Vidview Settings Reference Manual

Table of Contents

1 <root>

1.1 Properties

error *read, notify*
The most recent error message.

version *read*
The product version number.

1.2 Methods

close
Shut down the current application instance.

help
Generate and open help document in browser.

2 License

2.1 Properties

license.expires *read*
License expiration date.

license.key *read*
License key.

license.signed *read*
License signing date.

license.status *read, write, notify*
License status [missing | invalid | wronghardware | wrongdate | expired | valid]

license.valid *read, notify*
Whether the license is valid [true | false].

2.2 Methods

license.address <license key>
<license key> Your license key.
Address to the license on the license server. Use it to download a license file, which can then be loaded using the load method.
Example: `license.address 43j3khj8`

license.load <locator>
<locator> Locator of license file.
Load license from a file.
Example: `license.load file:///c:/license.xml`

license.request <license key>
<license key> Your license key.
Request and load a license from the license server.
Example: `license.request 43j3khj8`

3 Media

3.1 Properties

media.crop *read, write, notify*
The media frame cropping [left, right, top, bottom | horizontal, vertical | all].
Example: `media.crop 20, 40`

media.devices *read*
A list of all capture devices that can be opened.

media.end *read, notify*
Media end position in format [[h:]mm:]ss[.fff].

media.endmode *read, write, notify*
Media end mode [eject | pause | play | repeat].

media.extensions *read*
A list of all media file extensions that can be opened.

media.hasnext *read, notify*
Whether the input media position can seeked to the last captured position [true | false].

media.hasprevious *read, notify*
Whether the input media position can seeked to the first captured position [true | false].

media.offset *read, write, notify*
Media offset position in format [[h:]mm:]ss[.fff].

media.position *read, notify*
Current media position in format [[h:]mm:]ss[.fff].

media.ratio *read, write, notify*
The media aspect ratio, as a double value [integer.decimals] or integer fraction [nominator/denominator].
Example: `media.ratio 16/9`

media.resource *read, notify*
Locator of the currently opened resource.

media.scan *read, write, notify*
Media scan format [unknown | interlaced | progressive].

media.seekable *read, notify*
Whether the input media is seekable [true | false].

media.start *read, write, notify*
Media start position in format [[h:]mm:]ss[.fff].

media.state *read, notify*
Media state [closed | paused | playing].

3.2 Methods

media.eject
Eject currently opened media.

media.next
Play the media from the last captured position.

media.open <locator>
<locator> Locator of file, capture device or video stream.
Open media specified by locator argument.
Example: `media.open file:///c:/test.avi`

media.pause
Pause playback, if playing.

media.play
Start playback of opened media.

media.previous
Play the media from the first captured position.

media.seek <position>
Seek media to specified position in format [[h:]mm:]ss[.fff].
Example: `media.seek 03:12`

4 Settings

4.1 Properties

settings.asynchronous *read, write*
Asynchronous mode [None | Set | Notify | SetNotify | Call | SetCall | NotifyCall | All], where Set, Notify, and Call make property sets, notifications, and method calls, respectively, asynchronous. This means that the application will not wait for a response from the callee.

5 Snapshot

5.1 Properties

snapshot.sources *read*
Comma separated list of snapshot sources currently available for capturing.

5.2 Methods

snapshot.capture <source> <locator>
<source> Snapshot source [original | enhanced | viewer].
<locator> Where to store the snapshot.
Capture the last frame in the video stream.
Example: `snapshot.capture original file:///$(Pictures)/snapshot-$(Time).png`

6 Vidhance.contrast

6.1 Properties

vidhance.contrast.complexity *read, write, notify*
Computational complexity of contrast algorithm. Valid values are integers in the range [0 - 10], where 0 turns contrast enhancements off.
Default: `3`

vidhance.contrast.intensity *read, write, notify*
Contrast intensity value in the range [0 - 1] where 0 equals no change and 1 equals extreme changes.
Default: `0.5`

vidhance.contrast.mode *read, write, notify*
Contrast optimization mode [off | luma | color | colorize | monochrome].
Default: `luma`

7 Vidhance.deinterlace

7.1 Properties

vidhance.deinterlace.method *read, write, notify*
Deinterlace method [even | odd].
Default: `even`

vidhance.deinterlace.mode *read, write, notify*
Deinterlace mode [off | on].
Default: `off`

