



# QEFormStateChange Widget

Andrew Rhyder

Andrew Starritt

21<sup>st</sup> November 2019

Copyright (c) 2019 Australian Synchrotron

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License" within the QE\_QEGuiAndUserInterfaceDesign document.

## Contents

Introduction .....	3
Description .....	3
Image interaction .....	3
Primary image properties .....	4
Other properties .....	5
Window customisation .....	11
Image info area .....	15
Profile plots .....	16
FAQ.....	20
Usage examples .....	21

## Introduction

This document describes in detail the QEImage widget which is an EPICS aware widget provided by the EPICS Qt, aka QE Framework.

This document was created as a separate widget specification document. The main reason for this is ease of maintenance and avoiding editing large and unwieldy word documents.

The QE Framework is distributed under the GNU Lesser General Public License version 3, distributed with the framework in the file LICENSE. It may also be obtained from here:

<http://www.gnu.org/licenses/lgpl-3.0-standalone.html>

## Description

The QEImage widget is used to present an EPICS waveform (typically, but not necessarily from areaDetector) or an MPEG source (also typically from areaDetector) as an image. It provides local analysis tools, such as displaying pixel profiles of slices through the image, and interacts with central analysis tools, such as areaDetector's Region of Interest plugin and analysis plugins.

Images can be zoomed, panned, and scrolled, flipped and rotated. Images views can be captured to a local file, or recorded and played back within the widget. Brightness and contrast may be modified locally if required - that is independently of brightness and contrast related camera controls.

The image has functionality to support sample positioning, region of interest selection, horizontal, vertical, and arbitrary line profiling, including thick line profiling. Profile plots may be presented as part of the QEImage widget, or user selections may be connected back to area detector analysis plugins

## Image interaction

Mark-ups such as those shown in Figure 6 (page 21) are used to indicate coordinates, areas, and sections in the image. These mark-ups can be used to indicate samples, beam position, regions of interest, profiles, etc. Most of these mark-ups can be linked to variables and are interactive; that is, they will be positioned according to the variable values, and when moved by the user will update the variable values. Depending on the mark-ups, they can be moved with the mouse by dragging either the entire mark-up, or individual mark-up handles.

Note, sometimes it is important to position mark-ups accurately. While the mouse button is pressed, the left, right, up and down keys can be pressed and the cursor will move by one pixel. This is useful if you have difficulty positioning the cursor accurately with the mouse.

Markup legends can be changed by the user from the markup context menu. Right click over a markup and select "Set Legend" to alter the legend associated with each markup.

## Primary image properties

The primary inputs for the QEImage widget is the image data itself, image format, and image dimensions. All these must be available for an image to be displayed, but can be sourced in several ways as follows:

### MPEG stream

All image related information can be obtained from an MPEG stream defined using the following property:

(Note, the ability to use an MPEG stream as the source is an option when the QE framework is built. If this property is not available, then the QE framework needs to be rebuilt with this option on)

- **URL** an MPEG stream (typically from the areaDetector MPEG plugin)  
for example: <http://10.135.2.31:8080/avt.cam1.mjpg.mjpg>

### EPICS CA variables

If obtaining the image through EPICS variables, image data is available through the following property:

- **imageVariable** An EPICS waveform record (typically from areaDetector)

If obtaining the image through EPICS variables, the image format properties are as follows:

- **formatVariable** EPICS variable containing the image format. If defined, this value takes priority over the 'formatOption' property.
- **formatOption** The expected pixel format (unrelated to the data type of the waveform record, except that the pixel format must fit in the waveform record data type).  
This property is not used if the 'formatVariable' property is defined.
- **dataTypeVariable** The EPICS variable containing the AreaDetector data type from which bit depth can be inferred. For example, a bit depth of 15 will be used if the EPICS variable value is Int16. If defined, this value takes priority over the 'bitDepth' property. Do not define this property as well as the 'bitDepthVariable' property as there is no guaranteed precedence.
- **bitDepthVariable** The EPICS variable containing the pixel bit depth. If defined, this value takes priority over the 'bitDepth' property. Do not define this property as well as the 'dataTypeVariable' property as there is no guaranteed precedence.
- **bitDepth** The expected bit depth of each pixel. This property is not used if the 'bitDepthVariable' property is defined.

If obtaining the image through EPICS variables, the image dimensions are obtained in one of two ways.

If width and height variables are available, they can be specified in the following properties:

- **widthVariable** An EPICS record (typically from areaDetector)
- **heightVariable** An EPICS record (typically from areaDetector)






If area detector dimensions variables are available they can be specified in the following properties:

- **dimensionsVariable** An EPICS record containing the number of dimensions (typically from areaDetector)
- **dimension1Variable** An EPICS record containing the first dimension (typically from areaDetector). If there are two dimensions this will be used as the image width. If there are three dimensions this will be used as the number of elements per pixel.
- **dimension2Variable** An EPICS record containing the second dimension (typically from areaDetector). If there are two dimensions this will be used as the image height. If there are three dimensions this will be used as the image width.
- **dimension3Variable** An EPICS record containing the third dimension (typically from areaDetector). Only used then there are three dimensions when it is used as the image height.

If the Area Detector dimension variables are static, the 'widthVariable' and 'heightVariable' properties may be set to the appropriate Area Detector dimension variable. The properties 'dimensionsVariable', 'dimension1Variable', 'dimension2Variable' and 'dimension3Variable' must be used when the dimensions are changing. In this case the image width moves between 'dimension1Variable' and 'dimension2Variable', and the image height moves between 'dimension2Variable' and 'dimension3Variable'. The QEImage widget needs the 'dimensionVariable' data to determine which of the dimensions is the width and which is the height.

## Other properties

QEImage user interaction and other properties associated with presenting an image is as follows: (note the full context menu will not be available unless the 'fullContextMenu' property is set.)

- To pause image updating, press  if the Button Bar is displayed, or select 'Pause' from the context menu. To resume image updating, press  if the Button Bar is displayed, or select 'Play' from the context menu.
- To save the current image to a local file, press  if the Button Bar is displayed, or select 'Save...' from the context menu.
- To move the target position into the beam, mark the target and beam positions and press  on the Button Bar. To mark the target and beam, select 'Mark Target' and 'Mark Beam' from the select menu (available on the button bar and in the context menu) and mark the target and beam positions on the image with the mouse. When  is pressed, the EPICS variables representing the target and beam will be updated with pixel coordinates and the target trigger variable will be updated. (Note, the coordinates represent coordinates in the original image and are not affected by how the image is zoomed.) Two EPICS calc records can then be used to perform the required move in each dimension. Each calc record subtracts the target position from the beam position, applies a scaling factor to convert pixels to distance, adds this to the current position of the target and writes the result to the target positioned.

Target and Beam markers can be seen selected in Figure 6.

The EPICS variables written to when marking the beam and target are defined by the following properties:

- targetXVariable
- targetYVariable

- beamXVariable
  - beamYVariable
  - targetTriggerVariable
- To zoom, either:
  - Select the required zoom percentage from the 'Zoom' menu on the button bar or in the context menu.
  - Select 'Fit' from the 'Zoom' menu on the button bar or in the context menu to zoom to a percentage that will fit the image in the current window. The image will be resized if the window size changes.
  - Choose 'Select Area 1' (Region 1) from the Mode menu on the button bar or from the context menu, select an area within the image, then select 'Selected Area' from the 'Zoom' menu on the button bar or in the context menu.

