initial condition $\mu_0(x, y) = 0$.

1. **Squared Difference Sum Update:** The running sum of squared differences $Q_k(x, y)$ is updated as:

$$Q_k(x, y) = Q_{k-1}(x, y) + (I_{\text{in},k}(x, y) - \mu_{k-1}(x, y)) \cdot (I_{\text{in},k}(x, y) - \mu_k(x, y))$$

with initial condition $Q_0(x, y) = 0$.

1. **Standard Deviation Computation:** The output standard deviation image $I_{\text{out}}(x, y)$ at step k is defined as:

$$I_{\text{out}}(x, y) = \begin{cases} 0 & \text{if } k = 1 \\ \sqrt{\frac{Q_k(x,y)}{k-1}} & \text{if } k > 1 \end{cases}$$

Shading Correction

OVERVIEW

The `SHADING_CORRECTION` processor corrects for spatial variations in illumination intensity, vignetting, and optical dust artifacts. It normalizes each pixel in the input image using a pre-acquired background (or flatfield) calibration image.

MATHEMATICAL FORMULATION

Let $I_{\text{in}}(x, y)$ represent the input image pixel intensity, and let $B(x, y)$ represent the corresponding pixel intensity in the calibration background image.

Let μ_B represent the average pixel intensity computed across the entire background calibration image:

$$\mu_B = \frac{1}{W \cdot H} \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} B(x, y)$$

The output corrected pixel intensity $I_{\text{out}}(x, y)$ is computed as:

$$I_{\text{out}}(x, y) = \frac{I_{\text{in}}(x, y) \cdot \mu_B}{B(x, y)}$$

where:

- If the background image dimensions do not match the input image, the background image is automatically rescaled to match the input image dimensions using nearest-neighbor scaling (specifically, midpoint pixel subsampling) before correction.
- The background intensity $B(x, y)$ is assumed to be non-zero to avoid division by zero.

All rights reserved.

Max Intensity Projection

OVERVIEW

The processors `TIME_MAX` and `FOCUS_MAX` project an image series into a 2D image by selecting the maximum intensity value for each pixel location along the specified dimension.

- `TIME_MAX` calculates a maximum intensity projection along the time dimension.
- `FOCUS_MAX` calculates a maximum intensity projection along the Z-stack focus dimension.

MATHEMATICAL FORMULATION

Let $I_{\text{in}}(c, t, z, x, y)$ represent the input pixel intensity at coordinate (x, y) for channel c at time index t and Z-stack index z.

- **For `TIME_MAX`:** The output maximum intensity projection $I_{\text{out}}(c, z, x, y)$ across all T time points is defined as:

$$I_{\text{out}}(c, z, x, y) = \max_{t \in [0, T)} I_{\text{in}}(c, t, z, x, y)$$

- **For `FOCUS_MAX`:** The output maximum intensity projection $I_{\text{out}}(c, t, x, y)$ across all Z focus positions is defined as:

$$I_{\text{out}}(c, t, x, y) = \max_{z \in [0, Z)} I_{\text{in}}(c, t, z, x, y)$$

Average Intensity Projection

OVERVIEW

The processors `TIME_AVERAGE` and `FOCUS_AVERAGE` project an image series into a single 2D projection by calculating the arithmetic mean intensity value for each pixel location along the specified dimension.

- `TIME_AVERAGE` calculates the mean projection along the time dimension.
- `FOCUS_AVERAGE` calculates the mean projection along the Z-stack focus dimension.

MATHEMATICAL FORMULATION

Let $I_{\text{in}}(c, t, z, x, y)$ represent the input pixel intensity at coordinate (x, y) for channel c at time index t and Z-stack index z.

