User Manual - Silver Lab Microscopy Software

Table of Contents

[Deploying the source code 2](#_Toc461451924)

[Setting up the project directory 2](#_Toc461451925)

[Installing the common Vis directory in LabVIEW’s User.llb 3](#_Toc461451926)

[Installing required VI packages 3](#_Toc461451927)

[LabVIEW Version 4](#_Toc461451928)

[Minimum requirements for full software functionality 4](#_Toc461451929)

[Minimum requirements for demo mode only 4](#_Toc461451930)

[Required Modules, Toolkits, Drivers, & Dependencies 5](#_Toc461451931)

[User Login & Profiles 7](#_Toc461451932)

[Output Directory Format 12](#_Toc461451933)

[Calibrated Power Display 15](#_Toc461451934)

[Pockels Calibration 15](#_Toc461451935)

[Calibration Checks during Software Initialisation 18](#_Toc461451936)

[Managing Objectives Setup 19](#_Toc461451937)

[Stimulus Trigger 20](#_Toc461451938)

[Functional Imaging Viewer 22](#_Toc461451939)

[Exporting Data from the Functional Imaging Viewer 24](#_Toc461451940)

[ROI Grouping 27](#_Toc461451941)

[Time-series Viewer & Editor Window 30](#_Toc461451942)

[Exporting Time Series Videos for 3rd Party Players 33](#_Toc461451943)

[Setting up the Source Code’s Configuration Files Folder 36](#_Toc461451944)

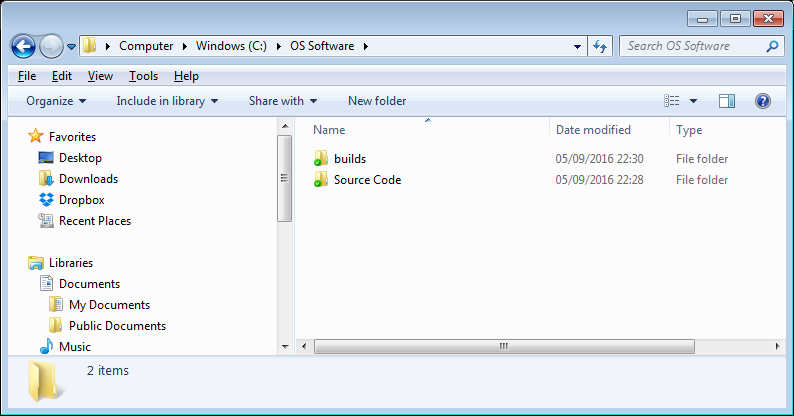
[Field Of View Setup 38](#_Toc461451945)

## Deploying the source code

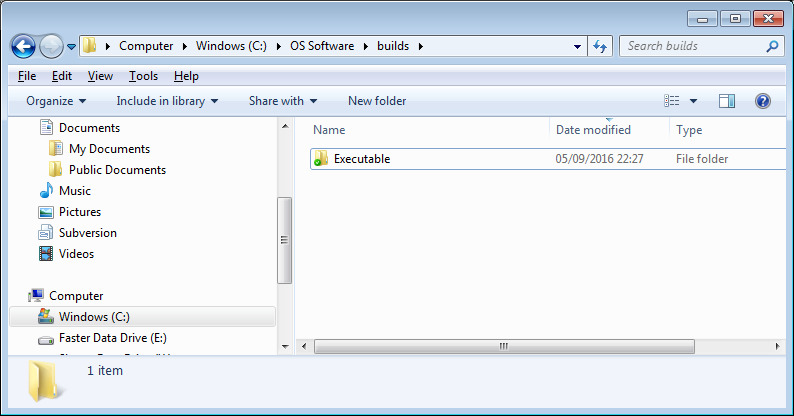
### Setting up the project directory

On downloading or cloning the code from github, move the folder ‘src’ to the following location -

C:\OS Software\Source code. Although the executable has not been provided with the release, the build specification is present in the project file ‘Microscope software - commonised.lvproj’. Interested parties can generate the executable themselves. Once generated, place the Executable folder in ‘C:\OS Software\builds\Executable’.



Ensure the directory arrangement is similar to the one shown, with the ‘builds’ folder at the same directory level as the ‘Source Code’ folder and the ‘Executable’ folder inside the ‘builds’ folder. If you do not set up the directory structure as described, you will need to update the executable’s build spec in the LabVIEW project to work with the modified directory structure.



By default the source code and the executable are set up to share the same configuration files. This means that if a setting is changed through the source code, it is seen by the executable and vice-versa. The executable and the source code’s settings are effectively always in sync.

By default to source code uses the executable’s configuration files at ‘C\:OS Software\builds\Executable\RIG-SPECIFIC FILES\Configuration Files’. This expected path for the configuration files is stored in the ‘Config pointer.ini’ file. The ‘Config pointer.ini’ file can be found in the ‘…\Source Code\RIG-SPECIFIC FILES’ folder.

You can delete the ‘Config pointer.ini’ file before running the source code if you want to cause an dialogue window to open during software initialisation allowing to tell the source code which configuration files folder to use. The ‘Config pointer.ini’ file will be regenerated automatically to hold the new path. You can use this approach if you want to set up the source code to use its own configuration files (these can be found in ‘C:\OS Software\Source Code\RIG-SPECIFIC FILES\Configuration Files’). You can also use this approach if you have placed the executable at a location other than the default expected location i.e. ‘C:\OS Software\builds\Executable’.

If you do not delete the ‘Config pointer.ini’ file before you run the source code, the default expected location for the configuration files is ‘C\:OS Software\builds\Executable\RIG-SPECIFIC FILES\Configuration Files’. Therefore you should ensure this path exists.

The Configuration Files folder holds the main setup file (i.e. ‘Setup.ini’) which applies globally for all users. The subfolder named ‘User Profiles’ stores all other user-specific settings.

### Installing the common Vis directory in LabVIEW’s User.llb

Before you attempt to open the top level VI first time, copy the ‘My Palettes.lib’ folder (found in ‘…\Add on\Common VIs\My Palettes’) to C:\Program Files (x86)\National Instruments\LabVIEW 2017\user.lib\My Palettes.lib (or similar folder depending on LabVIEW version). The ‘My Palettes.lib’ is one of the folders you downloaded.

### Installing required VI packages

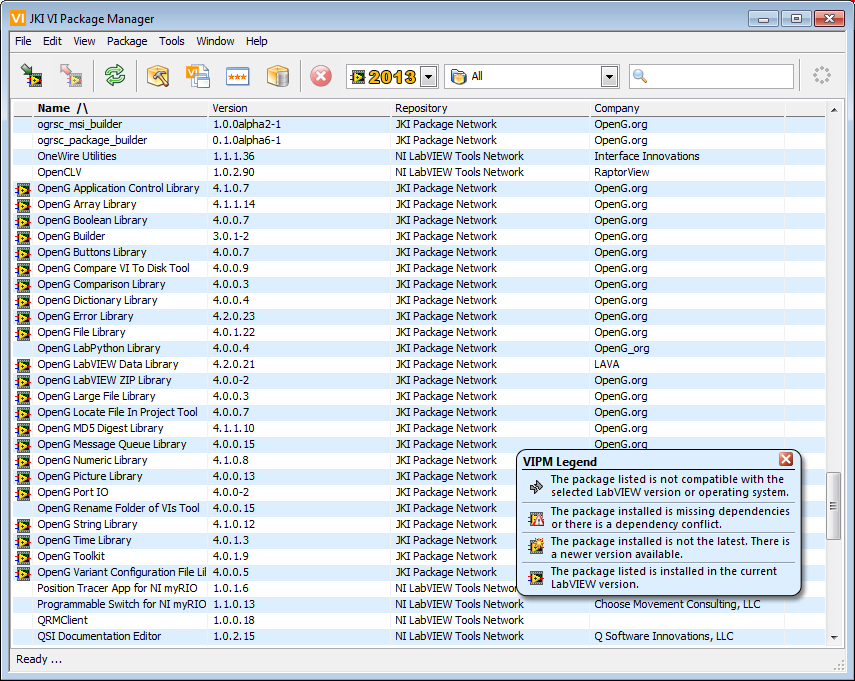
Install the OpenG and MGI packages using VI Package Manager (VIPM).

VIPM installs by default with the most recent LabVIEW versions.

If unavailable, download from <http://vipm.jki.net/>

Zipped backups have been included but do not use those backups. You must install the above packages properly via VIPM.

If when you try to open the top level VI, LabVIEW complains that certain Vis are missing this is because one or more required VI packages are missing. You can locate the missing packages and install them easily through VIPM. (Hint: The name prefix of missing VIs will typically match the name of the package the missing VIs belong to). Below you can see a screenshot of the VIPM window listing all available packages. The LabVIEW icon appears next to those packages that have been installed.



VI Package Manager Window

### LabVIEW Version

The initial release of the source code was in LabVIEW 2013 and LabVIEW 2017 and has been recently upgraded to LabVIEW 2020. If you use a higher version of LabVIEW and save the code, it will no longer be possible to open that source code in a previous version of LabVIEW. To enable collaboration, ensure that you are using the same LabVIEW version as that used by the Silver Lab.

### Minimum requirements for full software functionality

* OS: Windows 10
* RAM: 8GB
* Monitor resolution: 1920 x 1080
* Data drive throughput: 750MB/s supported by the motherboard’s front serial bus.
* Data drive size: 1TB recommended

### Minimum requirements for demo mode only

* OS: Windows 10
* RAM: 2GB
* Monitor resolution: 1920 x 1080

### Required Modules, Toolkits, Drivers, & Dependencies

NI Modules:

* Vision Development module
* FPGA Development module

NI Toolkits:

* Biomedical toolkit
* Report Generation toolkit

NI Drivers:

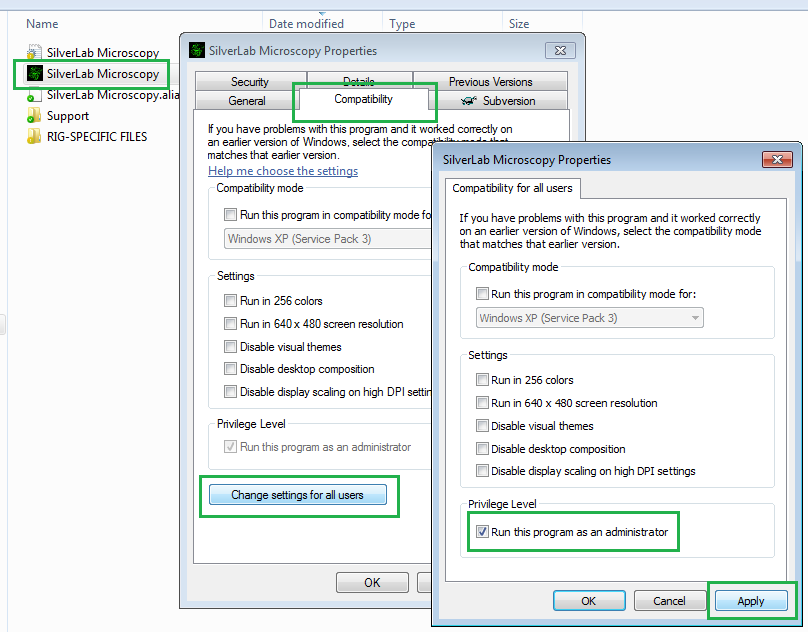
* CompactRIO
* NI FlexRIO
* NI R Series Multifunction
* NI System
* NI Serial
* NI DAQmx
* NI IMAQ
* NI RIO
* NI VISA

Other Source Code Dependencies

* Matlab
* WinPcap available at <https://www.winpcap.org/install/>

Executable’s Requirements

* WinPcap available at <https://www.winpcap.org/install/> (Normally already bundled in the executable’s installer)
* Full admin privileges (see screenshot below)



To find out what toolkits and modules are installed on a computer run the ‘NI License Manager’ software.