The image may zoomed and set to an initial scroll position by default using the following properties:

- resizeOption
  - zoom
  - initialHosScrollPos
  - initialVertScrollPos
- To rotate an image by 90 degrees clockwise or anticlockwise, or 180 degrees, select the appropriate option from the Flip/Rotate menu. Refer to Figure 9 for an example of rotated images.

The image may be rotated by default using the following property:

- rotation
- To flip image vertically or horizontally, select the appropriate options from the Flip/Rotate menu. Refer to Figure 9 for an example of flipped images.

The image may be flipped by default using the following properties:

- verticalFlip
  - horizontalFlip
- To apply contrast reversal to an image (present a negative view), check 'Contract Reversal' on the 'Image Display Properties' form. (note the 'Image Display Properties' form needs to be enabled using the 'enableImageDisplayProperties' property) or checking 'Image Display Properties' in the options dialog available from the QEImage context menu. Refer to Figure 9 for an example of contrast reversal.

The image contrast may be reversed by default using the following property:

- contrastReversal
- To apply logarithmic weighting to the image brightness scale (which emphasises the difference between lower value pixels), check 'Log Scale' on the 'Image Display Properties' form. (note the 'Image Display Properties' form needs to be enabled using the 'enableImageDisplayProperties' property) or checking 'Image Display Properties' in the options dialog available from the QEImage context menu.

A logarithmic image brightness scale can be applied by using the following property:

- logBrightness
- To apply a false colour representation of the image brightness, check 'False Color' on the 'Image Display Properties' form. (note the 'Image Display Properties' form needs to be enabled using the 'enableImageDisplayProperties' property) or checking 'Image Display Properties' in the options dialog available from the QEImage context menu.

False Colour can be applied by using the following property:

- useFalseColors
- The canvas used to present the fully processed image can be stretched or contracted in either X or Y directions using the 'XStretch' and 'YStretch' properties.

After all image processing is complete, including zoom, flip and rotate, the canvas used to present the image on the screen is normally resized to match the image exactly. This resizing process can be modified to stretch or contract the canvas in either X or Y directions. The fully processed image will then be presented scaled to this modified canvas size.

**NOTE, these properties affect only the presentation of an image AFTER it has been processed and do not take into account user interaction with the image. If an image has been stretched or contracted using these properties user interaction with the image should be avoided as all image interaction currently assumes an un-stretched canvas. For example, selecting a region of interest will ignore the contraction or expansion of the canvas and not return the pixel information expected.**

One use of these properties would be to better present a very narrow image.

While both X and Y can be stretched or contracted, it is recommended that only one dimension is stretched to obtain the aspect ratio required, then the image should be zoomed to get the overall size required.

The current stretch factors can be viewed when the 'displayCursorPixelInfo' property is set or 'Pixel and area information' has been checked in the 'Options' dialog available in the QEImage context menu. Stretch factors are displayed beside the current zoom level.

The canvas used to present an image can be stretched using the following properties:

- XStretch
- YStretch
- To display a timestamp in the top left corner of the image, select 'Show Time' from the context menu.

The timestamp may be shown by default using the following property:

- showTime
- To set the widget in and out of full screen mode, toggle 'Full Screen' in the context menu. Full screen mode may also be selected by default using the following property:
  - fullScreen
- To present a profile of pixel values on a vertical 'Horizontal Slice Profile', 'Vertical Slice Profile', or 'Line Profile' from the Mode menu and mark a vertical slice, a horizontal slice, or mark an arbitrary line on the image with the mouse. After the markup is drawn, the mouse can be used to drag the markup to a new location or, in the case of the arbitrary line, can also be used to drag either end of the line to a new location. The mark-ups can be cleared by right clicking over the outline and selecting 'Clear'

Figure 6 shows an image with Vertical, Horizontal and arbitrary profiles selected.

The profile thickness can be changed from a single line by grabbing the square handle in the middle of the line and moving the line boundary as required. The line boundary lines (dashed) can be grabbed anywhere and dragged to change the line thickness. When dragged back to the centre line the dashed boundary lines disappear, the thickness reverts to a single pixel, and the square handle used to set the thickness reappears in the centre of the line. The line thickness can also be returned to single line thickness from the line's context menu.

The profile plots are simple indicative plots of the profile data. For more detailed analysis, the profile data presented in the plot can be copied by selecting 'Copy Plot Data' from the plot context menu. This can then be pasted into another program such as Excel. Note the data displayed and copied is generated from the most recent image update using the full original image data. It is unaffected by the current zoom level. The current zoom level will affect how accurately the lines can be positioned.

- To set the area in up to 4 areaDetector Region of Interest plugins, select 'Select Area 1', 'Select Area 2', 'Select Area 3' or 'Select Area 4' from the Mode menu on the button bar or in the context menu, and mark the area in the image using the mouse. When marked, the four EPICS areaDetector variables representing the Region of Interest area position and size will be updated. Figure 8 shows an example of this.

After the area mark-ups are drawn, the mouse can be used to drag the markups to a new location to drag individual sides or corners to a new location. The area can be cleared by right clicking over the outline and selecting 'Clear'

The four EPICS areaDetector variables for each area are defined by the following properties:

- regionOfInterest1XVariable
- regionOfInterest1YVariable
- regionOfInterest1WVariable (width)
- regionOfInterest1HVariable (height)
- regionOfInterest2XVariable
- regionOfInterest2YVariable
- regionOfInterest2WVariable (width)
- regionOfInterest2HVariable (height)
- regionOfInterest3XVariable
- regionOfInterest3YVariable
- regionOfInterest3WVariable (width)
- regionOfInterest3HVariable (height)
- regionOfInterest4XVariable
- regionOfInterest4YVariable
- regionOfInterest4WVariable (width)
- regionOfInterest4HVariable (height)
- To highlight an area of an image, for example an area detector defined centroid, a non-interactive ellipse can be drawn over an image at coordinates defined by variables with the following properties:
  - ellipseXVariable



- ellipseYVariable
  - ellipseWVariable (width)
  - ellipseHVariable (height)
  - ellipseRotationVariable (optional – specifies clockwise rotation in degrees. When not specified the rotation is 0 degrees)
  - ellipseVariableDefinition (specifies if X and Y are the ellipse centre, or top left of a bounding rectangle)
- Image clipping can be achieved by defining clipping variables with the following properties:
  - clippingLowVariable
  - clippingHighVariable
  - clippingOnOffVariable
- To simplify the user interfaces, some options can be disabled by default using the following properties:
  - enableVertSlice1Selection to enableVertSlice5Selection
  - enableHozSlice1Selection to enableHozSlice5Selection
  - enableProfileSliceSelection
  - enableAreaSliceSelection (for all area and region selection)
  - enableTargetSliceSelection (for beam and target selection)
  - enableArea1Selection
  - enableArea2Selection
  - enableArea3Selection
  - enableArea4Selection
- Some image mark-ups and other options can be displayed when the image is first presented using the following properties: (Note, mark-ups are generally displayed when first drawn by the user, or can be displayed from the QEImage context menu 'Markup Display' sub-menu.
  - displayVertSlice1Selection to displayVertSlice5Selection
  - displayHozSlice1Selection to displayHozSlice5Selection
  - displayProfileSelection
  - displayArea1Selection
  - displayArea2Selection
  - displayArea3Selection
  - displayArea4Selection
  - displayTargetSelection
  - displayBeamSelection
  - displayEllipses
  - displayCursorPixelInfo ('cursor pixel info' is not a markup. It is an area under the image displaying information about the image, in particular the pixel currently under the cursor)
- Markup colors can be altered using the following properties:
  - vertSlice1Color to vertSlice5Color
  - hozSlice1Color to hozSlice5Color
  - profileColor