- **For `TIME_AVERAGE`:** The output average intensity projection $I_{\text{out}}(c, z, x, y)$ across all T time points is defined as:

$$I_{\text{out}}(c, z, x, y) = \frac{1}{T} \sum_{t=0}^{T-1} I_{\text{in}}(c, t, z, x, y)$$

- **For `FOCUS_AVERAGE`:** The output average intensity projection $I_{\text{out}}(c, t, x, y)$ across all Z focus positions is defined as:

$$I_{\text{out}}(c, t, x, y) = \frac{1}{Z} \sum_{z=0}^{Z-1} I_{\text{in}}(c, t, z, x, y)$$

Subtract Background

OVERVIEW

The `SUBTRACT_BACKGROUND` processor reduces background fluorescence or ambient light noise. It calculates the average pixel intensity within a user-defined Region of Interest (ROI) representing background, and subtracts this constant value from every pixel in the image.

MATHEMATICAL FORMULATION

Let $I_{\text{in}}(x, y)$ represent the input pixel intensity. Let $\mathcal{A}$ be the set of coordinates defining the background region.

1. **Background Mean Calculation:** The average background intensity value μ_{bg} is calculated as:

All rights reserved.

$$\mu_{\text{bg}} = \frac{1}{|\mathcal{A}|} \sum_{(x_r, y_r) \in \mathcal{A}} I_{\text{in}}(x_r, y_r)$$

1. **Subtraction and Clamping:** The output intensity $I_{\text{out}}(x, y)$ is computed as:

$$I_{\text{out}}(x, y) = \max\left(0, I_{\text{in}}(x, y) - \mu_{\text{bg}}\right)$$

If no ROI is defined, the processor acts as a pass-through and performs no modification:

$$I_{\text{out}}(x, y) = I_{\text{in}}(x, y)$$

2026-09-08

All rights reserved.

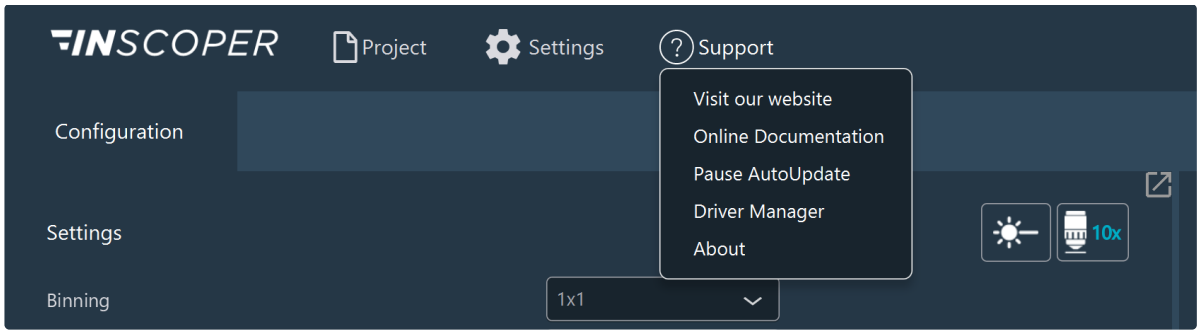
1.9 Support

If you have any questions or encounter any issues with your software, please contact us through our ticketing tool, Mainti4 (your login credentials will be provided after installation). To help us diagnose the issue quickly, we will need your software version and logs folder.