8 Vidhance.mosaic

8.1 Properties

vidhance.mosaic.fade *read, write, notify*
Local mosaic fading speed [0 - 1].
Default: `0.5`

vidhance.mosaic.mode *read, write, notify*
Local mosaic mode [off | on].
Default: `on`

vidhance.mosaic.size *read, write, notify*
Local mosaic size (width, height).
Default: `1024,1024`

9 Vidhance.sharp

9.1 Properties

vidhance.sharp.interval *read, write, notify*
The amount of time allowed for sharp image selection. Will clamp to values between 0 and 10 seconds.
Default: `00:00:00.400`

vidhance.sharp.mode *read, write, notify*
Sharp mode [off | stills | pause | filter].
Default: `stills, pause`

10 Vidhance.stabilize

10.1 Properties

vidhance.stabilize.mode *read, write, notify*
Stabilization mode [off | overview | zoom | operator | auto].
Default: `overview`

vidhance.stabilize.scale *read, write, notify*
The scale of the stabilization area target. If set to 0 it defaults to the viewer's scale, or 1 when no viewer is present.

vidhance.stabilize.size *read, write, notify*
The size of the stabilization area target. If set to (0, 0) it defaults to the viewer's size, or (1024, 768) when no viewer is present.

10.2 Methods

vidhance.stabilize.reset
Reset stabilization.

vidhance.stabilize.resetpan
Reset stabilization panning.

vidhance.stabilize.resetrotation
Reset stabilization rotation.

vidhance.stabilize.resetscale
Reset stabilization scale.

11 Vidhance.track

11.1 Properties

vidhance.track.active *read, write, notify*
Whether tracking is currently active [false | true].

vidhance.track.bounds *read, notify*
The tracked object region bounds (left, top, width, height) in a normalized coordinate system, where (-0.5,-0.5) is the upper left corner and (0,0) is the image center.

vidhance.track.defaultsize *read, write, notify*
The default size of the bounding box around the target.

vidhance.track.enabled *read, write, notify*
Enables / disables pointer interaction.

vidhance.track.maximumsize *read, write, notify*
The maximum bounding box size around the target.

vidhance.track.minimumsize *read, write, notify*
The minimum bounding box size around the target.

vidhance.track.mode *read, write, notify*
The current tracking mode [adaptive | static].

vidhance.track.quality *read, notify*
Quality of tracked object region bounds, as a value between 0 (bad) and 1 (good).

11.2 Methods

vidhance.track.start <start>
Start tracking of object in the region specified by start.
Example: `vidhance.track.start 0.1, -0.3`

vidhance.track.stop
Stop tracking current object.

12 Viewer

12.1 Properties

viewer.contentsize *read, notify*
The size of the content viewed, in pixels (width, height).

viewer.rotation *read, write, notify*
The media rotation in the viewer, given in radians.

viewer.scalemode *read, write, notify*
Defines how scaling reacts to changes in viewer and content size.

viewer.scaling *read, write, notify*
The media scaling in the viewer.

viewer.serial *read, notify*
Serial number of content currently shown.

viewer.size *read, notify*
The size of the viewing area in pixels (width, height).

viewer.transform *read, write, notify*
The media 2D 3x2 transform matrix [a, b, c, d, e, f].

viewer.translation *read, write, notify*
The media translation within the viewer (x,y).

viewer.updaterate *read, notify*
The current update rate of the content viewed, given in Hz.

12.2 Methods

viewer.fill
Set video scaling so that the video fills the viewer without any borders.

viewer.fit
Set video scaling so that the video fills the viewer and still shows the full video.

13 Viewer.overlays.cross

13.1 Properties

viewer.overlays.cross.align *read, write*
The alignment of this overlay [Left | Center | Right | Top | Middle | Bottom | LeftTop | CenterTop | RightTop | LeftMiddle | CenterMiddle | RightMiddle | LeftBottom | CenterBottom | RightBottom].

viewer.overlays.cross.fill *read, write*
The fill color used, e.g. red or #ff0000.

viewer.overlays.cross.fillopacity *read, write*
The fill opacity used, e.g. 0.7 or 1.

viewer.overlays.cross.hidden *read, write*
Whether this overlay is hidden (true) or visible (false).

viewer.overlays.cross.remote *read*
The object type of this remote overlay.

viewer.overlays.cross.size *read*
The size of the overlay when rendered, measured in pixels.

viewer.overlays.cross.sizetag *read, write*

viewer.overlays.cross.stroke *read, write*
The stroke color used, e.g. red or #ff0000.

viewer.overlays.cross.strokeopacity *read, write*
The stroke opacity used, e.g. 0.7 or 1.

viewer.overlays.cross.strokewidth *read, write*
The stroke width, e.g. 1px.

viewer.overlays.cross.transform *read, write*
The 2D 3x2 transform matrix [a, b, c, d, e, f]. See www.w3.org/TR/css3-transforms/#MatrixDefined.

viewer.overlays.cross.transformtag *read, write*

viewer.overlays.cross.type *read*
The object type of this overlay.