- areaColor
  - beamColor
  - targetColor
  - timeColor
  - ellipseColor
- Markup legends can be altered using the following properties:
  - hozSlice1Legend to hozSlice5Legend
  - vertSlice1Legend to vertSlice5Legend
  - profileLegend
  - areaSelection1Legend
  - areaSelection2Legend
  - areaSelection3Legend
  - areaSelection4Legend
  - targetLegend
  - beamLegend
  - ellipseLegend
- Displays of information about the image such as line profile plots, and controls such as the 'Image Display Properties' control, are presented within the image by default. The following property can be set to request the application displaying the QEImage widget host these displays in docks. The QEGui application will honour these requests. Note, they may initially be created hidden. If the menu system has not been customised to add menu items to hide and unhide these controls, you may need to enable them from the menu bar context menu (right click on the menu bar).
  - externalControls
- The context menu may be a full context menu containing options to manipulate the image or a simpler context menu containing just the standard context menu options. The following property determines which context menu is available.
  - fullContextMenu
- Controls to manipulate local image display properties controls can be enabled by setting the 'enableImageDisplayProperties' property, or by checking 'Image Display Properties' in the option dialog available from the widget's context menu. Display properties such as brightness and contrast can then be set as required. Note, local brightness and contrast are independent of areaDetector brightness and contrast settings. If 'auto brightness and contrast' is checked then selecting any area or region of interest will cause the brightness and contrast to be adjusted so match the range of pixel in the selected area. The reset button above the brightness and contrast sliders can be pressed to reset the controls to 'normal'.
- Controls to record and playback images can be enabled by setting the 'enableRecording' property, or by checking 'Recording' in the option dialog available from the widget's context menu.
- If the image is not being displayed correctly, the QEImage context menu option 'About Image...' can be used to display a message box documenting how the QEImage widget is interpreting the image data.

- Markups can be displayed as soon as variable data is available for them by setting the 'displayMarkups' property. This is intended for uses when an image is displayed for a specific purpose such as target positioning or region of interest area selection. In these cases a GUI is presented with the mark-ups already shown ready for manipulation. In a more general GUI, it may not be appropriate to display mark-ups until the user selects a markup mode and interacts with the widget. If 'displayMarkups' is selected, only mark-ups for which there is available data are presented. For example, if a GUI is designed for target positioning and of all the markup related variables only variables for beam position and target position are defined, then if 'displayMarkups' is set only the target and beam mark-ups will be shown when the widget is first displayed. Note that this property will over-ride other markup properties, such as displayProfileSelection. E.g. if displayMarkups is true but displayProfileSelection is false, you might expect that the profile markup would not be shown, but in fact if the variable data associated with the profile changed, that markup would in fact become visible. So, if you want to be able to completely control the visibility of the markup, regardless of changes in its underlying data, you should set displayMarkups to be false, and then control visibility via the displayProfileSelection, displayArea1Selection or one of the other display<Something>Selection properties mentioned above.
- The image presented in the QEImage widget can be used in other application by using the standard 'Copy' function in the context sensitive menu, or by saving the image to a file using the 'Save...' button or by selecting 'Save...' from the context sensitive menu. Copy is also available as a request from the window customisation mechanism. An integrated sequence of feeding a saved image to another application is also available. When the window customisation mechanism requests 'LaunchApplication1' or LaunchApplication2' from a QEImage widget, the widget
  - Saves the current image to a temporary file
  - Launches the application specified in property 'program1' or 'program2' with the temporary filename appended to, or included within, the arguments specified in property 'arguments1' or 'argument2'. If any arguments specified in the property 'arguments1' or 'argument2' include the keyword <FILENAME>, this keyword is replaced by the temporary filename. If this keyword is not found, the temporary filename is added as the last parameter.
  - Deletes the temporary file when the launched application exits.

### Window customisation

The QEImage widget can act on requests from the window customisation mechanism. Refer to the QE\_QEGuiAndUserInterfaceDesign document for details on the window customisation mechanism.

The following request names are valid for a QEImage widget:

- Save...
- Pause
- Move target position into beam
- About image...

- Zoom
- Flip/Rotate
- Mode
- Markup Display
- Options...
- Copy
- LaunchApplication1
- LaunchApplication2
- Place the image in 'full screen' mode

The following QEImage controls are available for hosting in an application such as QEGui:

- Image Display Properties
- Recorder
- Arbitrary Profile
- Horizontal Slice Profile
- Vertical Slice Profile

Examples of elements in a window customisation menu definition file are:

(Note, these examples refer to a widget named 'BeamImage'.)

- Button to save the image presented in a QEImage named BeamImage:

```
<Item Name="Save...">
  <BuiltIn Name="Save...">
    <WidgetName>BeamImage</WidgetName>
  </BuiltIn>
</Item>
```

- Button to allow the QEImage named BeamImage to be paused:

```
<Item Name="Pause">
  <BuiltIn Name="Pause">
    <WidgetName>BeamImage</WidgetName>
  </BuiltIn>
</Item>
```

- Menu to zoom the QEImage:

```
<Item Name="Zoom">
  <BuiltIn Name="Zoom">
    <WidgetName>BeamImage</WidgetName>
  </BuiltIn>
</Item>
```

- Menu to flip and rotate the QEImage:

```
<Item Name="Flip/Rotate">
  <BuiltIn Name="Flip/Rotate">
    <WidgetName>BeamImage</WidgetName>
  </BuiltIn>
</Item>
```

- Menu to chose the current user interaction with the image: (for example, when clicking and dragging over the image, is the user panning the image, or selecting a region of interest, etc)

```
<Item Name="Mode">
  <BuiltIn Name="Mode">
    <WidgetName>BeamImage</WidgetName>
  </BuiltIn>
</Item>
```

- Menu to chose the hide or reveal the enabled mark-ups.

```
<Item Name="Markup Display">
  <BuiltIn Name="Markup Display">
    <WidgetName>BeamImage</WidgetName>
  </BuiltIn>
</Item>
```

- Button to present the Options dialog for the QEImage named BeamImage:

```
<Item Name="Options...">
  <BuiltIn Name="Options...">
    <WidgetName>BeamImage</WidgetName>
  </BuiltIn>
</Item>
```