Most drivers already install by default during LabVIEW installation. If you are installing LabVIEW now, ensure you tick the above items to be installed if you see them available in the installation options list.

To find out what drivers and NI software versions are already installed, open the NI Measurement & Automation Explorer software (MAX) and expand the ‘Software’ category in the list. If any of the above components are missing download them from the National Instruments web and install them. It is important that you restart the computer ideally after each major component is installed, or at least when all installations have completed.

The software natively works with Scientifica translation stage, but support is available for Brucker MAMC and Olympus IX (z motor) stages. Please contact us for core drivers and process for setup.

When first starting the software there may be a pop-up asking for location of the Olympus ix or Brucker MAMC drivers, either select the default or hit ignore.

## User Login & Profiles

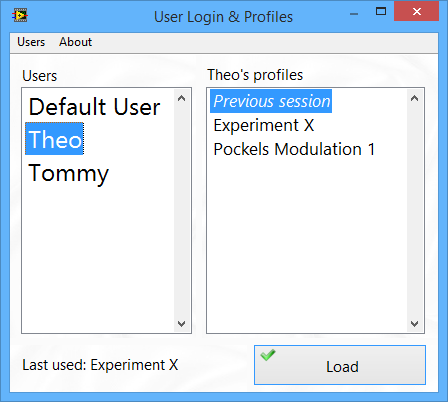
This is an important feature which eliminates interference between users and allows convenient switching between setup profiles. You can create an unlimited number of profiles under your own user account.

Settings like the triggers setup, zoom, dwell time, laser power, etc. are part of the user profiles functionality.

General system settings are not part of the user profiles functionality (e.g. hardware channels setup, AOL properties, encoder setup etc.) Such settings apply globally regardless of user profile loaded.

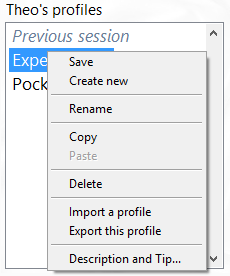
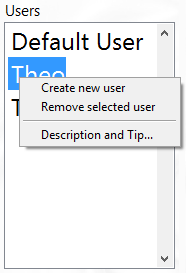
When the software starts, the ‘User login & Profiles’ window appears. There is no password login, so all you will normally need to do is double click on your name or on the desired profile to load (or alternatively select a profile and press the *Load* button).

Double clicking on your name instantly loads the *‘Previous Session’* without loading one of your user profiles. The *‘Previous Session’* is a built-in profile with special functionality which automatically stores the set up each user used last. This is useful for recovering unsaved changes. For example if you forgot to save some profile changes and another user logged in, your changes would be safely stored *‘Previous Session’* allowing you to recover them and save them in one of your profiles at a later time.



The User Profiles functionality keeps memory of which profile each user has used last. When you click on a user name, the last profile used by the selected user is shown in the space below the list of users as seen above.

The User Login & Profiles window includes a whole array of tools for creating or deleting users, and managing user profiles. It is possible to copy a profile from another user, rename or compare profiles, copy a subset of settings from one profile to another, import/export profiles among rigs and so on. All this, is available through right-click menus as seen below.

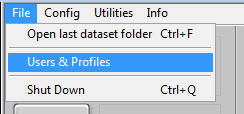


The User login & Profiles window can be accessed at any time when the rig is not acquiring data. There are two ways to open the User login & Profiles window. The simplest one is by clicking on the user name that appears in the information bar on the bottom left hand side of the main software screen. Besides acting as a quick access button for the User login & Profiles window, this area shows at a glance which user is logged in and the profile used.



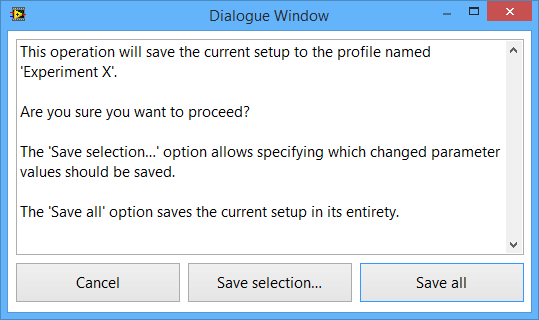


The User login & Profiles window can also be accessed through the File menu.



**Profile Saving & Profile compare**

Profiles other than the built-in *‘Previous Session’* profile are never saved automatically. To save a profile use the *Save* option from the right-click menu. This brings up the following window.



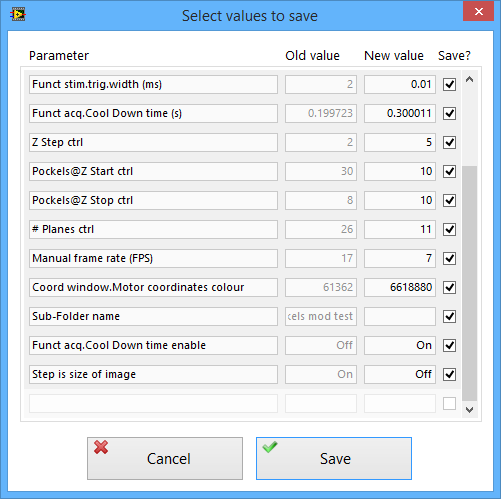
There are two options when saving a profile.

The *Save All* option saves the current setup to the selected profile in the list of profiles.

The *Save Selection…* option opens the following window which shows only changed parameter values and it has a dual purpose.

a) It allows to selectively save one or more parameter values.

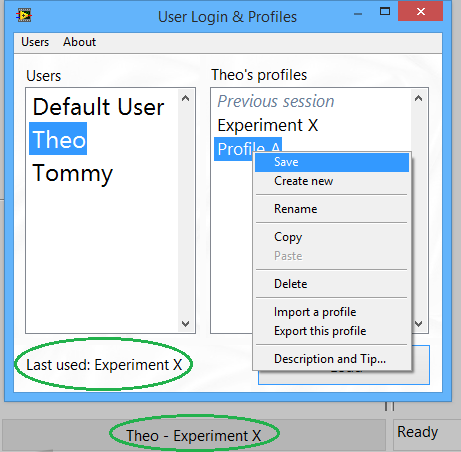
b) It allows comparing two profiles. For example if you load a profile and then select another profile in the list and use the Save selection function, the window seen below will show the differences between the two profiles. The values of the loaded profile will be shown in the “New Value” column. The values of the profile to be saved will be shown in the “Old Value” column.



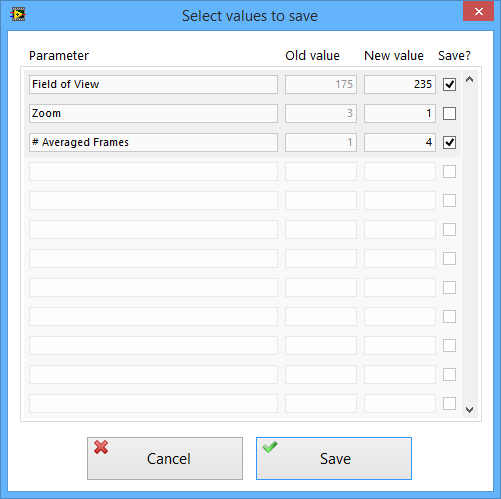
Untick the parameters whose value you do not wish to save. All parameters are ticked by default.

If you only wished to compare two profiles, press *Cancel* to abort the save operation.

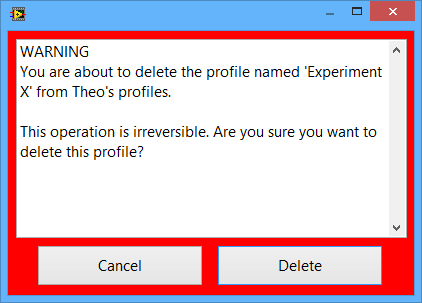
In the following example assume that we want to update “*Profile A”* with some settings from “*Experiment X”*. You would first need to load the source profile by double clicking on it in the Profiles list. In this example the source profile is “*Experiment X”.* The loaded profile’s name is shown in the two areas encircled below. Next, you would click the destination profile in the profiles list to select it and then right click on it and select *Save*, and then press *Save selection*.



In this example the two profiles have three settings which are different as seen in the following window. If we were to copy the “Field of View” and the “# Averaged Frames” setting from *Experiment X* to *Profile A* but didn’t want to copy the Zoom setting we would untick the Zoom setting as shown.

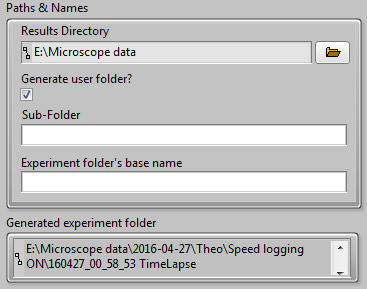


Critical operations include interlock dialogues to avoid accidental deletion of profiles, users, or accidental data overwrites.



## Output Directory Format

The ‘Paths & Names’ group of controls allows setting up the output directory format to suit each user’s and each experiment’s specific requirements. This group of controls and indicators is located in the main screen.



**Results Directory**

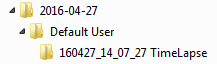
The results directory is the top level folder within which all data is saved. This setting applies globally, affecting all rig users.

To avoid problems with data recording, the results directory must be located on a fast hard disk drive capable of data throughputs of 6Gb/s or better and connected to a ≥6Gb/s front serial bus on the computer’s motherboard.

**Generate user folder?**

Tick this option to generate all data within a user-specific subfolder. An additional folder level is added to the output path. The name of the folder is the name of the user currently logged in. if no user is logged in the folder’s name is *Default User*.

Examples of possible path formats:



User folder is used

D:\Microscope data\Daily folder\User\ Experiment folder

D:\Microscope data\Daily folder\User\Subfolder\ Experiment folder

User folder is not used

D:\Microscope data\Daily folder\ Experiment folder

D:\Microscope data\Daily folder\Subfolder\ Experiment folder

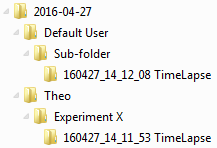
**Sub-Folder**

When a sub-folder name is specified, all experiment folders are generated within that sub-folder. This may be used to keep your data organised by experiment type.

For example a separate user-profile can be created for each type of experiment and set up to output all its data within a folder named after the type of experiment.

No sub-folder is created when left blank.