14 Viewer.overlays.jitter

14.1 Properties

viewer.overlays.jitter.align *read, write*
The alignment of this overlay [Left | Center | Right | Top | Middle | Bottom | LeftTop | CenterTop | RightTop | LeftMiddle | CenterMiddle | RightMiddle | LeftBottom | CenterBottom | RightBottom].

viewer.overlays.jitter.fill *read, write*
The fill color used, e.g. red or #ff0000.

viewer.overlays.jitter.fillopacity *read, write*
The fill opacity used, e.g. 0.7 or 1.

viewer.overlays.jitter.fontfamily *read, write*
The font family used, e.g. Verdana.
Default: `Verdana`

viewer.overlays.jitter.fontsize *read, write*
Font size, e.g. 12px.
Default: `12px`

viewer.overlays.jitter.fontweight *read, write*
Font weight [normal | bold].

viewer.overlays.jitter.format *read, write*
The format used to display the values of this overlay, e.g. "jitter:\navg {0:fff} ms \nmax {1:fff} ms" (default).

viewer.overlays.jitter.hidden *read, write*
Whether this overlay is hidden (true) or visible (false).

viewer.overlays.jitter.remote *read*
The object type of this remote overlay.

viewer.overlays.jitter.size *read*
The size of the text, measured in pixels.

viewer.overlays.jitter.sizetag *read, write*

viewer.overlays.jitter.stroke *read, write*
The stroke color used, e.g. red or #ff0000.

viewer.overlays.jitter.strokeopacity *read, write*
The stroke opacity used, e.g. 0.7 or 1.

viewer.overlays.jitter.strokewidth *read, write*
The stroke width, e.g. 1px.

viewer.overlays.jitter.textalign *read, write*
Text alignment [start | middle | end].

viewer.overlays.jitter.transform *read, write*
The 2D 3x2 transform matrix [a, b, c, d, e, f]. See www.w3.org/TR/css3-transforms/#MatrixDefined.

viewer.overlays.jitter.transformtag *read, write*

viewer.overlays.jitter.type *read*
The object type of this overlay.

viewer.overlays.jitter.value *read, write*
The text content.

15 Viewer.overlays.latency

15.1 Properties

viewer.overlays.latency.align *read, write*
The alignment of this overlay [Left | Center | Right | Top | Middle | Bottom | LeftTop | CenterTop | RightTop | LeftMiddle | CenterMiddle | RightMiddle | LeftBottom | CenterBottom | RightBottom].

viewer.overlays.latency.fill *read, write*
The fill color used, e.g. red or #ff0000.

viewer.overlays.latency.fillopacity *read, write*
The fill opacity used, e.g. 0.7 or 1.

viewer.overlays.latency.fontfamily *read, write*
The font family used, e.g. Verdana.
Default: `Verdana`

viewer.overlays.latency.fontsize *read, write*
Font size, e.g. 12px.
Default: `12px`

viewer.overlays.latency.fontweight *read, write*
Font weight [normal | bold].

viewer.overlays.latency.format *read, write*
The format used to display the values of this overlay, e.g. "latency: avg {0:fff} ms max {1:fff} ms" (default).

viewer.overlays.latency.hidden *read, write*
Whether this overlay is hidden (true) or visible (false).

viewer.overlays.latency.remote *read*
The object type of this remote overlay.

viewer.overlays.latency.size *read*
The size of the text, measured in pixels.

viewer.overlays.latency.sizetag *read, write*

viewer.overlays.latency.stroke *read, write*
The stroke color used, e.g. red or #ff0000.

viewer.overlays.latency.strokeopacity *read, write*
The stroke opacity used, e.g. 0.7 or 1.

viewer.overlays.latency.strokewidth *read, write*
The stroke width, e.g. 1px.

viewer.overlays.latency.textalign *read, write*
Text alignment [start | middle | end].

viewer.overlays.latency.transform *read, write*
The 2D 3x2 transform matrix [a, b, c, d, e, f]. See www.w3.org/TR/css3-transforms/#MatrixDefined.

viewer.overlays.latency.transformtag *read, write*

viewer.overlays.latency.type *read*
The object type of this overlay.

viewer.overlays.latency.value *read, write*
The text content.