- Button to launch an application using the current image: (current image is saved to a temporary file and passed to the application as an argument)

```
<Item Name="Open image in 'Gimp'">
  <BuiltIn Name="LaunchApplication1">
    <WidgetName>BeamImage</WidgetName>
  </BuiltIn>
</Item>
```

- Button to hide or view the Image Display Properties control: (This assumes the 'enableImageDisplayProperties' property is set, and the 'externalControls' property is set.) (Note, the item name is redundant as the Dock itself supplied the item loaded into the menu bar)

```
<Item Name="Image Display Properties">
  <Dock>
    <Title>Image Display Properties</Title>
```

```
</Dock>
</Item>
```

Note, if the 'name' property has been defined to differentiate between docks from multiple QEImage widgets, then the title element will need to include the name value as follows:

```
<Title>name - Image Display Properties</Title>
```

- Button to hide or view the Image Recording control: (This assumes the 'enableRecording' property is set, and the 'externalControls' property is set.)  
(Note, the item name is redundant as the Dock itself supplied the item loaded into the menu bar)

```
<Item Name="Recorder">
  <Dock>
    <Title>Recorder</Title>
  </Dock>
</Item>
```

Note, if the 'name' property has been defined to differentiate between docks from multiple QEImage widgets, then the title element will need to include the name value as follows:

```
<Title>name - Recorder</Title>
```

- Button to hide or view the Arbitrary Line profile plot: (This assumes the 'enableProfilePresentation' property is set, and the 'externalControls' property is set.)  
(Note, the item name is redundant as the Dock itself supplied the item loaded into the menu bar)

```
<Item Name="Arbitrary Line Profile Plot">
  <Dock>
    <Title>Arbitrary Profile</Title>
  </Dock>
</Item>
```

Note, if the 'name' property has been defined to differentiate between docks from multiple QEImage widgets, then the title element will need to include the name value as follows:

```
<Title>name - Arbitrary Profile</Title>
```

- Button to hide or view the Horizontal Slice profile plot: (This assumes the 'enableHozSlicePresentation' property is set, and the 'externalControls' property is set.)  
(Note, the item name is redundant as the Dock itself supplied the item loaded into the menu bar)

```
<Item Name="Horizontal Profile Plot">
  <Dock>
    <Title>Horizontal Slice Profile</Title>
  </Dock>
</Item>
```

Note, if the 'name' property has been defined to differentiate between docks from multiple QEImage widgets, then the title element will need to include the name value as follows:

```
<Title>name - Horizontal Slice Profile</Title>
```

- Button to hide or view the Vertical Slice profile plot: (This assumes the enableVertSlice1Presentation' property is set, and the 'externalControls' property is set.) (Note, the item name is redundant as the Dock itself supplied the item loaded into the menu bar)

```
<Item Name="Vertical Profile Plot">
  <Dock>
    <Title>Vertical Slice Profile</Title>
  </Dock>
</Item>
```

Note, if the 'name' property has been defined to differentiate between docks from multiple QEImage widgets, then the title element will need to include the name value as follows:

```
<Title>name - Vertical Slice Profile</Title>
```

- Button to place the image in 'full screen' mode:

```
<Item Name="Full Screen">
  <BuiltIn Name="Full Screen">
    <WidgetName>BeamImage</WidgetName>
  </BuiltIn>
</Item>
```

### Image info area

An information area can be displayed below the image if the 'displayCursorPixelInfo' property is set. This can be brief, showing pixel information under the current pointer position, image update status, and current zoom level, or more extensive giving coordinates of profile and area selections, and of targeting points. The 'briefInfoArea' property is set to give a brief info area or cleared to give an extensive info area.

The image update status consists of the word 'Live' or 'Paused'. Even when 'Live' the image will only update as frames are delivered from the imaging device. When viewing a static scene it may not be clear if fresh frames are being displayed. An animated graphic beside the word 'Live' gives an indication when new frames arrive. The animation moves only when a new frame is delivered.

For the extensive info area, the following keys are used to identify the additional items:

- V: Vertical slice selection (position and thickness)
- H: Horizontal slice selection (position and thickness)
- L: Arbitrary line profile selection (position and thickness)
- R1, R2, R3, R4: Region selections
- T: Target selection

- B: Beam selection

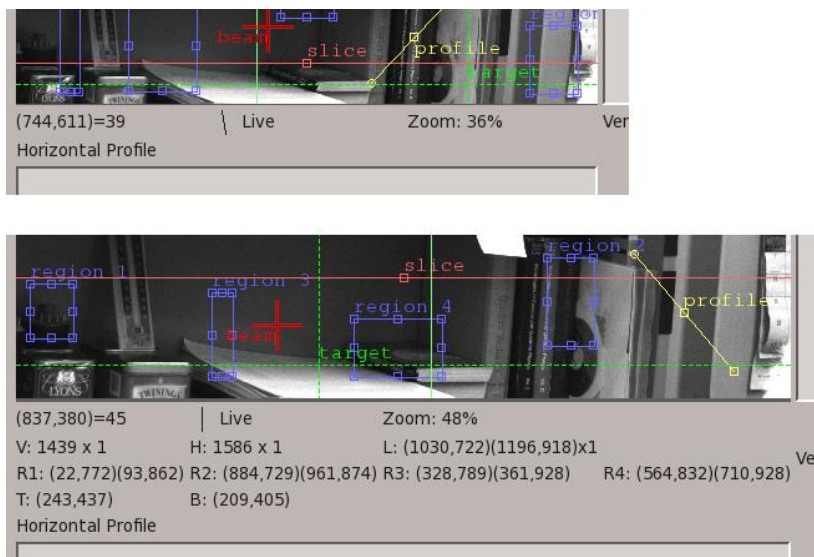


Figure 1 Image info area

## Profile plots

A user can select horizontal or vertical lines through the entire image, or an arbitrary line anywhere within the image and display a plot of the pixel data on the line. The user can select a line thickness to an average of include multiple pixels in the profile data.

To select a vertical or horizontal line through the entire image, select 'Mode -> Horizontal Slice 1' or 'Mode -> Vertical Slice 1' from the context menu (right click over the image). Then click on any point in the image. A line will appear across the screen. (Note, there are five vertical and five horizontal mark-ups available. The vertical and horizontal profile plots are always associated with the first of each of these sets.)

To select an arbitrary line through any part of the image, select 'Mode -> Line Profile' from the context menu (right click over the image). Then drawn a line anywhere in the image.

To move the line press the mouse button on the line anywhere except the small marker in the centre of all the lines, or the small markers at the end of the arbitrary profile line, then drag the line.

To specify a line thickness greater than one, click and drag the marker in the centre of the lines. Dotted lines will appear either side of the main line indicating the thickness selected. These dotted lines can then be dragged to change the line thickness, including setting it back to 1. It may be easier to use the context sensitive menu to reset the line thickness to 1, or to select a specific pixel width.

Note, while the mouse button is pressed, the left, right, up and down keys can be pressed and the cursor will move by one pixel. This is useful if you have difficulty positioning the cursor accurately with the mouse.

Refer to Figure 2 to see how the profile selection indicators can be manipulated.