Examples of possible path formats:



Sub-folder is used

D:\Microscope data\Daily folder\Subfolder\Experiment folder

D:\Microscope data\Daily folder\User\Subfolder\ Experiment folder

Sub-folder is not used

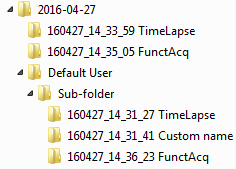
D:\Microscope data\Daily folder\ Experiment folder

D:\Microscope data\Daily folder\User\ Experiment folder

**Experiment folder's base name**

When an experiment folder name is specified, all generated experiment folders show this name. When left blank the folder name automatically shows the type of acquisition (e.g. TimeLapse, FunctAcq, etc.)

Examples of possible path formats:



Base name used

D:\Microscope data\Daily folder\ Timestamp\_Base name

D:\Microscope data\Daily folder\User\ Timestamp\_Base name

D:\Microscope data\Daily folder\Subfolder\ Timestamp\_Base name

D:\Microscope data\Daily folder\User\Subfolder\ Timestamp\_Base name

Base name not used

D:\Microscope data\Daily folder\Timestamp\_imaging type

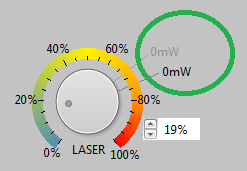
D:\Microscope data\Daily folder\User\Timestamp\_imaging type

D:\Microscope data\Daily folder\Subfolder\Timestamp\_imaging type

D:\Microscope data\Daily folder\User\Subfolder\Timestamp\_imaging type

## Calibrated Power Display

There feature in the software showing the power before the objective (back-aperture) and after the objective. The grey value seen in the circle is the back-aperture power and the black one is the calibrated power after the objective. When no calibration exists for a specific combination of objective and laser wavelength the values are identical and a tip-strip indicates that no calibration exists.



For this feature to work it is first necessary that both the Pockels (back-aperture) and the objective are calibrated at the current wavelength. That’s done using the Pockels calibration wizard from the menu shown below.

## Pockels Calibration

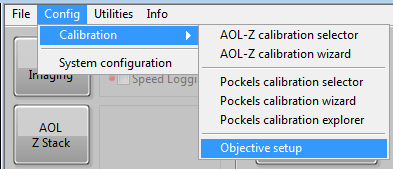
Through the *‘Pockels calibration wizard’* it is possible to calibrate the back-aperture power and any number of objectives all at the same go. Calibration curves for new objectives at different laser wavelengths can be added at any time by running the wizard again.

New objectives can be set up to display the correct “power after the objective” by using the ‘*Objective setup’* option at any time.

The ‘*Objective setup’* option adds the new objective measurements in the relevant section of the existing Pockels calibration file (an example of which can be seen near the end of this document).

Objective setup involves just 5 measurements and it is much quicker to allow quick setup of new objectives.

The Pockels calibration file can hold calibration curves and scaling data for any number of laser wavelengths and objectives.



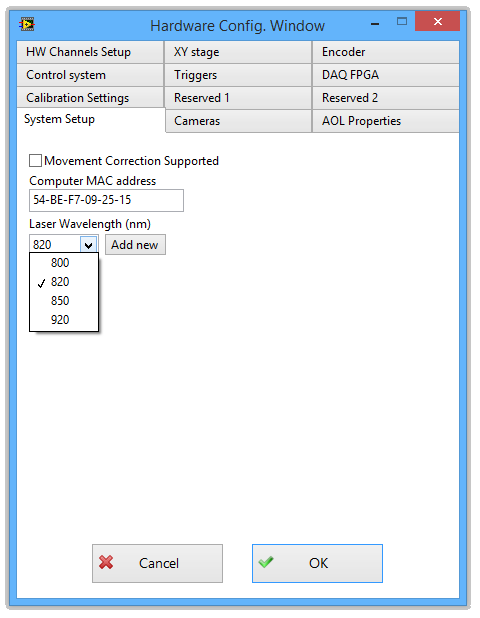
The software only uses the back-aperture calibration curve to calibrate the laser power. That is a 21-point curve. Pockels (back-aperture) calibration takes around 10 minutes. The laser-power measurements taken after the objective through the ‘*Objective setup’* option, are not used to calibrate the laser power as such. This information is simply used in the calculation of the average power loss introduced by an objective which can then be used to calculate the value displayed in the “Power after the objective” indicator. That’s the black indicator encircled in the first screenshot. This calculation is based on the assumption that the power loss introduced by an objective will be a constant percentage.

Objective setup is only allowed if a pockels calibration curve already exists in the Pockels calibration file for the current laser wavelength.

When a calibration is performed, certain conditions are also saved to the Pockels calibration file.

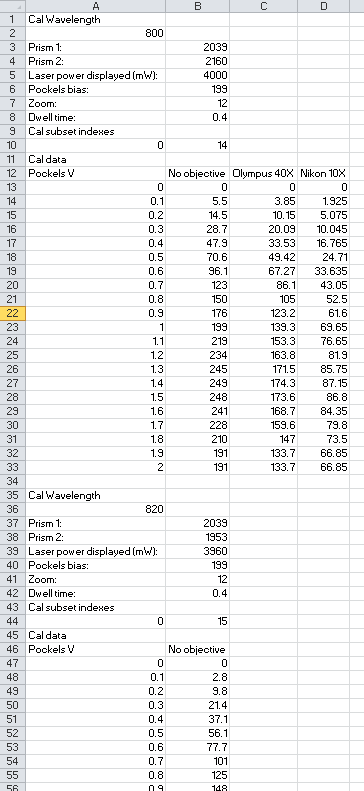
When Pockels calibration has been carried out for more than one laser wavelength of interest the software will automatically switch to use the correct calibration when the wavelength is switched.

Wavelengths of interest can now be added and switched through the system configuration window as seen below.

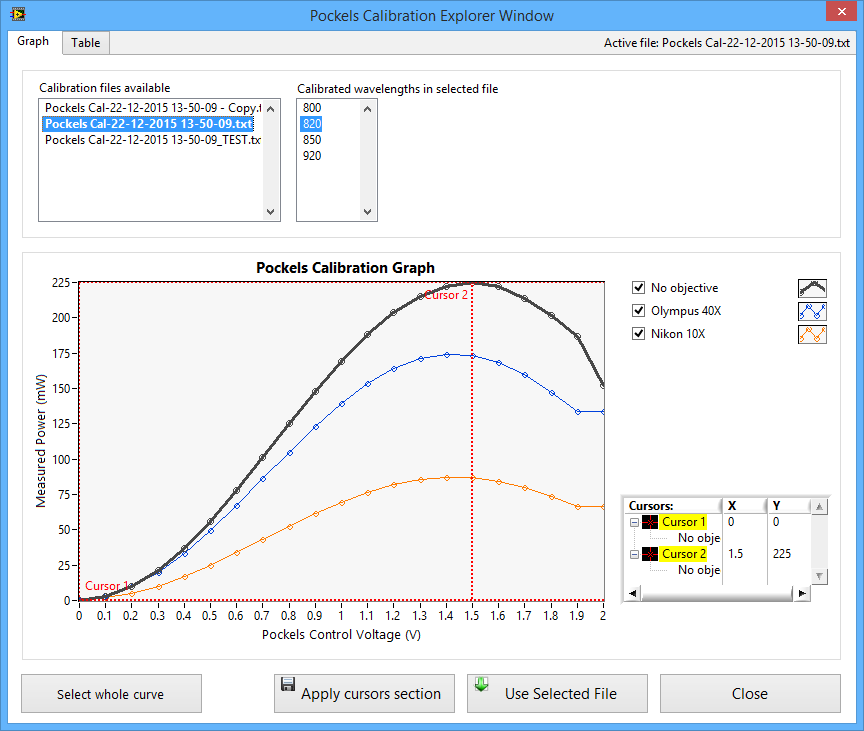


The wavelengths list seen above will only show wavelengths for which Pockels calibration has been carried out, plus it will hold just a single “Uncalibrated” wavelength.

This allows running the rig an “Uncalibrated” wavelength (any). When running the rig with an uncalibrated wavelength a built-in default Pockels calibration curve is used just to allow the rig to run. In that case the software will appropriately inform the user that until Pockels calibration has been carried out the error in the displayed output laser power can be over 15%.



Example of Pockels calibration file structure.



The Pockels calibration explorer window offers instant top level access to any existing calibration file (more than one calibration files can exist if necessary for development, rig commissioning, and debugging purposes).

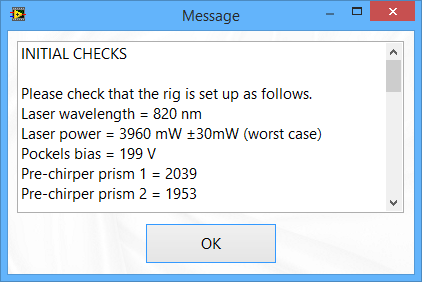
Using this tool it is possible to view and compare information within a single file and among separate files.

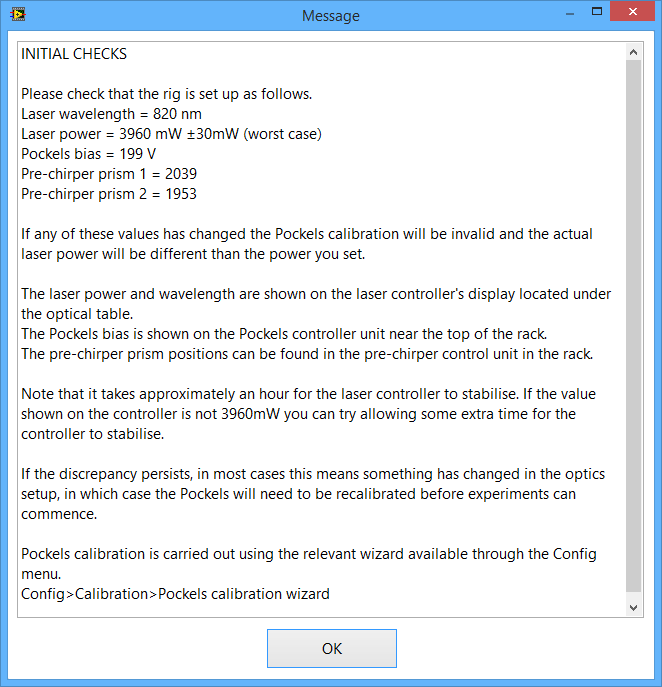
There is a lot more peripheral functionality around the above upgrades which I am not going to mention here. All you need to know is that it has been designed to be intuitive and it will offer lots of tips and detailed information through pop up dialogue windows as and when needed.

Once you have the opportunity to use the various new features available in the software (not just those mentioned here) I would really appreciate it if you could get into the habit of providing feedback and suggestions for improvements.

## Calibration Checks during Software Initialisation

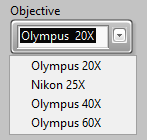
Each time the software is started, the following window prompts the user to check that those conditions are still valid. There are also several other interlock dialogues built in the calibration functionality which will pop up as needed, ensuring that the rig setup is valid and that the user do not accidentally cause an existing calibration to become invalid.



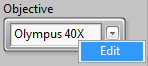


## Managing Objectives Setup

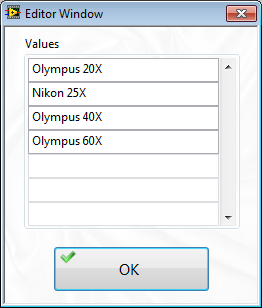
It is possible to make new objectives available in the drop-down menu.