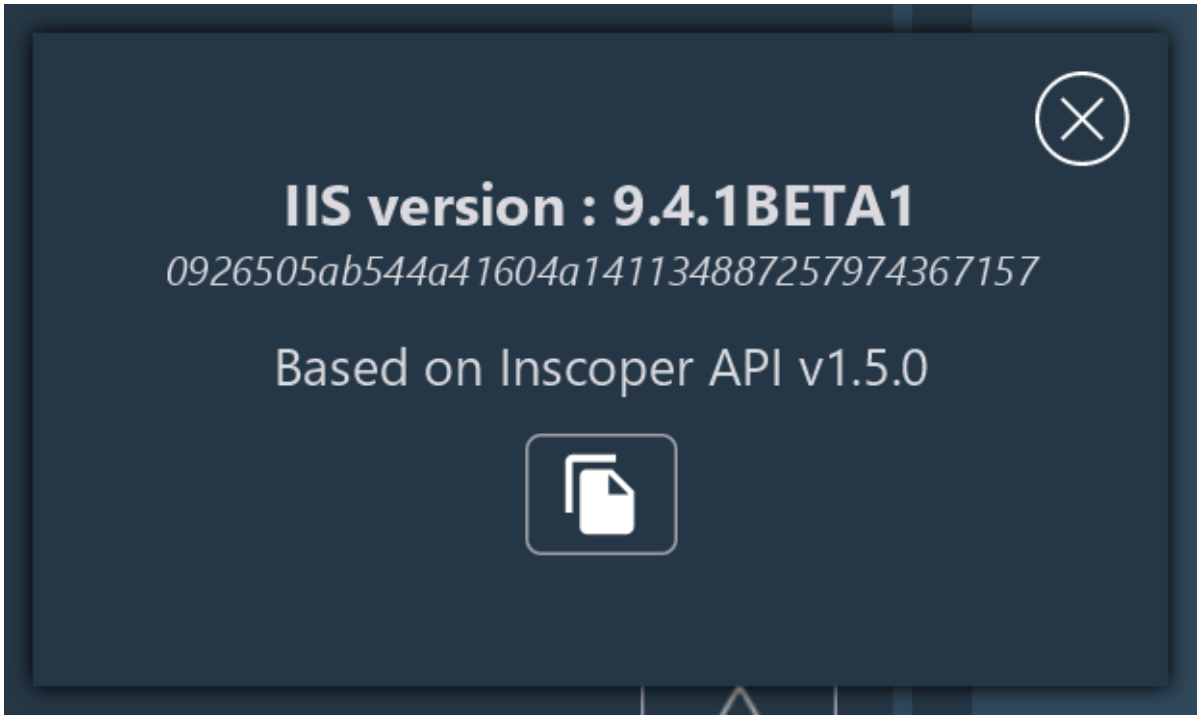
This page explains exactly where to find this information.

1.9.1 Inscoper Software Version

To find your Inscoper version, open the **Support** drop-down menu and click **About**.



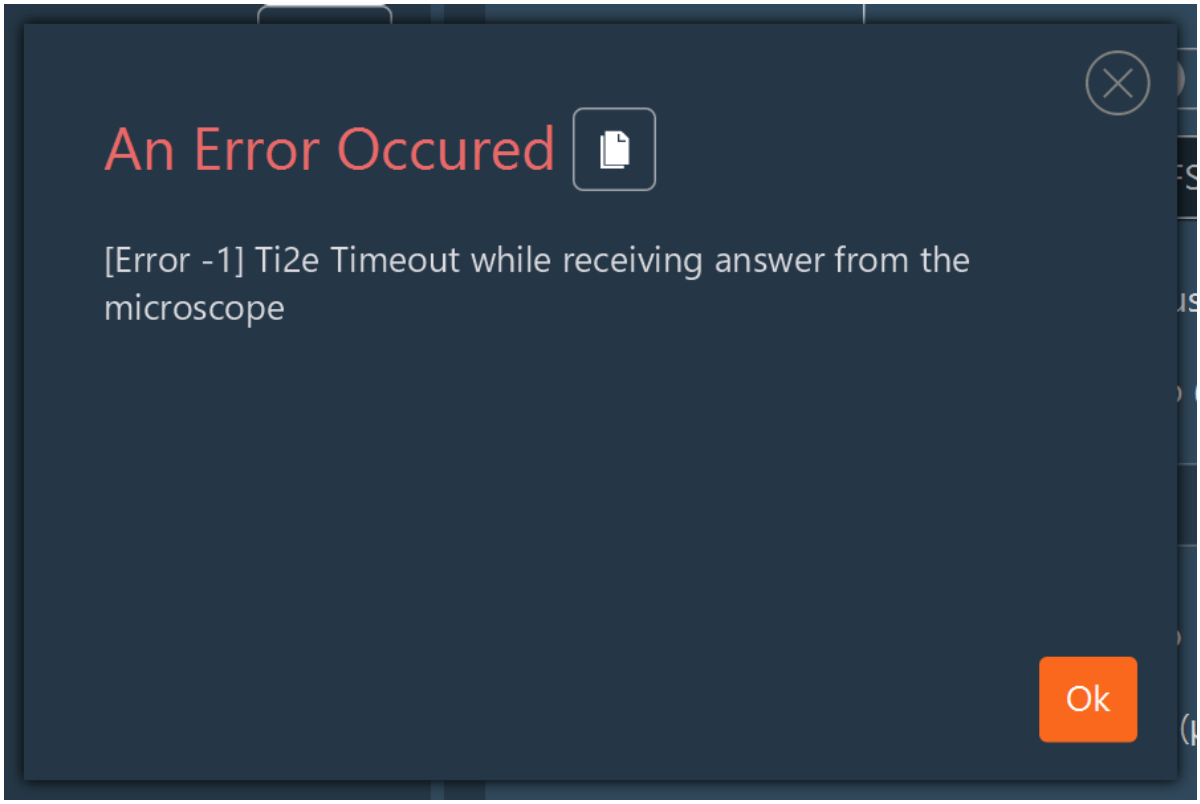
A pop-up will appear displaying all technical details, including the version and API numbers. If you encounter any issues with your software, you can copy and paste this information directly into your support ticket.