A context sensitive menu is also provided for the profile lines. This menu provides the following:

- Hide  
Remove the profile marker from the image and hide the profile plot if present.
- Single Pixel Line Thickness  
Set the line thickness to one.
- Set Line Thickness  
set the line thickness to a specific value.

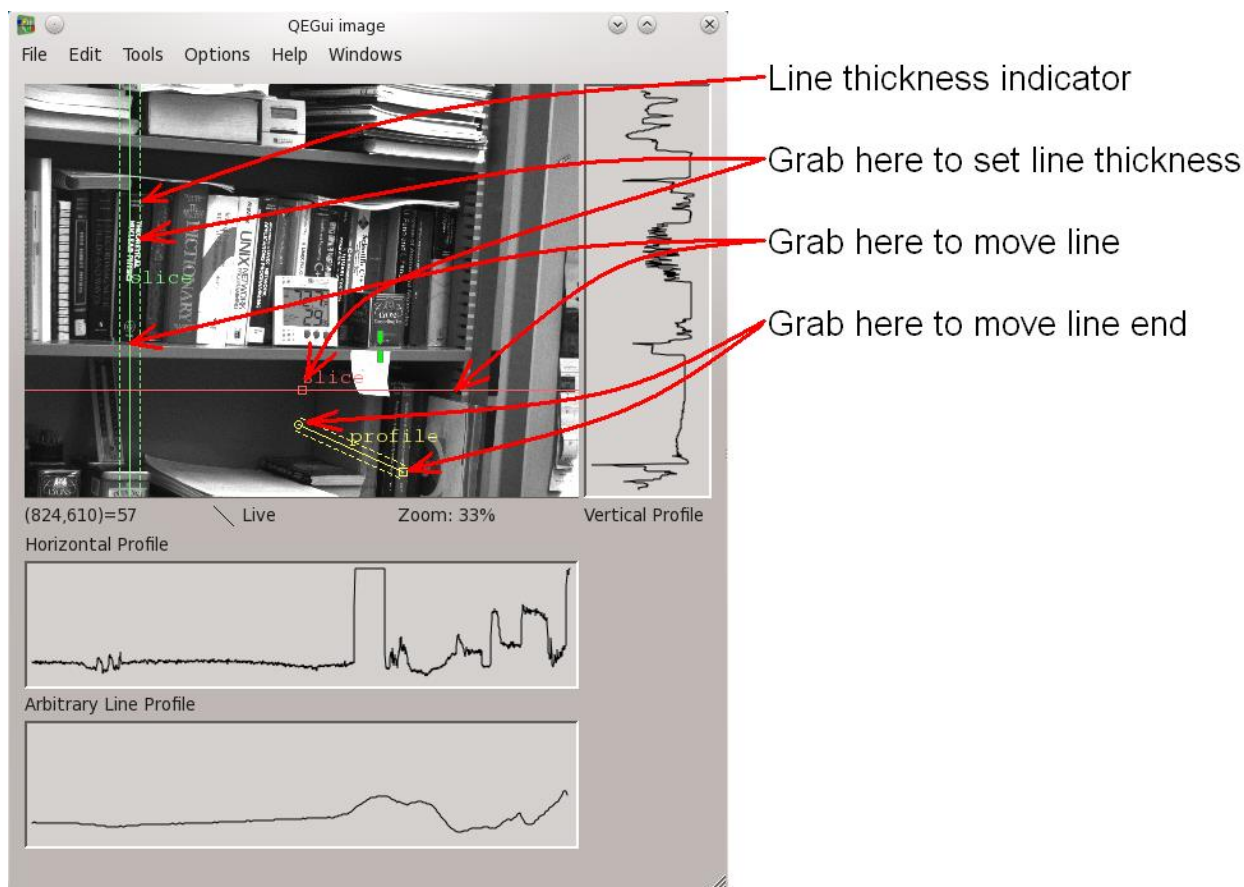


Figure 2 Profile selection and manipulation

To select and use any profile, the 'enableVertSlice1Selection', 'enableHozSlice1Selection', and 'enableProfileSlice1Selection' properties must be set.

When a profile is selected, the profile data may be presented in many ways:

- **Profile data presented within the QEImage widget.**  
When the profile marker is displayed in the image, an associate plot of the profile data is added within the QEImage widget beside the image.  
To present a profile plot within the QEImage widget the 'enableVertSlicePresentation', 'enableHozSlicePresentation', and 'enableProfileSlicePresentation' properties must be set.  
Figure 3 show an example of this configuration.

- Profile data presented by the QEImage widget, but displayed elsewhere in the application.**  
 QE widgets have the ability to create controls or other widgets and request the host them. The QEImage can be configured to request the application host its profile plots. If the QEImage widget is being presented with the QEGui application, QEGui will add the profile plots as docks. To present a profile plot elsewhere within the application the 'externalControls' property must be set. Also, the 'enableVertSlicePresentation', 'enableHozSlicePresentation', and 'enableProfileSlicePresentation' properties must be set. Figure 4 show an example of this configuration.
- Profile data is written to a variable and displayed in another widget monitoring this variable, or used in some other way.**  
 To only use the profile data elsewhere via a variable the 'enableVertSlicePresentation', 'enableHozSlicePresentation', and 'enableProfileSlicePresentation' properties should be cleared. Profile data is written to the variables given in the 'profileHoz1ArrayVariable', 'profileVert1ArrayVariable', and 'lineProfileArrayVariable' properties. Figure 5 show an example of this configuration.
- Coordinates from the profile selection are written to variables and used for external analysis. For example, area detector centroid calculations. Profile data is not generated by the QEWidget.**  
 To only use the profile coordinates via variables the 'enableVertSlicePresentation', 'enableHozSlicePresentation', and 'enableProfileSlicePresentation' properties should be be cleared. Profile coordinates are written to the variables given in the 'profileHozVariable', 'profileHozThicknessVariable', 'profileVertVariable', 'profileVertThicknessVariable', 'lineProfileX1Variable', 'lineProfileY1Variable', 'lineProfileX2Variable', 'lineProfileY2Variable', and 'lineProfileThicknessVariable' properties.