This is done through the *‘Edit’* function available by right-clicking on the *‘Objective’* control.



The *‘Edit’* function brings up the editor window seen below, through which objective names may be added, deleted, and modified.



Objectives editor window

## Stimulus Trigger

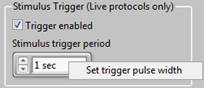
The stimulus trigger can be enabled and set up to meet the needs of each experiment. The settings are part of the user profiles functionality allowing users to create multiple profiles with the stimulus trigger set up differently in each one. Each type of experiment can have its own profile.

During live imaging protocols the stimulus trigger (if enabled) fires continuously at the specified rate.

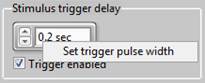
During functional acquisition protocols the stimulus trigger fires at the start of each “trial”. A delay can be added to offset the trigger reference to the start of the trial.

The stimulus trigger pulse width can be set up individually for live imaging and functional acquisition.

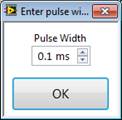
Both pulse widths are set to 0.1ms by default, but this can be changed intuitively by right clicking on the *‘Stimulus trigger period’* or the *‘Stimulus trigger delay’* delay control and entering a new value in the pop-up window, as shown below.



In the *‘Miscellaneous’* tab page of the main screen.



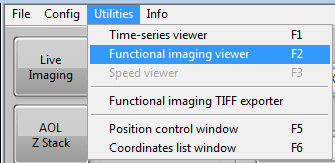
In the ROI entry screen.



Allowed values are 0.1ms to 1000ms.

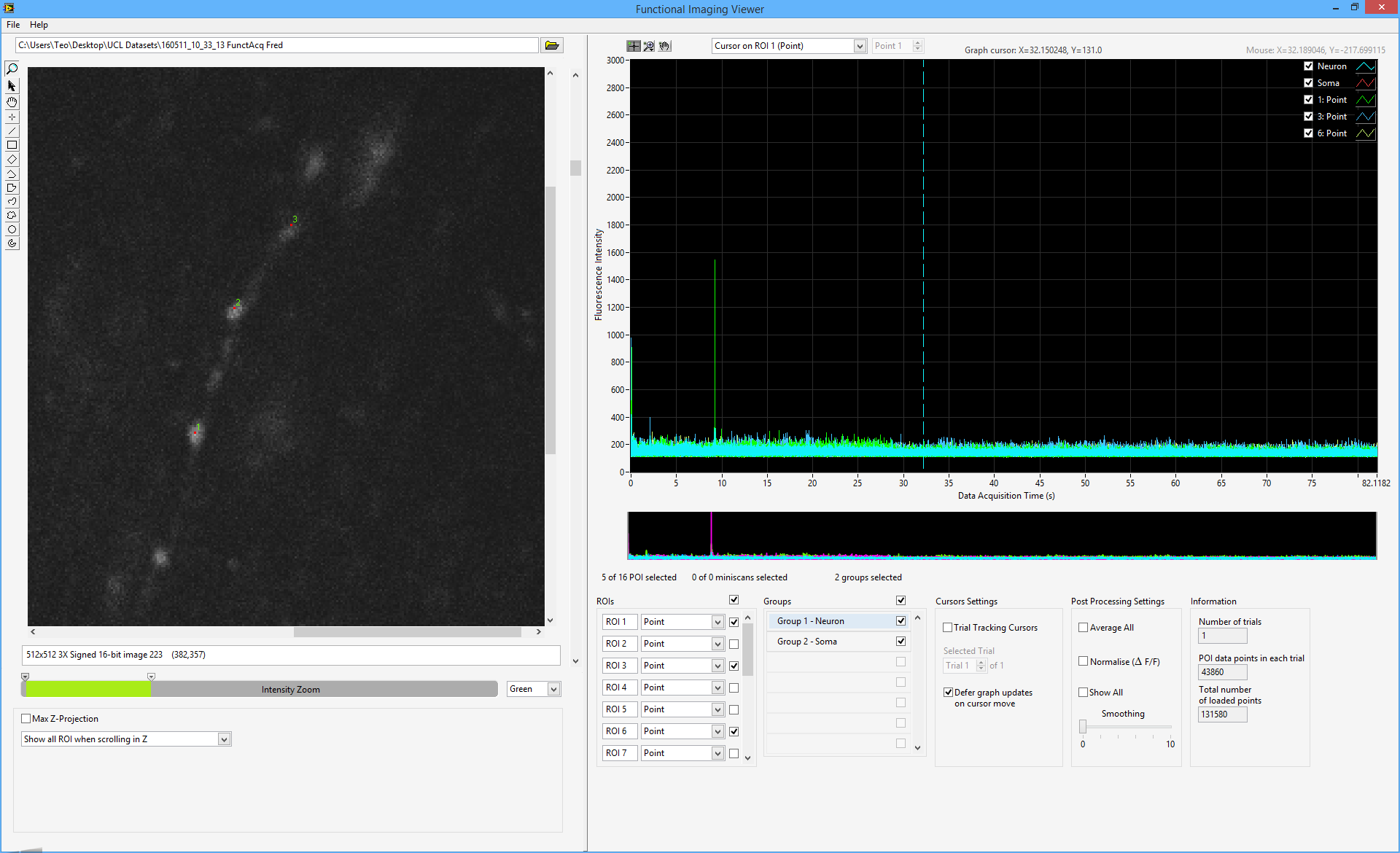
## Functional Imaging Viewer

The functional acquisition data viewer opens automatically each time a functional acquisition protocol has completed. It displays the acquired functional acquisition data. The viewer can be accessed at any time via the relevant menu option (see image below), or by simply pressing the F2 key on the keyboard.



Opening the functional imaging viewer through the user menu

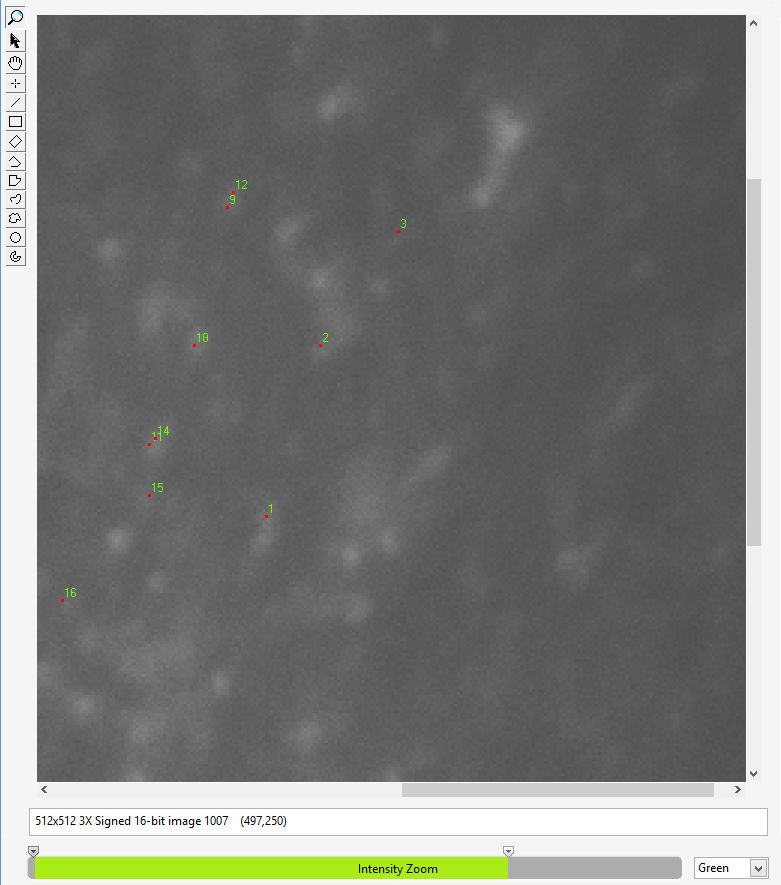
The image display on the left shows the Z-stack which was acquired before functional acquisition. Use the vertical scroll bar to scroll through the stack of images. The location or shape of acquired ROI is shown on the images.



Functional imaging viewer

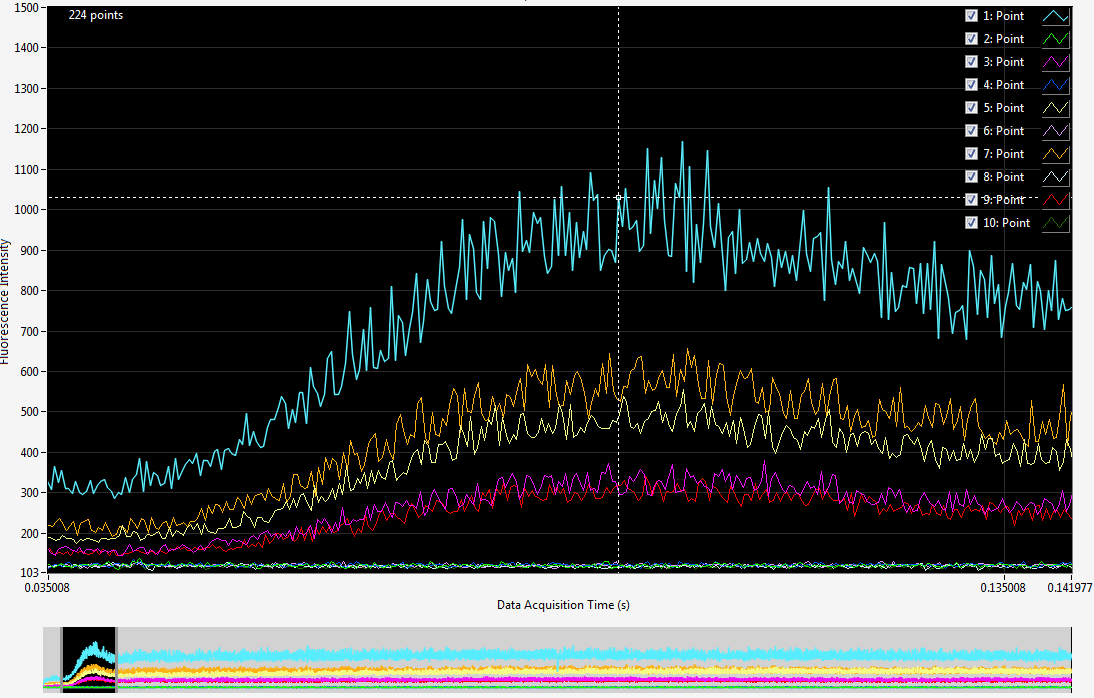
Use the palette (located to the left of the image display) and the image’s right-click menu for common operations. To zoom in select the magnifier glass in the palette and click on the image. To zoom out, hold the Ctrl button on the keyboard and click on the image.

Use the *‘Max Z-Projection’* check box to toggle the max Z-projection on/off. If the max Z-projection is enabled and the vertical scroll bar is used to transverse the Z-stack images, the max Z-projection is disabled automatically.