1.9.2 Error Windows

If you encounter an error while using the software, please copy and paste the error message directly into your support ticket.

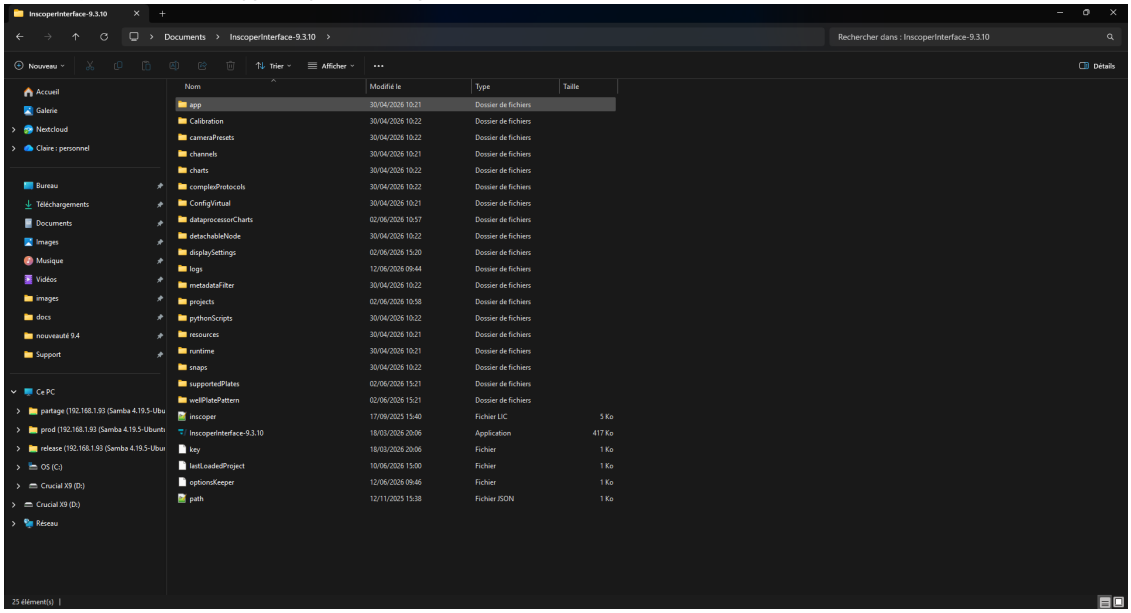
All rights reserved.