16 Viewer.overlays.logotype

16.1 Properties

viewer.overlays.logotype.align *read, write*
The alignment of this overlay [Left | Center | Right | Top | Middle | Bottom | LeftTop | CenterTop | RightTop | LeftMiddle | CenterMiddle | RightMiddle | LeftBottom | CenterBottom | RightBottom].

viewer.overlays.logotype.fill *read, write*
The fill color used, e.g. red or #ff0000.

viewer.overlays.logotype.fillopacity *read, write*
The fill opacity used, e.g. 0.7 or 1.

viewer.overlays.logotype.hidden *read, write*
Whether this overlay is hidden (true) or visible (false).

viewer.overlays.logotype.remote *read*
The object type of this remote overlay.

viewer.overlays.logotype.size *read*
The size of the overlay when rendered, measured in pixels.

viewer.overlays.logotype.sizetag *read, write*

viewer.overlays.logotype.stroke *read, write*
The stroke color used, e.g. red or #ff0000.

viewer.overlays.logotype.strokeopacity *read, write*
The stroke opacity used, e.g. 0.7 or 1.

viewer.overlays.logotype.strokewidth *read, write*
The stroke width, e.g. 1px.

viewer.overlays.logotype.transform *read, write*
The 2D 3x2 transform matrix [a, b, c, d, e, f]. See www.w3.org/TR/css3-transforms/#MatrixDefined.

viewer.overlays.logotype.transformtag *read, write*

viewer.overlays.logotype.type *read*
The object type of this overlay.

17 Viewer.overlays.northarrow

17.1 Properties

viewer.overlays.northarrow.align *read, write*
The alignment of this overlay [Left | Center | Right | Top | Middle | Bottom | LeftTop | CenterTop | RightTop | LeftMiddle | CenterMiddle | RightMiddle | LeftBottom | CenterBottom | RightBottom].

viewer.overlays.northarrow.fill *read, write*
The fill color used, e.g. red or #ff0000.

viewer.overlays.northarrow.fillopacity *read, write*
The fill opacity used, e.g. 0.7 or 1.

viewer.overlays.northarrow.hidden *read, write*
Whether this overlay is hidden (true) or visible (false).

viewer.overlays.northarrow.remote *read*
The object type of this remote overlay.

viewer.overlays.northarrow.size *read*
The size of the overlay when rendered, measured in pixels.

viewer.overlays.northarrow.sizetag *read, write*

viewer.overlays.northarrow.stroke *read, write*
The stroke color used, e.g. red or #ff0000.

viewer.overlays.northarrow.strokeopacity *read, write*
The stroke opacity used, e.g. 0.7 or 1.

viewer.overlays.northarrow.strokewidth *read, write*
The stroke width, e.g. 1px.

viewer.overlays.northarrow.transform *read, write*
The 2D 3x2 transform matrix [a, b, c, d, e, f]. See www.w3.org/TR/css3-transforms/#MatrixDefined.

viewer.overlays.northarrow.transforma *read, write*
The 2D 3x2 transform matrix [a, b, c, d, e, f] for transform A.

viewer.overlays.northarrow.transformb *read, write*
The 2D 3x2 transform matrix [a, b, c, d, e, f] for transform B.

viewer.overlays.northarrow.transformtag *read, write*

viewer.overlays.northarrow.type *read*
The object type of this overlay.

18 Viewer.overlays.rate

18.1 Properties

viewer.overlays.rate.align *read, write*
The alignment of this overlay [Left | Center | Right | Top | Middle | Bottom | LeftTop | CenterTop | RightTop | LeftMiddle | CenterMiddle | RightMiddle | LeftBottom | CenterBottom | RightBottom].

viewer.overlays.rate.fill *read, write*
The fill color used, e.g. red or #ff0000.

viewer.overlays.rate.fillopacity *read, write*
The fill opacity used, e.g. 0.7 or 1.

viewer.overlays.rate.fontfamily *read, write*
The font family used, e.g. Verdana.
Default: `Verdana`

viewer.overlays.rate.fontsize *read, write*
Font size, e.g. 12px.
Default: `12px`

viewer.overlays.rate.fontweight *read, write*
Font weight [normal | bold].

viewer.overlays.rate.format *read, write*
The string format used to display the value of this overlay, e.g. "{0:F1} Hz" (default).

viewer.overlays.rate.hidden *read, write*
Whether this overlay is hidden (true) or visible (false).