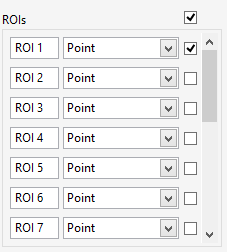
Max Z-Projection

The viewer’s right hand side graphs display data traces for the selected ROIs and ROI groups. The narrow graph at the bottom is an overview of the whole experiment and incorporates navigation cursors which can be used to select the region of data to be displayed in the main graph as illustrated below.



If the dataset contains “trials”, enable the *‘Trial Tracking Cursors’* option to view a single “trial” in the graph. Once this option is enabled you can navigate through the available trials using the *‘Selected Trial’* control.

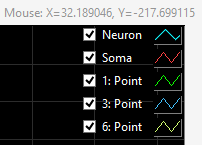
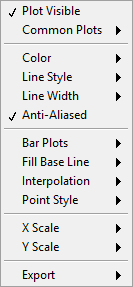
The *‘ROIs’* control lists all acquired ROI. Tick the box next to one or more ROI to add their data to the graph. Unticking the box at the top of the ROI control deselects all ROI removing their data from the graph. Note that adding too much data to the graph can slow down the viewer.



Selecting ROI to display in the graph

The ‘Average All’ option averages all traces in the graph including ROI groups, resulting in a single averaged trace.

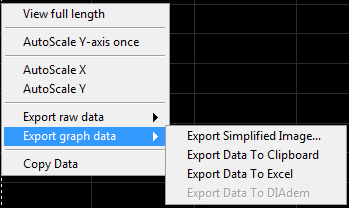
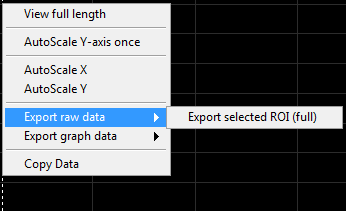
The names of individual ROI and ROI groups displayed in the graph are shown in the graph’s plot legend. The plot legend also indicates the type of ROI (e.g. point, miniscan, surface etc.). The plot legend scales automatically to show as many elements as the total number of traces displayed in the graph. If the number of elements is large, the legend stops auto-sizing and makes its scrollbar visible, allowing to scroll through the list of elements. The individual check boxes seen in the graph legend are a useful feature allowing to quick-toggle trace visibility without having to remove the trace’s data from the graph. This is helpful in identifying which trace belongs to which ROI or ROI group when trace colours are similar. It is also possible to change the colour, size and style of any individual trace.

Graph’s plot legend, and legend’s right-click menu

## Exporting Data from the Functional Imaging Viewer

Right-clicking on the main graph in the functional imaging viewer brings up its shortcut menu shown below.

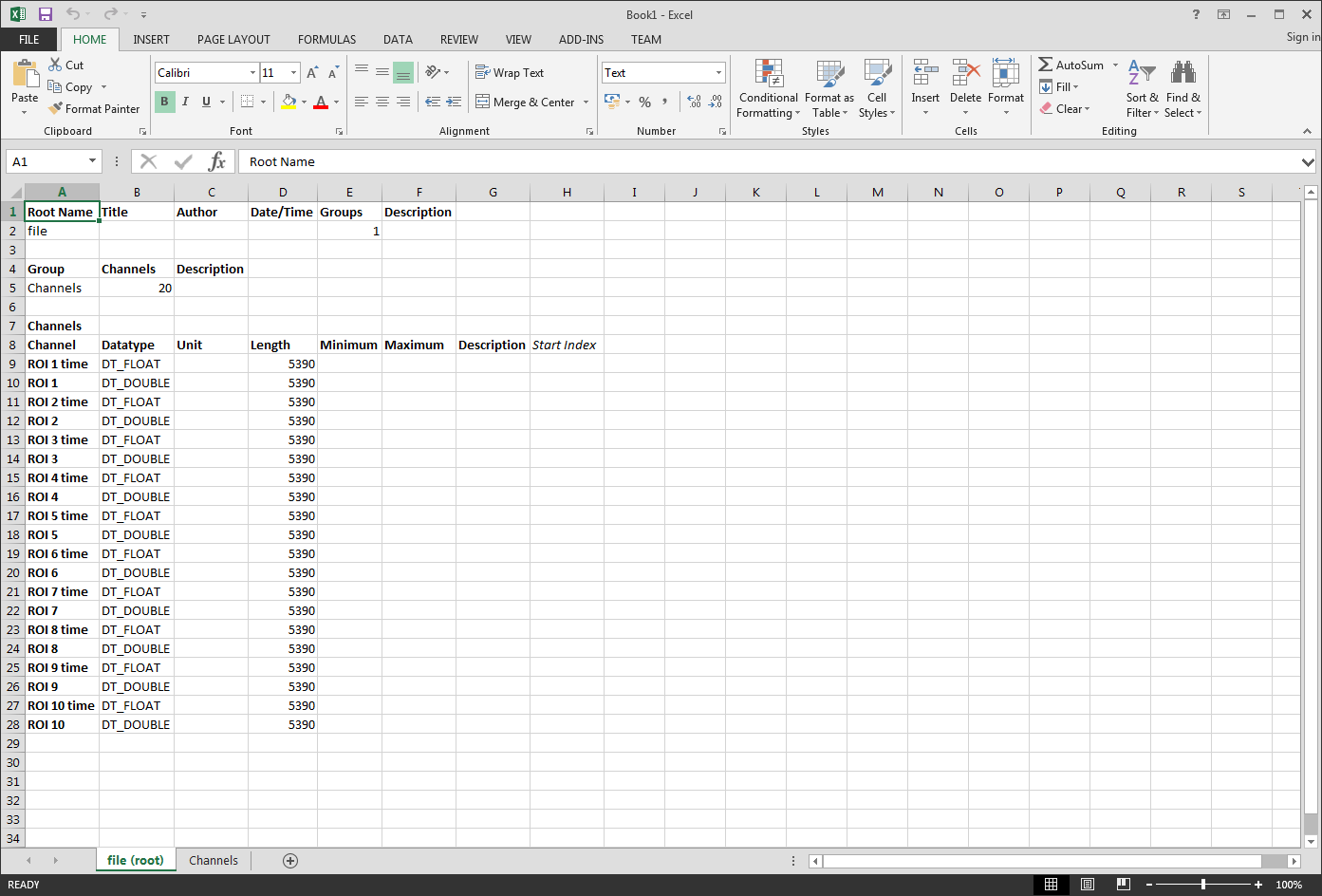
 

Graph’s menu

An important detail to notice is the “Export” group of options. Traces visible in the graph can be exported as an image, copied to the clipboard, exported to Excel, or to NI Diadem if that tool is installed. Note that the graph data export functions will only export graph data which is typically of lower resolution than the raw data due to data decimation applied to the graph. Data decimation is heavier when the full length of the dataset is displayed.

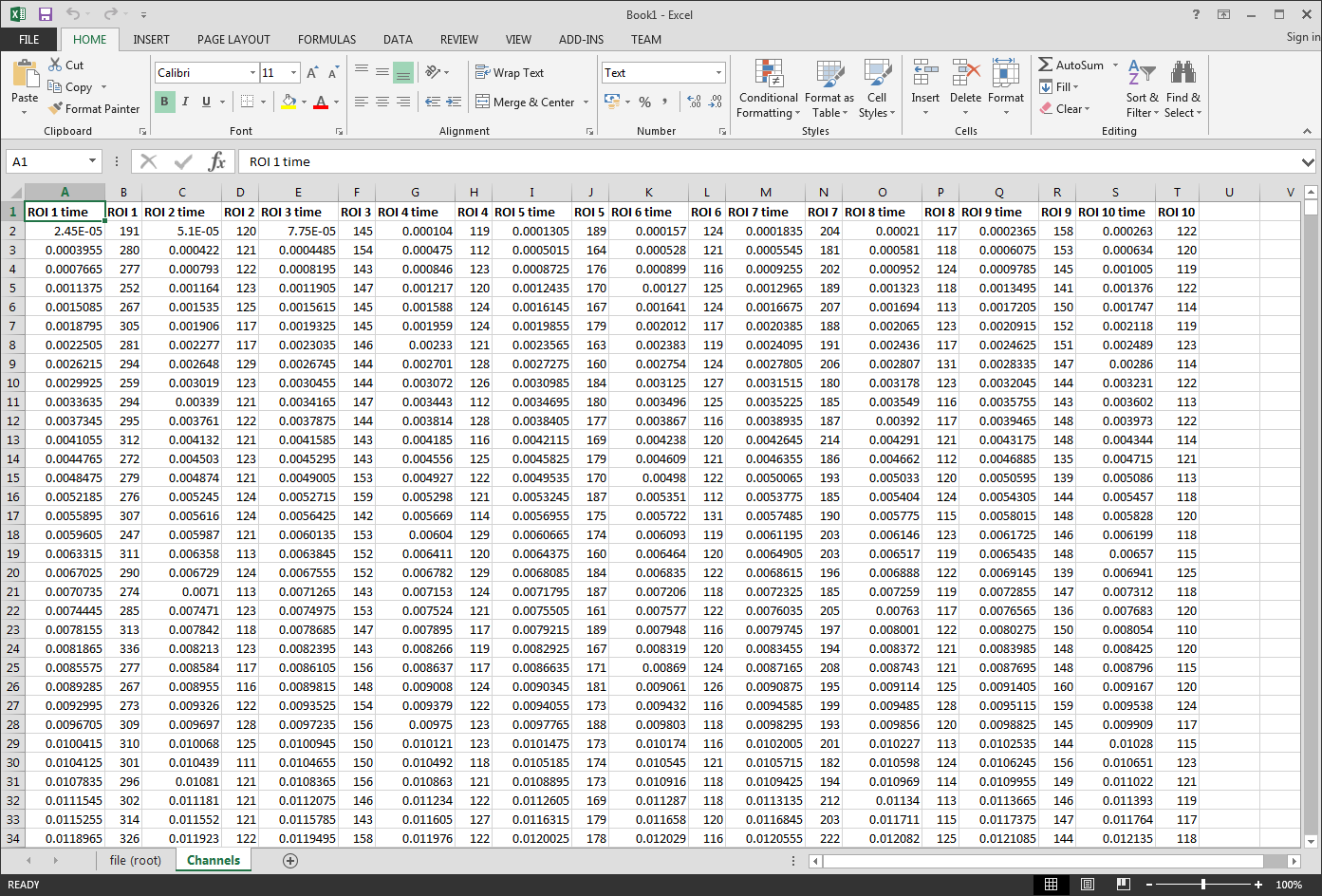
If you wish to export the raw data use the ‘Export Selected ROI’ function instead. This function exports the full raw data of all ROI that are visible in the graph. It generates a TDMS file which can be opened in Excel.

To view TDMS files in Excel install the required add in from <http://www.ni.com/example/27944/en/>



TDMS file header page

The first page of the TDMS file shows the number of channels, channel names, and number of data points in each channel. The second page contains the data in the format shown below. Notice that each channel consists of two columns, i.e. “time” and “intensity”.



TDMS file data page

The file can be manually saved as an Excel workbook.