1.9.3 Find log folder

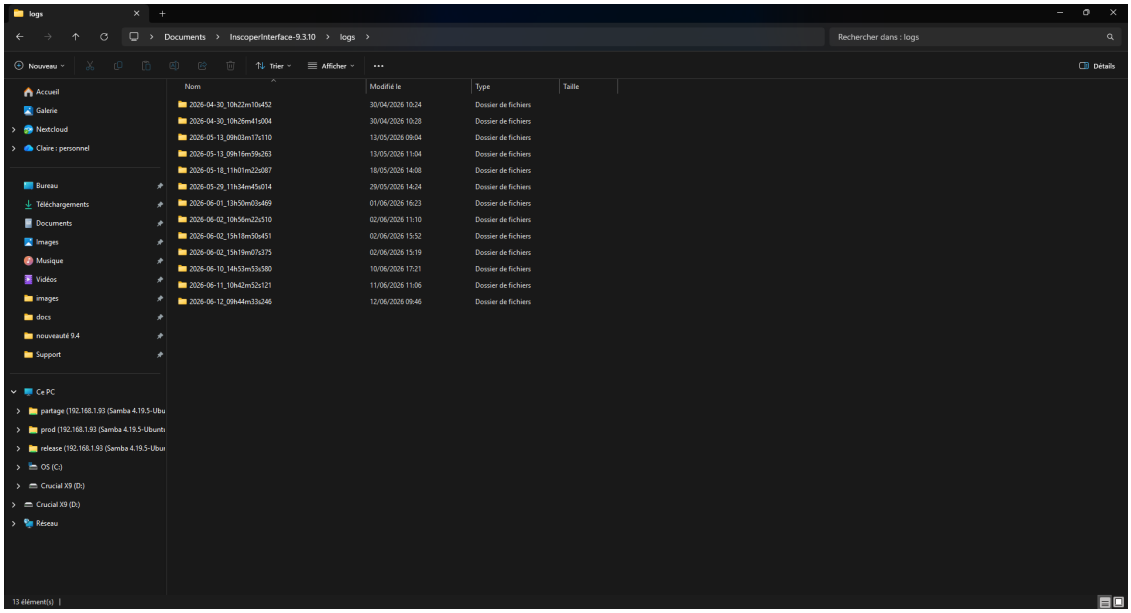
Follow these steps to locate your latest log folder:

- Open your **Inscoper directory** (typically: C:\Inscoper).



- Open the **executable folder**, then locate and open the **logs folder**.

All rights reserved.



- Select the most **recently created** folder inside.
- Attach this **folder** to your support ticket.

> **Note**
>
> A new log folder is generated every time the software interface is opened, so the newest folder will contain the details of your current session.

2026-09-08

All rights reserved.

1.10 Legal

1.10.1 Legal Information

Welcome to the INSCOPER Documentation Portal.

This section provides the legal information related to INSCOPER products, including copyright notices, trademarks, certifications, patents, disclaimers and contact information.

We strive to keep this documentation accurate and up to date. If you identify information that is incomplete, unclear or inaccurate, please contact the INSCOPER team.

Disclaimer

The information contained in this documentation is provided "as is", without warranties, conditions or representations of any kind, whether express, implied, statutory or otherwise, including, but not limited to, warranties of merchantability, non-infringement or fitness for a particular purpose.

INSCOPER shall not be liable for any direct, indirect, incidental, consequential or other damages arising from the use of this documentation or the information it contains.

Copyright

© INSCOPER SAS. All rights reserved.
INSCOPER and the INSCOPER logo are trademarks of INSCOPER SAS.

This documentation and the associated illustrations remain the property of INSCOPER SAS. They may not be copied, reproduced, distributed or disclosed, in whole or in part, without prior written permission, except as required for the normal use of the product.

The information contained in this documentation does not grant any license or right to use patents, trademarks, copyrights, or other intellectual property owned by INSCOPER or third parties.

Patents

INSCOPER products include technologies protected by the following patents:

Optimized Microscope Control:

- EP3123149
- US10330911
- FR3019324

Smart Microscope Automation:

- EP4111249
- US12474564
- FR3107604

Product updates

INSCOPER continuously improves its products and documentation. Product specifications and documentation may change without notice and are incorporated into future software releases and documentation updates.

Contact

INSCOPER SAS
3771 boulevard des Alliés
35510 Cesson-Sévigné
France

Email: contact@inscoper.com

All rights reserved.

2026-10-21

All rights reserved.