viewer.overlays.rate.remote *read*
The object type of this remote overlay.

viewer.overlays.rate.size *read*
The size of the text, measured in pixels.

viewer.overlays.rate.sizetag *read, write*

viewer.overlays.rate.stroke *read, write*
The stroke color used, e.g. red or #ff0000.

viewer.overlays.rate.strokeopacity *read, write*
The stroke opacity used, e.g. 0.7 or 1.

viewer.overlays.rate.strokewidth *read, write*
The stroke width, e.g. 1px.

viewer.overlays.rate.textalign *read, write*
Text alignment [start | middle | end].

viewer.overlays.rate.transform *read, write*
The 2D 3x2 transform matrix [a, b, c, d, e, f]. See www.w3.org/TR/css3-transforms/#MatrixDefined.

viewer.overlays.rate.transformtag *read, write*

viewer.overlays.rate.type *read*
The object type of this overlay.

viewer.overlays.rate.value *read, write*
The text content.

19 Viewer.overlays.text1

19.1 Properties

viewer.overlays.text1.align *read, write*
The alignment of this overlay [Left | Center | Right | Top | Middle | Bottom | LeftTop | CenterTop | RightTop | LeftMiddle | CenterMiddle | RightMiddle | LeftBottom | CenterBottom | RightBottom].

viewer.overlays.text1.fill *read, write*
The fill color used, e.g. red or #ff0000.

viewer.overlays.text1.fillopacity *read, write*
The fill opacity used, e.g. 0.7 or 1.

viewer.overlays.text1.fontfamily *read, write*
The font family used, e.g. Verdana.
Default: `Verdana`

viewer.overlays.text1.fontsize *read, write*
Font size, e.g. 12px.

Default: 12px

viewer.overlays.text1.fontweight *read, write*
Font weight [normal | bold].

viewer.overlays.text1.hidden *read, write*
Whether this overlay is hidden (true) or visible (false).

viewer.overlays.text1.remote *read*
The object type of this remote overlay.

viewer.overlays.text1.size *read*
The size of the text, measured in pixels.

viewer.overlays.text1.sizetag *read, write*

viewer.overlays.text1.stroke *read, write*
The stroke color used, e.g. red or #ff0000.

viewer.overlays.text1.strokeopacity *read, write*
The stroke opacity used, e.g. 0.7 or 1.

viewer.overlays.text1.strokewidth *read, write*
The stroke width, e.g. 1px.

viewer.overlays.text1.textalign *read, write*
Text alignment [start | middle | end].

viewer.overlays.text1.transform *read, write*
The 2D 3x2 transform matrix [a, b, c, d, e, f]. See www.w3.org/TR/css3-transforms/#MatrixDefined.

viewer.overlays.text1.transformtag *read, write*

viewer.overlays.text1.type *read*
The object type of this overlay.

viewer.overlays.text1.value *read, write*
The text content.

20 Viewer.overlays.text2

20.1 Properties

viewer.overlays.text2.align *read, write*
The alignment of this overlay [Left | Center | Right | Top | Middle | Bottom | LeftTop | CenterTop | RightTop | LeftMiddle | CenterMiddle | RightMiddle | LeftBottom | CenterBottom | RightBottom].

viewer.overlays.text2.fill *read, write*
The fill color used, e.g. red or #ff0000.

viewer.overlays.text2.fillopacity *read, write*
The fill opacity used, e.g. 0.7 or 1.

viewer.overlays.text2.fontfamily *read, write*
The font family used, e.g. Verdana.
Default: `Verdana`

viewer.overlays.text2.fontsize *read, write*
Font size, e.g. 12px.
Default: `12px`

viewer.overlays.text2.fontweight *read, write*
Font weight [normal | bold].

viewer.overlays.text2.hidden *read, write*
Whether this overlay is hidden (true) or visible (false).

viewer.overlays.text2.remote *read*
The object type of this remote overlay.

viewer.overlays.text2.size *read*
The size of the text, measured in pixels.

viewer.overlays.text2.sizetag *read, write*

viewer.overlays.text2.stroke *read, write*
The stroke color used, e.g. red or #ff0000.

viewer.overlays.text2.strokeopacity *read, write*
The stroke opacity used, e.g. 0.7 or 1.

viewer.overlays.text2.strokewidth *read, write*
The stroke width, e.g. 1px.

viewer.overlays.text2.textalign *read, write*
Text alignment [start | middle | end].

viewer.overlays.text2.transform *read, write*
The 2D 3x2 transform matrix [a, b, c, d, e, f]. See www.w3.org/TR/css3-transforms/#MatrixDefined.

viewer.overlays.text2.transformtag *read, write*

viewer.overlays.text2.type *read*
The object type of this overlay.

viewer.overlays.text2.value *read, write*
The text content.