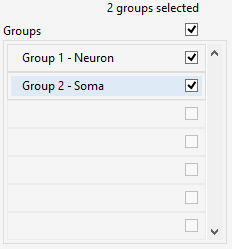
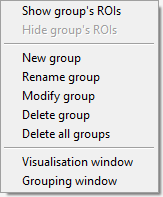
To find out more about the NI TDMS file format go to <http://www.ni.com/white-paper/3727/en/>

## ROI Grouping

Use the *‘Groups’* control in the functional imaging viewer’s panel to create, modify, and visualise ROI groups. Each ROI group appears as a single trace in the graph which is the average of all ROI the group consists of. Tick an existing group to make its data visible in the graph.

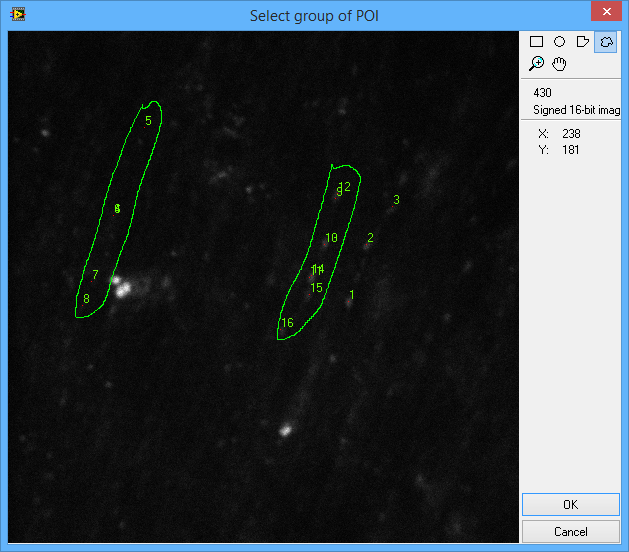
*Note: Adding too many groups and/or ROI to the graph can impact the graph display’s response time.*

Right clicking on the *‘Groups’* list brings up the following context-sensitive menu. Only applicable options will be available. Non-applicable options will appear as greyed out.

Managing ROI groups / Right-click menu

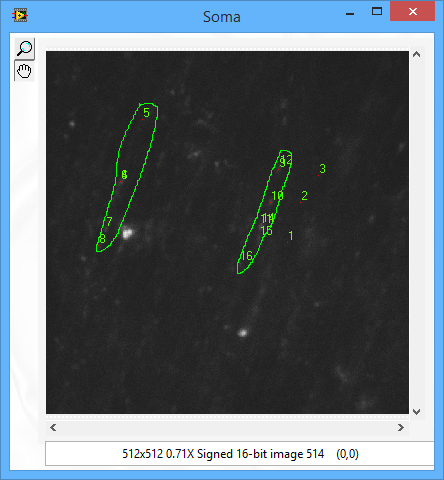
The *‘New group’* and *‘Modify group’* options open the group editor window shown below. This window allows specifying groups visually by drawing a line around them. Use the Ctrl key on the keyboard to make multiple selections as necessary. In the image below all encircled ROI form a single ROI group. A drawn group boundary can be deleted by clicking on it and pressing the Delete button on the keyboard.



ROI group entry window

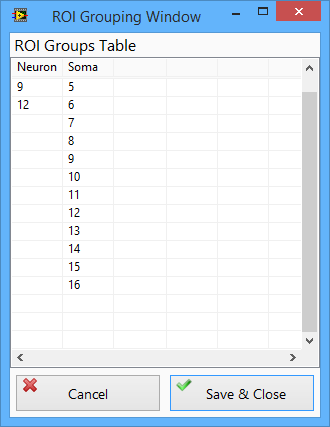
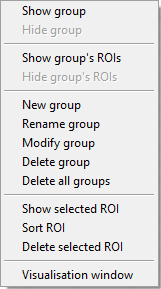
The *‘Show group’s ROIs’* and *‘Hide group’s ROIs’* options add/remove the individual ROI traces of a group to the graph by automatically ticking the group’s ROIs in the *‘ROIs’* list.

Use the ‘Visualisation window’ option to open the group visualisation window shown below. This window is similar to the group editor window but only allows visualising groups. What’s more this window can be left open. Each time you click on a different group in the ‘Groups’ list this window will refresh automatically to display the group you selected. This allows on-the-fly comparisons among groups.



ROI group visualisation window

The ‘Grouping window’ option opens the window shown below, which offers an overview of all groups, listing the ROI contained in each group. Through its right-click menu this window allows performing all of the operations mentioned previously. In addition it gives the ability to modify manually the ROIs included in each group. This can be done by typing/modifying ROI numbers directly in the table and/or by using the *‘Delete selected ROI’* option in the table’s right-click menu. Manual ROI entry is automatically limited to valid numbers only. Attempts to enter invalid values are prevented.

ROI grouping window / Right-click menu

Clicking on a group (i.e. column) in the table causes the group visualisation window (if open) to show that group.

Use the *‘Show group’* and *‘Hide group’* option to add or remove a group’s trace from the graph.

The *‘Show group’s ROIs’* and *‘Hide group’s ROIs’* options add/remove the individual ROI traces of a group to the graph by automatically ticking the group’s ROIs in the *‘ROIs’* list.

The *‘Show selected ROI’* option adds the selected ROI to the viewer’s graph. Multiple selections are allowed.

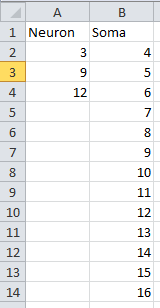
The *‘Delete selected ROI’* option can be used to delete one or more ROI form one or more ROI groups. Multiple selections are allowed.

The *‘Sort ROI’* option sorts all ROI numbers in ascending order.

The ROI grouping window can be resized at will and includes size & position memory.

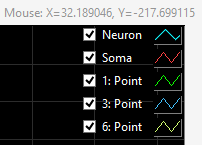
Pressing the ‘Cancel’ button rejects any changes and closes the ROI grouping window. If changes exist the user is given the option to save or reject the changes.

ROI groups are saved with the data. The ROI groups’ setup is saved in a subfolder named “Setup” inside the dataset’s folder. The XML file named “ROI Groups setup.xml” holds advanced information like group boundary overlays. This file is for internal use. The simple text file named “ROI Groups info.txt” holds the basic groups information and can be used to carry out further group data post processing in third party software.



ROI groups info file (ROI Groups info.txt)

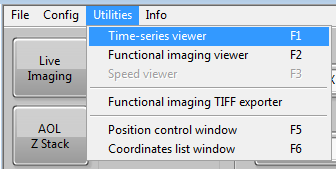
The names of ROI groups displayed in the graph are shown in the graph’s plot legend. The legend shows ROI groups on top of all individual ROIs.



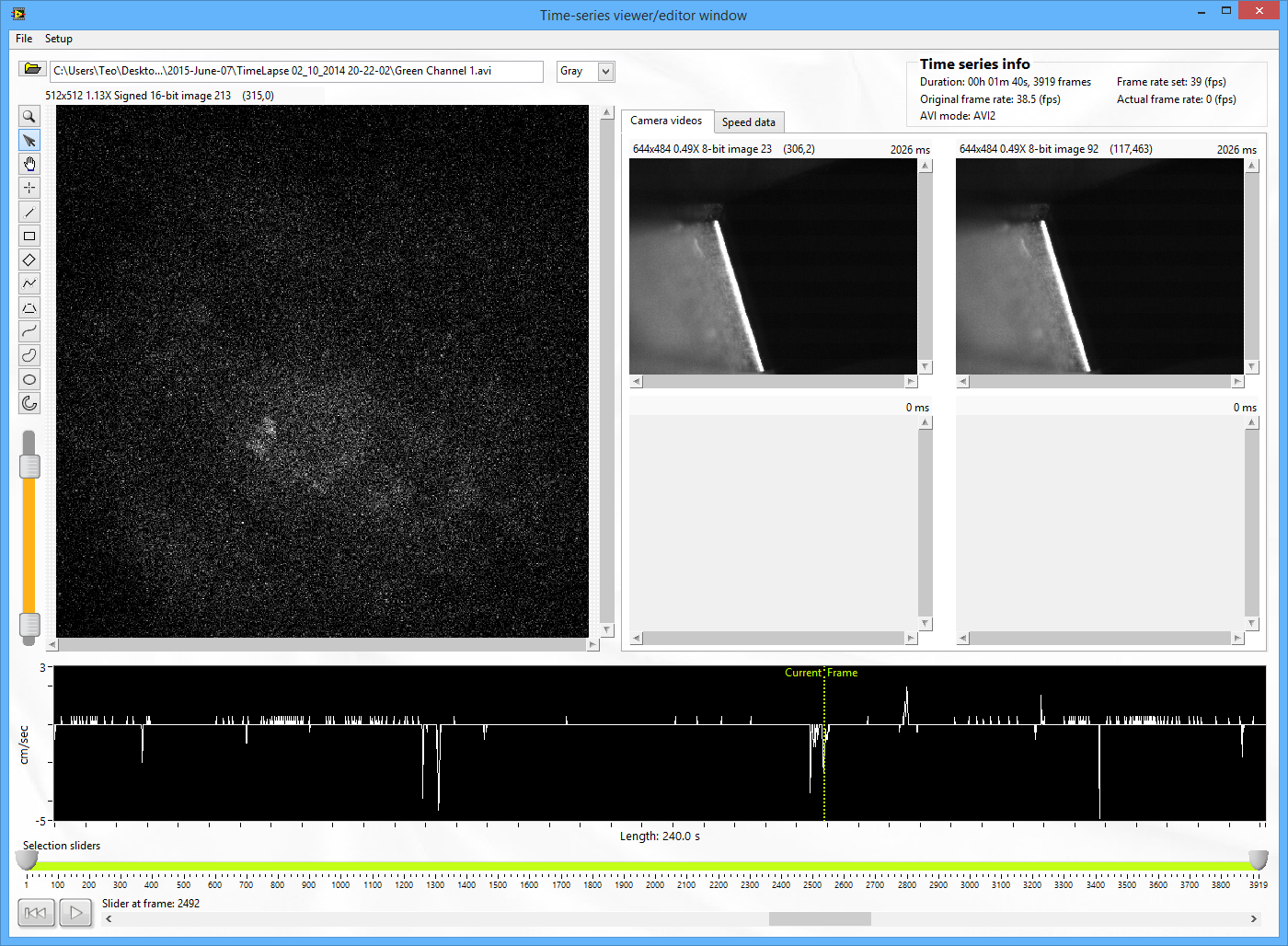
ROI groups in the graph’s plot legend

## Time-series Viewer & Editor Window

The Time-Series viewer can be used to visualise time-series data which along with the microscope recordings it can include animal speed data and up to four camera video recordings. The viewer can be accessed at any time via the relevant menu option (see image below), or by simply pressing the F1 key on the keyboard.

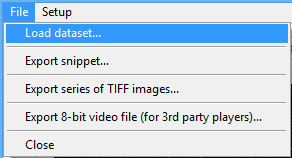


Opening the time-series viewer through the user menu



Time-Series viewer & editor window

Through this window it is also possible to play back just a section of the recording, export snippets, export series of TIFF images, as well as exporting 8-bit composite videos for playback in third party video players. Most of these options are available through the window’s main menu seen below.

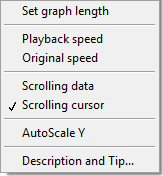


Use the *‘Selection Sliders’* control to specify a section of the recording. Moving any of these two sliders automatically refreshes the images to show brain activity and camera videos at the position of the slider.