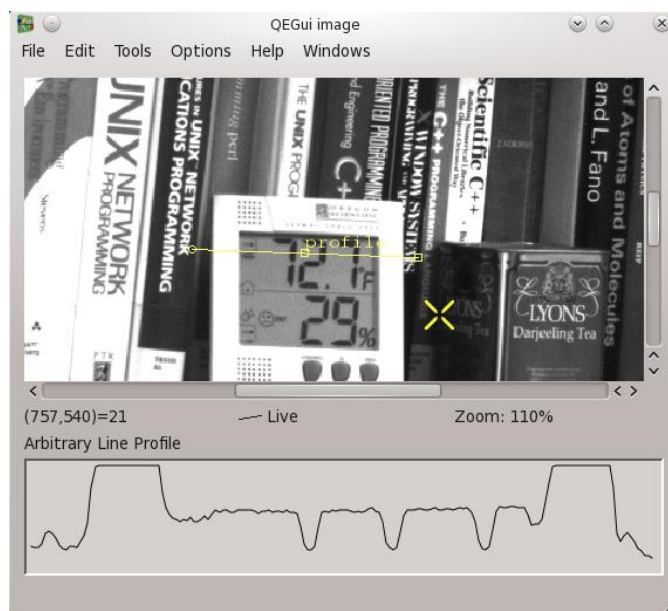


Figure 3 QEImage presenting an arbitrary line profile plot within the widget

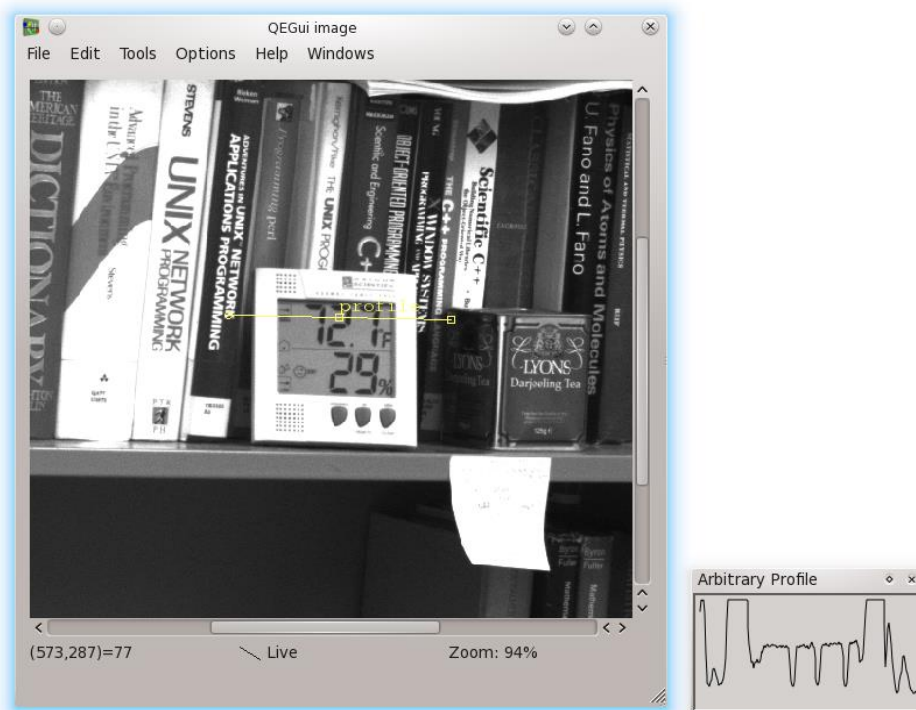


Figure 4 QEGui application presenting an arbitrary line profile plot managed by a QEImage widget

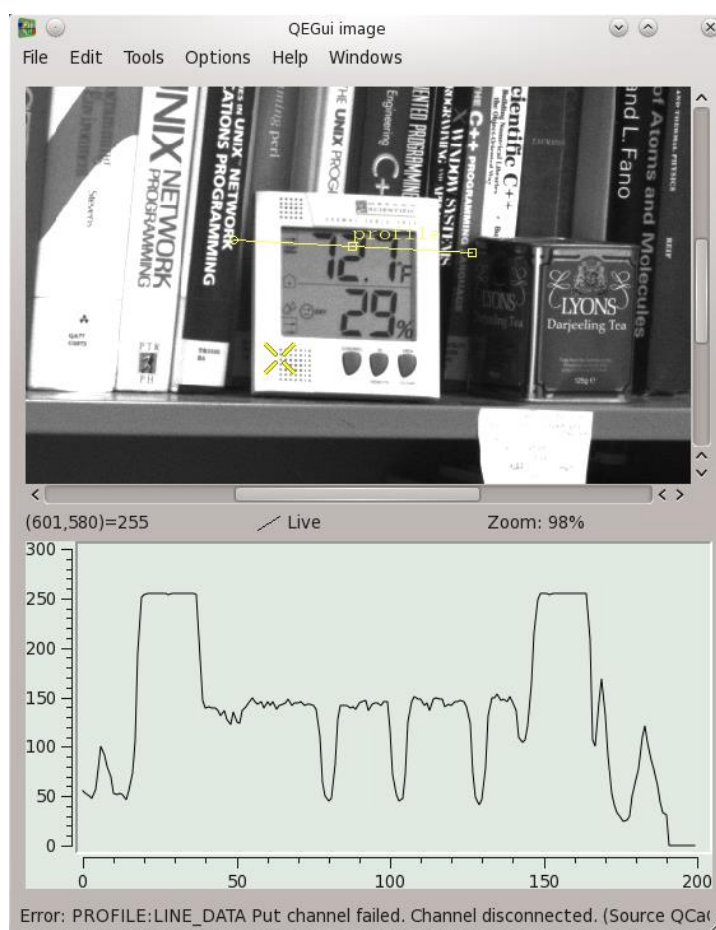


Figure 5 QEPlot widget used to display arbitrary line profile plot data written to a variable by a QEImage widget

## FAQ

- **I tried turning on the Image Display Properties and the Recording controls but they didn't appear.**

If the 'externalControls' property is set the 'Image Display Properties' and the 'Recording' controls will not appear within the QEImage widget. Instead, the controls are available for the application to locate elsewhere. In the case of the QEGui application, the controls will be available as a docked widget.

- **I have an arbitrary line (or profile) markup connected to a set of PVs that determine its angle and position in the image. I tried hiding the markup by setting displayProfileSelection to false but it keeps reappearing.**

If the 'displayProfileSelection' property is set to false it will hide the profile selection line (or horizontal selection line or other similar markup) but if displayMarkups is true, that markup will be redrawn next time its data changes. If you want to hide the profile selection line (or any other similar markup), then you need to set 'displayMarkups' to false as well as setting the appropriate 'display<Something>Selection' property to false.

- **I selected a profile through the image (horizontal, vertical, or arbitrary) but the profile plot doesn't appear.**

QEImage includes profile plots as shown in Figure 6. These profile plots appear when the

relevant selection is made in the image. If the 'externalControls' property is set, however, they will not appear within the QEImage widget. Instead, the control is available for the application to locate elsewhere. In the case of the QEGui application, the control will be available as a docked widget. Also, if the 'enableHozSlicePresentation', 'enableVertSlicePresentation', or 'enableProfilePresentation' are not enabled, the plots will never be generated. This is appropriate when the profile data is being written to a variable and used elsewhere. See 'Profile plots' (page 16) for details.

- **How do I position interactive items such as region of interest areas accurately?**

The left, right, up and down keys can be pressed instead of moving the mouse. The cursor will move one pixel each key press. Note, the mouse is remains active and will also move the cursor if moved. Also, key presses will only move the cursor when mouse the button is pressed.