21 Viewer.overlays.track

21.1 Properties

viewer.overlays.track.align *read, write*
The alignment of this overlay [Left | Center | Right | Top | Middle | Bottom | LeftTop | CenterTop | RightTop | LeftMiddle | CenterMiddle | RightMiddle | LeftBottom | CenterBottom | RightBottom].

viewer.overlays.track.fill *read, write*
The fill color used, e.g. red or #ff0000.

viewer.overlays.track.fillopacity *read, write*
The fill opacity used, e.g. 0.7 or 1.

viewer.overlays.track.hidden *read, write*
Whether this overlay is hidden (true) or visible (false).

viewer.overlays.track.remote *read*
The object type of this remote overlay.

viewer.overlays.track.size *read*
The size of the overlay when rendered, measured in pixels.

viewer.overlays.track.sizetag *read, write*

viewer.overlays.track.stroke *read, write*
The stroke color used, e.g. red or #ff0000.

viewer.overlays.track.strokeopacity *read, write*
The stroke opacity used, e.g. 0.7 or 1.

viewer.overlays.track.strokewidth *read, write*
The stroke width, e.g. 1px.

viewer.overlays.track.transform *read, write*
The 2D 3x2 transform matrix [a, b, c, d, e, f]. See www.w3.org/TR/css3-transforms/#MatrixDefined.

viewer.overlays.track.transformtag *read, write*

viewer.overlays.track.type *read*
The object type of this overlay.

22 Viewer.overlays.watch

22.1 Properties

viewer.overlays.watch.align *read, write*
The alignment of this overlay [Left | Center | Right | Top | Middle | Bottom | LeftTop | CenterTop | RightTop | LeftMiddle | CenterMiddle | RightMiddle | LeftBottom | CenterBottom | RightBottom].

viewer.overlays.watch.fill *read, write*
The fill color used, e.g. red or #ff0000.

viewer.overlays.watch.fillopacity *read, write*
The fill opacity used, e.g. 0.7 or 1.

viewer.overlays.watch.fontfamily *read, write*
The font family used, e.g. Verdana.
Default: `Verdana`

viewer.overlays.watch.fontsize *read, write*
Font size, e.g. 12px.
Default: `12px`

viewer.overlays.watch.fontweight *read, write*
Font weight [normal | bold].

viewer.overlays.watch.hidden *read, write*
Whether this overlay is hidden (true) or visible (false).

viewer.overlays.watch.remote *read*
The object type of this remote overlay.

viewer.overlays.watch.size *read*
The size of the text, measured in pixels.

viewer.overlays.watch.sizetag *read, write*

viewer.overlays.watch.stroke *read, write*
The stroke color used, e.g. red or #ff0000.

viewer.overlays.watch.strokeopacity *read, write*
The stroke opacity used, e.g. 0.7 or 1.

viewer.overlays.watch.strokewidth *read, write*
The stroke width, e.g. 1px.

viewer.overlays.watch.textalign *read, write*
Text alignment [start | middle | end].

viewer.overlays.watch.transform *read, write*
The 2D 3x2 transform matrix [a, b, c, d, e, f]. See www.w3.org/TR/css3-transforms/#MatrixDefined.

viewer.overlays.watch.transformtag *read, write*

viewer.overlays.watch.type *read*
The object type of this overlay.

viewer.overlays.watch.value *read, write*
The text content.

23 Window

23.1 Properties

window.border *read, write*
The window border type [resizable | fixed | hidden].

window.closeaction *read, write*
Which action to perform when the window is closed [close | hide | reject].

window.handle *read, write*
The OS handle of the window.

window.position *read, write, notify*
The window position, expressed as x,y.

window.showintaskbar *read, write*
Whether or not to show the window in the Taskbar [true | false].

window.size *read, write, notify*
The size of the window, excluding border, as width,height.

window.state *read, write*
The window state [minimized | normal | maximized].

window.title *read, write, notify*
The window title.

window.topmost *read, write*
Whether to keep the window topmost (always on top) [true | false].

window.totalsize *read, write, notify*
The size of the window, including border, as width,height.

window.visible *read, write, notify*
Whether the window is visible [true | false].