It is also possible to drag the green slider in the speed data graph. All displays will refresh synergistically.

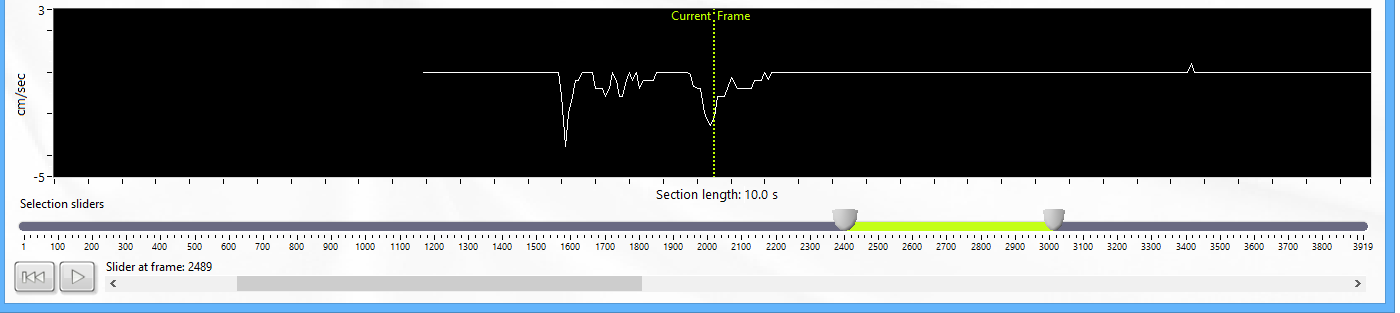
Pressing the *‘Play’* button starts playback from the position of the leftmost slider. Playback stops at the position of the rightmost slider.

Right clicking on the speed graph brings up the following menu.



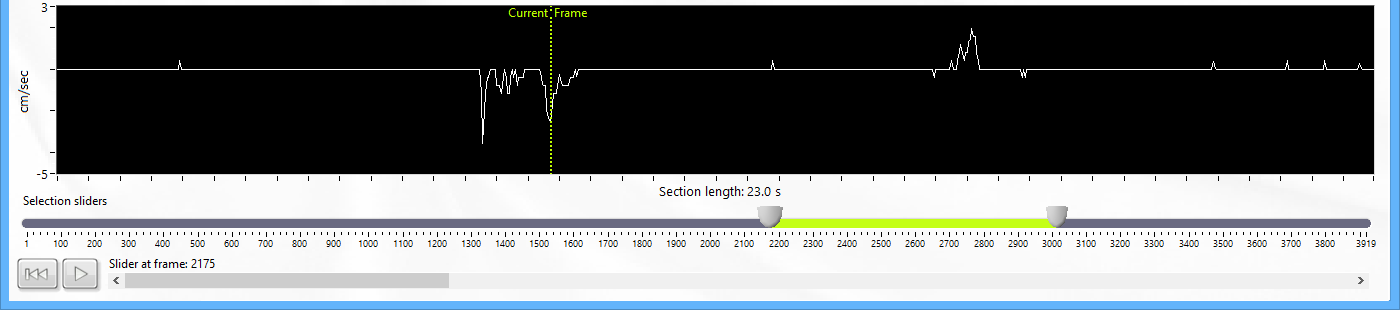
Through this menu it is possible to adjust the length of the speed graph, modify the playback speed, auto scale the Y axis, and change the scrolling behaviour of the speed graph.

The *‘Scrolling data’* option is ideal for viewing detail in the speed data and for investigating the data frame-by-frame. When this setting is selected the graph length is fixed to a user-configurable setting (10 seconds in the image below). The green cursor is fixed to the centre of the speed graph and during playback the data scrolls towards the left.



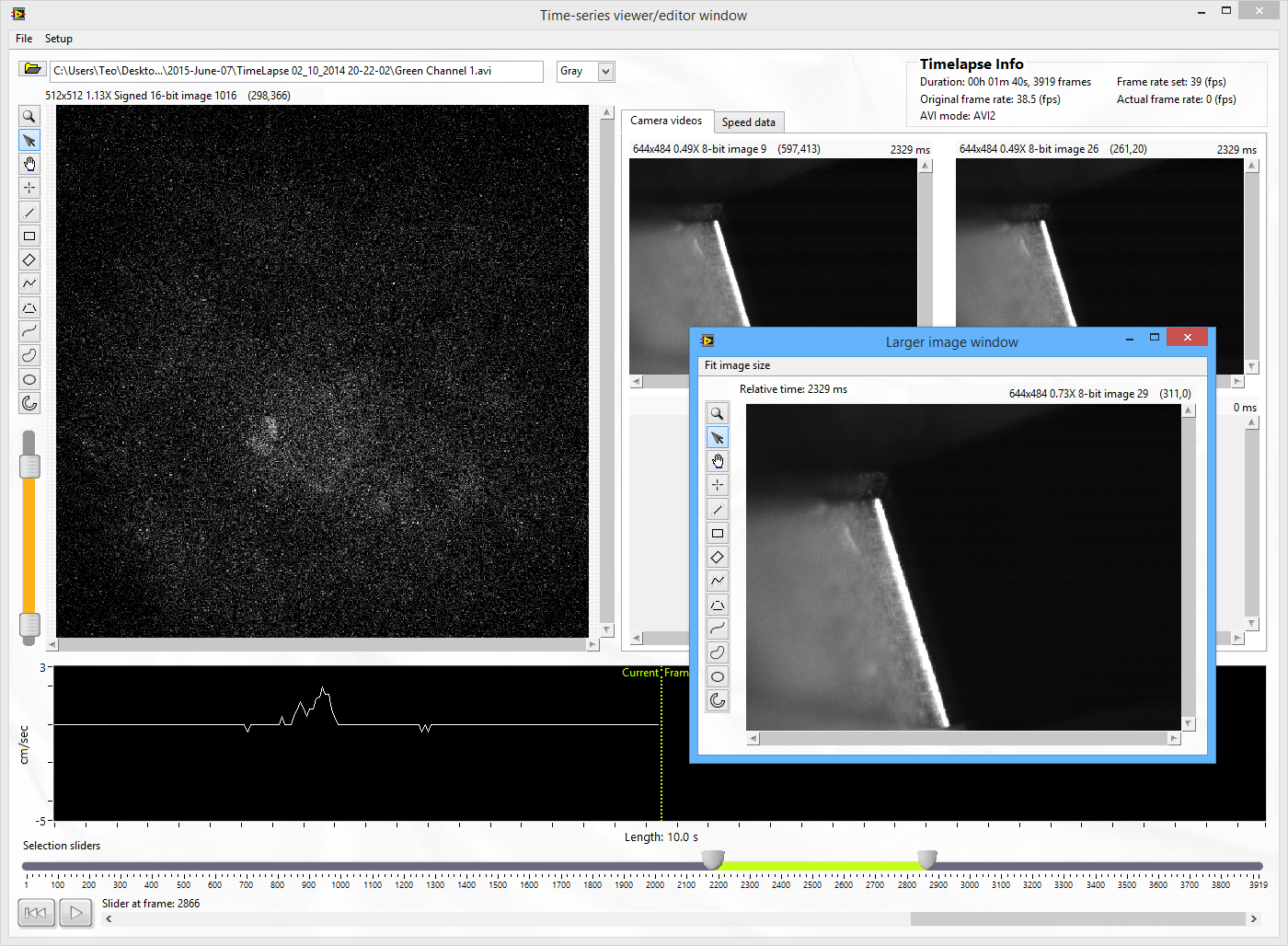
Mode = Scrolling data

The *‘Scrolling cursor’* option is particularly useful for glancing at the entire set of speed data. In this mode the speed graph will display the entire section of selected data. The graph’s X-axis label shows the length defined by the selection sliders. During playback it is the green cursor that scrolls instead of the data. The cursor scrolls towards the right.



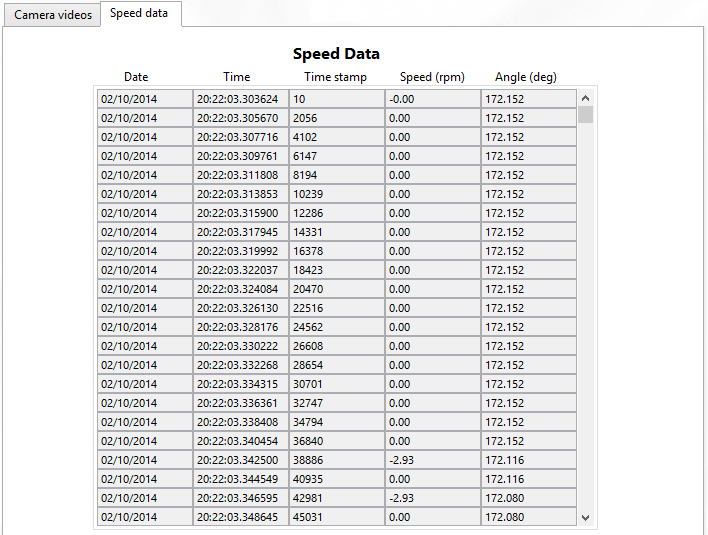
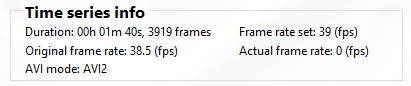
Mode = Scrolling cursor

Each one of the four camera videos can be opened in its individual pop-out window by dragging a video image out of its embedded display. The pop-out window can be resized, minimised to the task bar, or maximised to fill the entire monitor. Double clicking on an embedded video display opens its pop-out window maximised.



Pop out camera video windows

The *‘Speed data’* tab page shows the data in the speed data file. The timelapse info section shows information about the loaded dataset and its play-back speed.

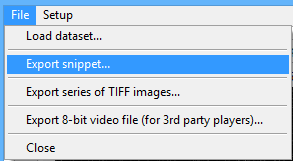
 Speed data table and time series info

The grey scroll bar at the bottom of the window can be used to scroll through the selected portion of the dataset manually.



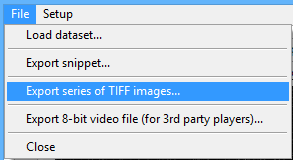
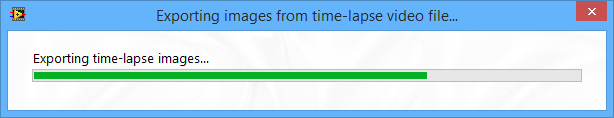
Manual scrolling

The *‘Export snippet…’* option allows exporting sections of the dataset saved in a format identical to the original dataset. Time series datasets can be very large, consuming a lot of HDD space while only fractions of each dataset may contain useful information. Using the ‘Export snippet…’ it is possible to export just the useful section(s) in their original data format, and then discard the original dataset.



Exporting dataset snippets

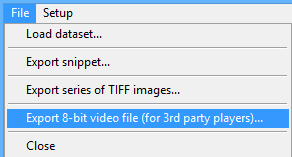
The export series of TIFF images allows exporting microscope data frames as individual TIFF files. Before you use this function, use the *‘Selection sliders’* control to specify which section of the dataset to be exported.