## Usage examples

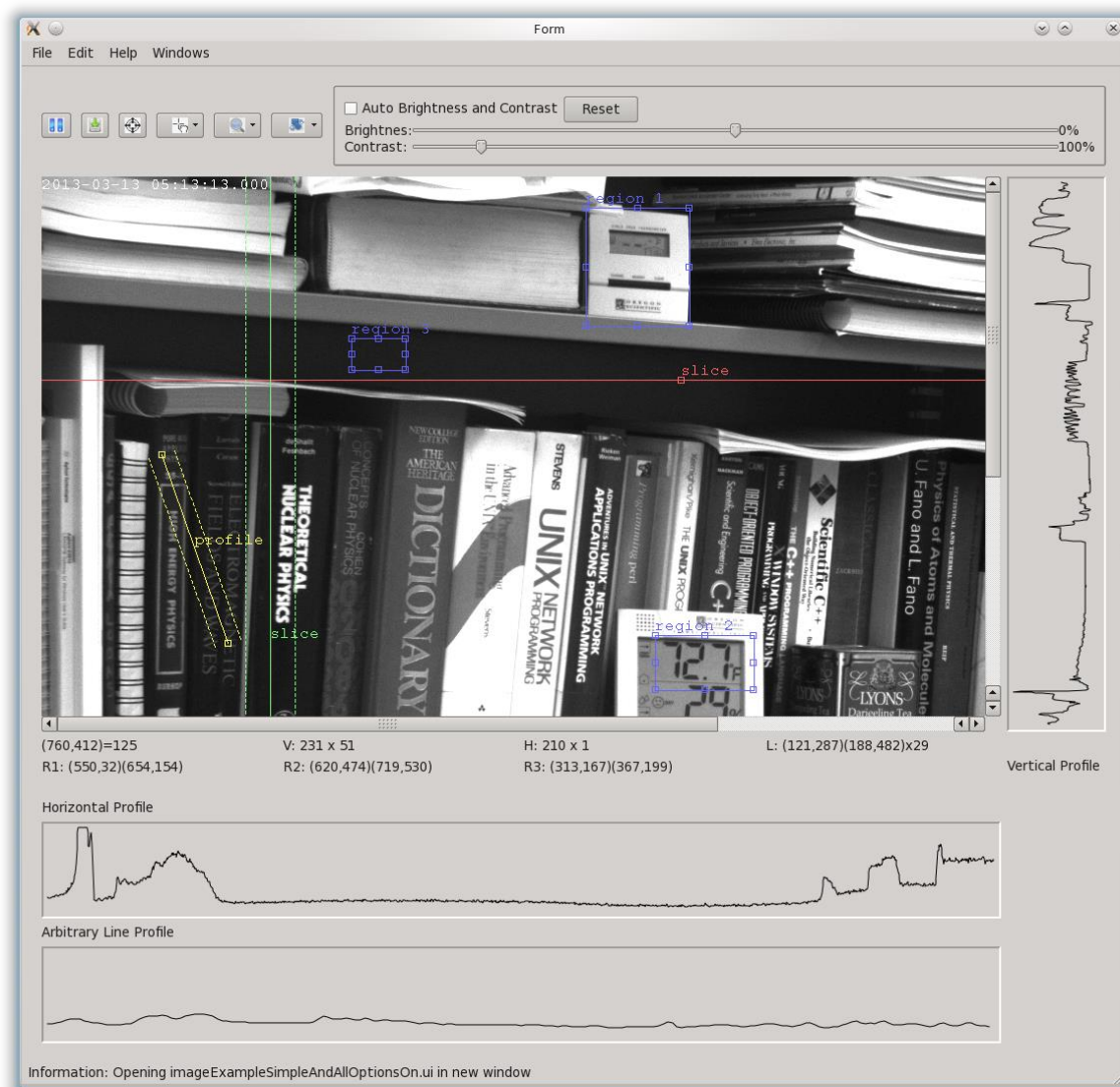


Figure 6 QEImage with most options activated



Figure 7 Minimal use of QEImage



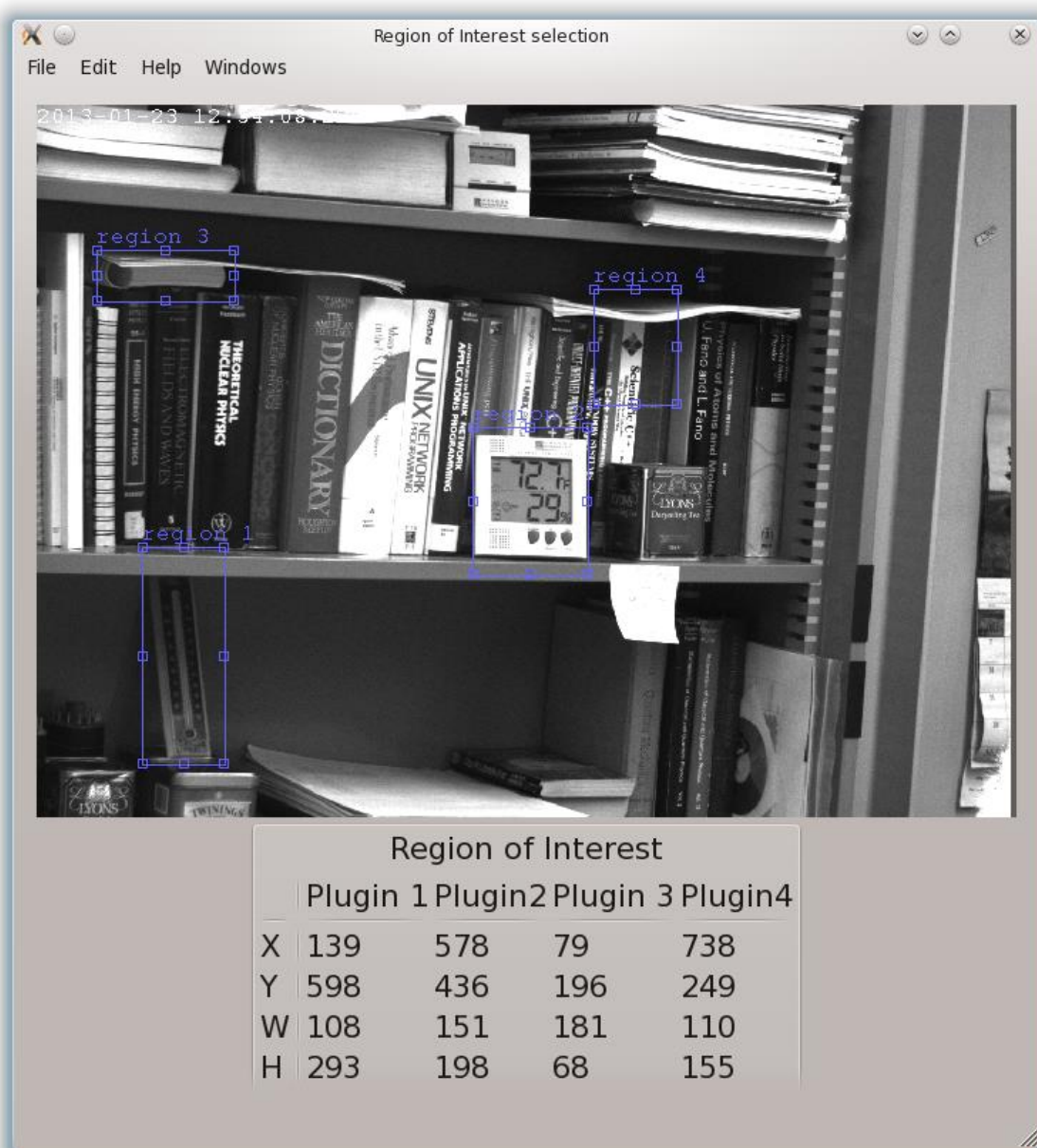


Figure 8 QEImage specifying areaDetector Region of Interest

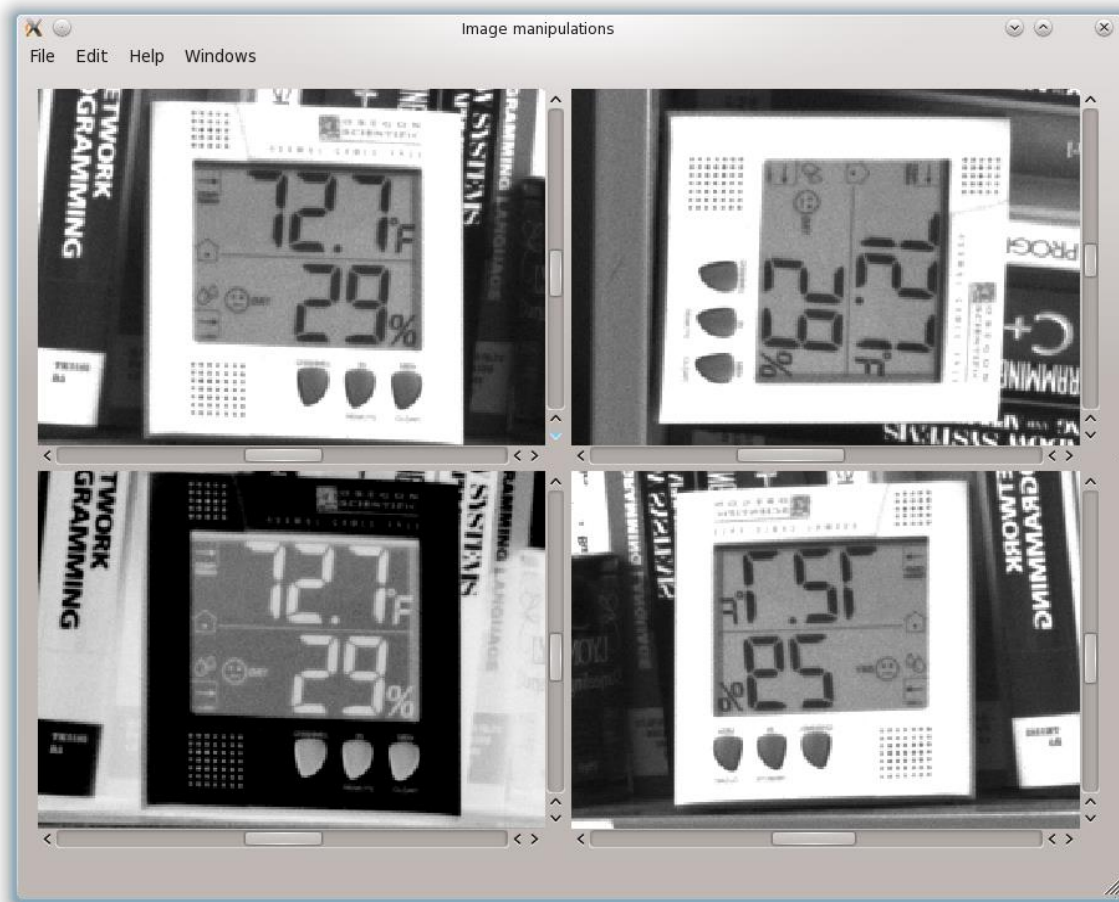


Figure 9 Some QEImage image manipulation options



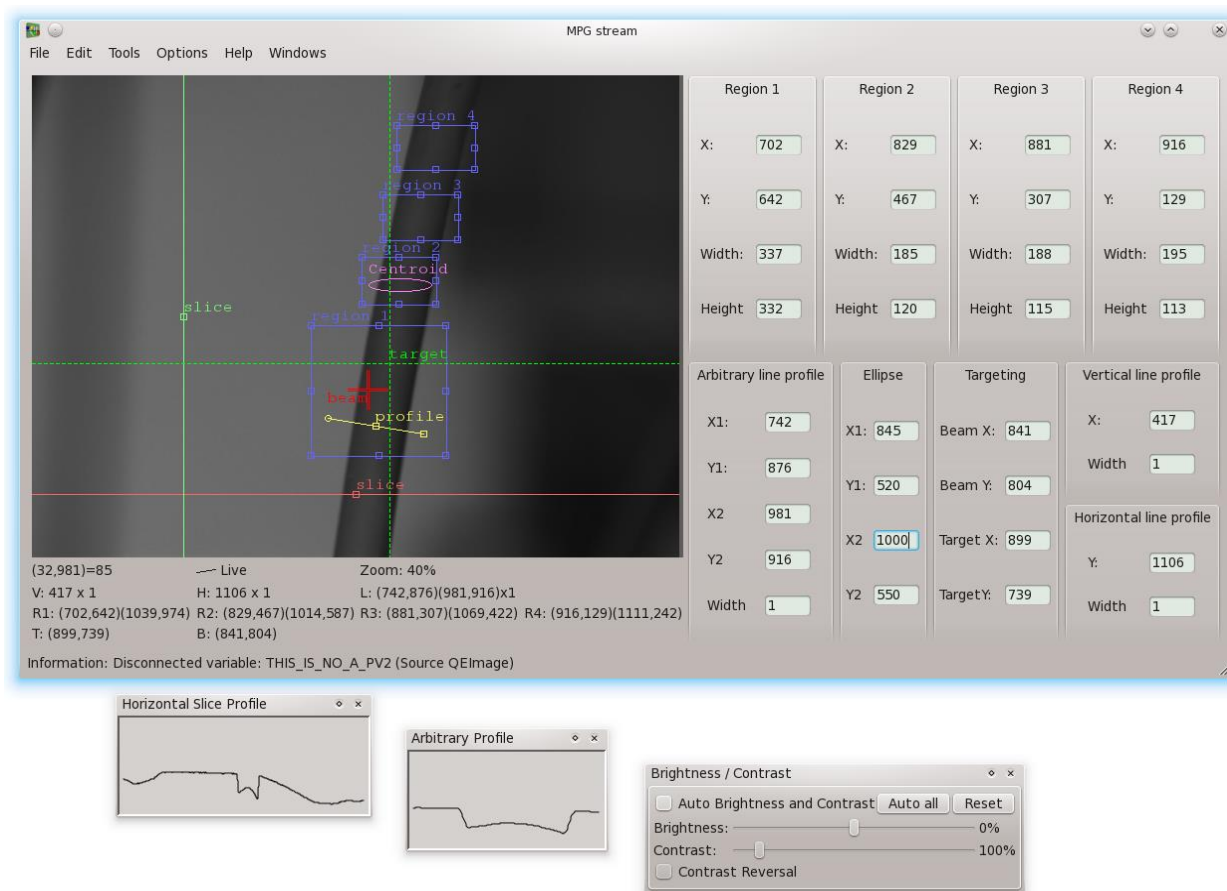


Figure 10 Image with associated docks