Exporting TIFF images from microscope data

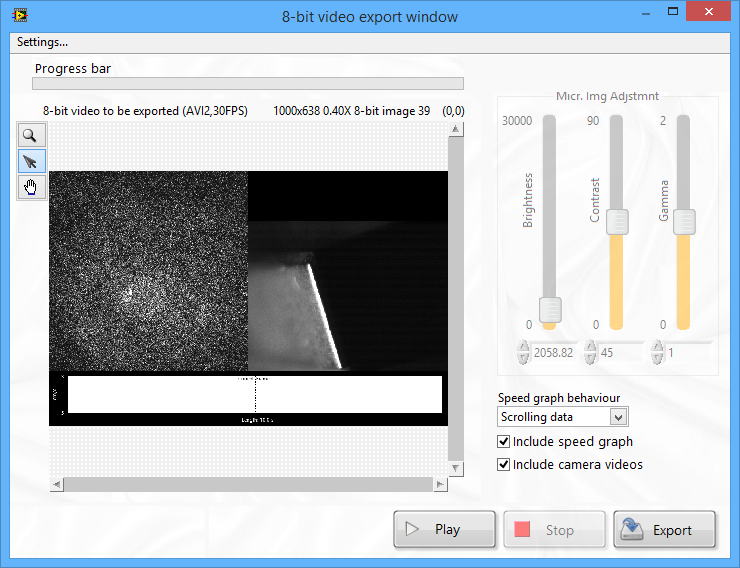
## Exporting Time Series Videos for 3rd Party Players

Use the following option from the Time Series viewer’s menu to generate time series videos which can be played in third party players. These videos can optionally include the speed graph and up to four videos recorded by cameras.



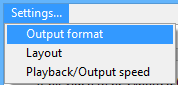
In the exporter window press the *‘Play’* button and make any desired visual adjustments during video playback. Once ready, press the ‘Export’ button to generate the video.

If in the Time Series Viewer window you selected just a section of the dataset (using the *‘Selection sliders’*) you will be given the option to export the video from the selected section or from the entire dataset.

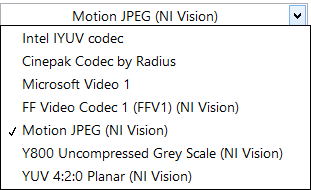


Exporting time series videos for 3rd party players

In the 8-bit video export window’s menu select *‘Output format’* to specify the format of the output video.

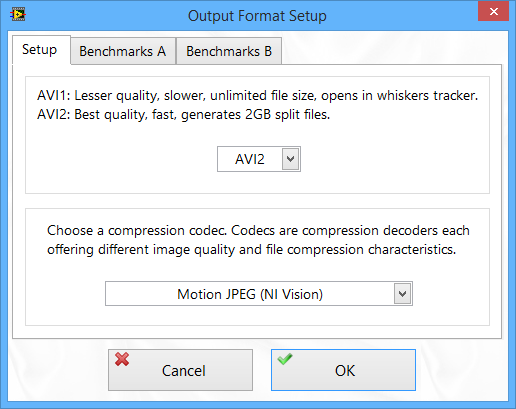


This brings up the output format setup window. The available video standards are, AVI1 and AVI2. The codecs drop-down menu automatically updates to show all available National Instruments and third-party codecs (**Co**nversion **Dec**oders).



The quality of the output video will depend mainly on the codec used. All codecs should be available to use with the AVI2 standard. Most codecs will not work with the AVI1 standard. Typically you will be using the AVI2 standard which is the latest standard and has some benefits like better image quality, and faster encoding.

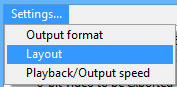
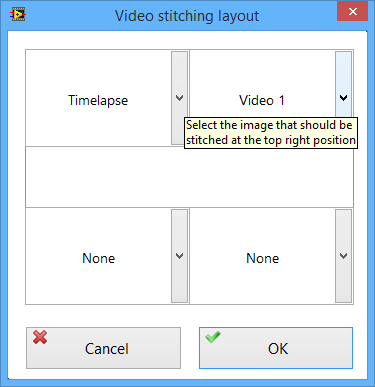
One limitation of the AVI2 standard (at least as implemented in NI LabVIEW) is that it cannot generate videos larger than approximately 2GB. As a workaround to this issue, the exporter will generate split video files when the exported dataset is too long. Each split file (except for the last one) will be 2GB. In that case, a set of generated videos will span the entire length of the exported dataset section.



Output format setup

The default and seemingly most optimal codec from those shown above is the *“Motion JPEG (NI Vision)”* codec. Note that depending on the selected codec the size of the output video can vary dramatically. The two additional tabs in the output format setup window hold useful benchmarking information about video quality and output file sizes, which can help decide which codec to use.

The *‘Layout’* option brings up the video stitching layout window, offering a degree of user-configurability for the output video. The speed graph will appear in the middle section (if the ‘Include speed graph’ option was previously ticked).

Video stitching layout

Use this window to include any of the available camera videos and position them as desired.

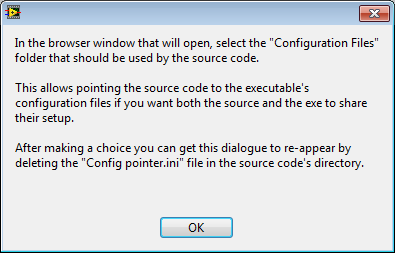
## Setting up the Source Code’s Configuration Files Folder

It is possible to set up the source code to use the executable’s configuration files. Sharing the same set of configuration files with the executable can be useful during software development.

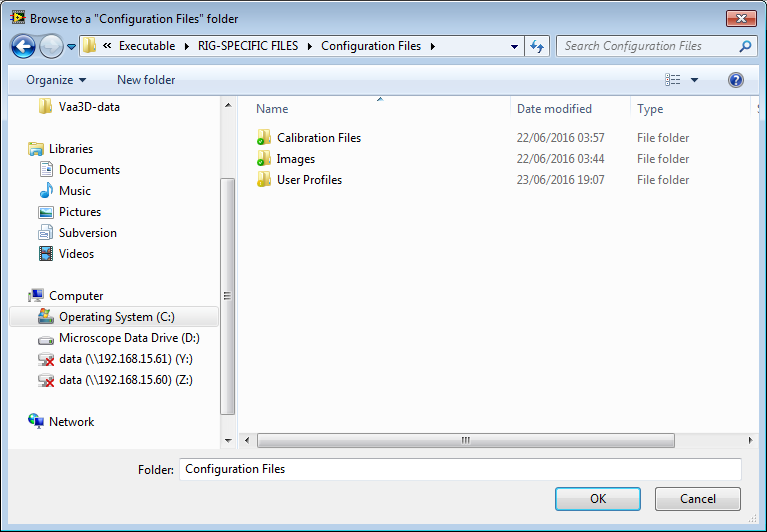
The source code and the executable will behave identically and this also guarantees that user profile changes cannot be accidentally lost when switching between source code and executable.

The “Config pointer.ini” file in C:\Rig 4 Software\Source Code\RIG-SPECIFIC FILES is responsible for holding the path of the config files folder to be used.

This file is automatically generated if it is missing. Deleting the file and running the source code brings up the following message.



Press OK and the file browser window opens to let you choose which set of configuration files you want the source code to use.



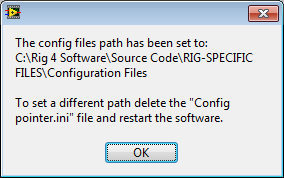
You may browse to the executable’s Configuration Files folder if you want the source code and the executable to share the config files.

Otherwise browse to the source code’s configuration files folder: \Source Code\RIG-SPECIFIC FILES\Configuration Files.

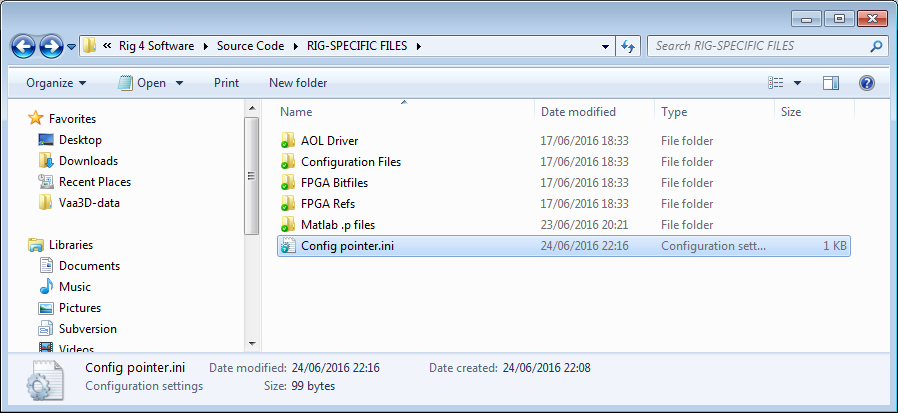
The “Config pointer.ini” file will be generated and it will hold the path you specified in the previous step.

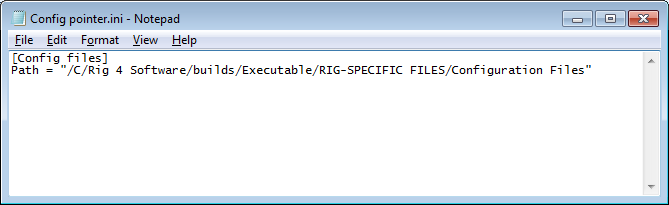
Choosing a folder whose name is not “Configuration Files” is prevented. If you do so the file browser window will persist.

If you press Cancel to close the file browser window the following message will appear and the configuration files path will be automatically set to the source code’s config files folder.



In either case a “Config pointer.ini” file will be generated.





The process described above is a one-step process. The dialogue will be hidden until the “Config pointer.ini” file is deleted manually and the source code restarted.

Note that the executable will always automatically use its own set of configuration files. The above process relates to the source code only, and allows setting the source code to also use the executable’s configuration files as already explained.

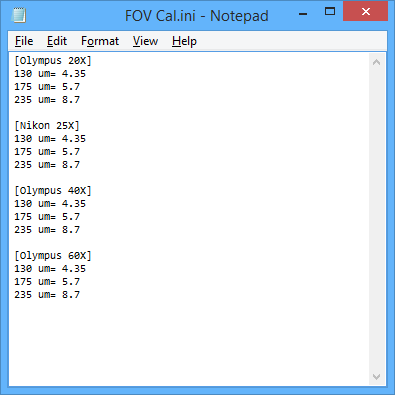
IMPORTANT!

Each time the executable is rebuilt its configuration files are overwritten with those from the source code, except for its custom user profiles. Therefore it is important that the source code’s configuration files are kept up to date during development. For example, if you run a calibration routine using the executable you should manually update the source code’s config files with the latest calibration file.

## Field Of View Setup

Currently the FOV control is set up using the “FOV Cal.ini” file which needs to be created manually. A future upgrade will allow interfacing with this file through the software’s user interface. A new section will be generated in this file each time a new objective is added to the software’s setup.

This file holds the field of view value and acceptance angle for each FOV for each objective.



\RIG-SPECIFIC FILES\Configuration Files\Calibration Files\FOV Cal.ini