Sherlock 7 Technical Resource

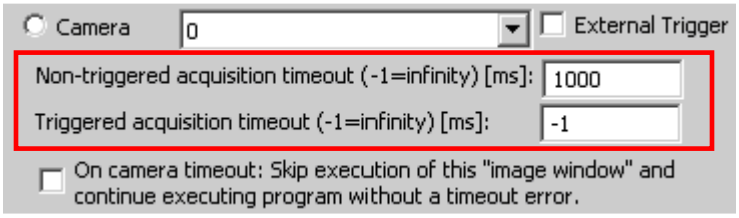
Teledyne DALSA Corp., Industrial Products (IPD)
www.goipd.com 978.670.2002 (U.S.A.)

Document Revision: August 9, 2012

Acquisition Timeouts

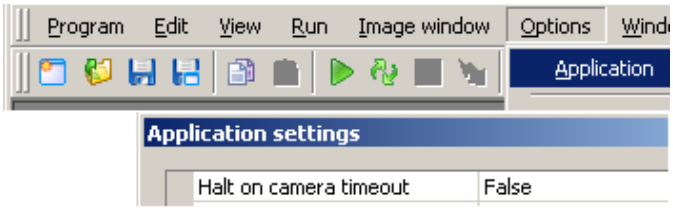
In many applications, the failure of an acquisition to complete within a defined amount of time is an error that needs to be detected and dealt with. Sherlock has several options for handling acquisition timeouts; which options you select depend on whether the application is written to recover from an acquisition failure, or to terminate because a failure indicates a catastrophic system failure.

Setting timeouts



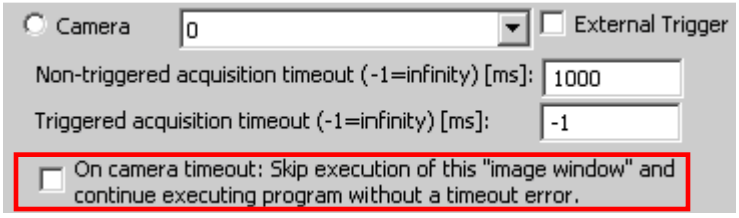
Whether acquisition is non-triggered (also called free-running) or triggered by an external hardware signal, you can set an acquisition timeout (in milliseconds) on an image window's **Options → Image Source** page. If an acquisition does not occur within the timeout period, Sherlock generates an acquisition timeout error.

Halting the investigation



An entry in the **Options → Application** menu determines whether an acquisition timeout error causes the investigation to halt. If this entry is set to **True** and an acquisition timeout error is generated, the investigation stops running immediately, as if the **Abort investigation** button were clicked. (See **Automatic subroutine execution** below.) **Sherlock itself does not shut down.** If this entry is set to **False** and an acquisition timeout error is generated, the investigation continues to run.

Suppressing timeout errors



You can suppress acquisition timeout errors for an image window by checking its **On camera timeout…** box. If this box is checked and an acquisition times out, no error is generated.
The ROIs and any instructions within the scope of the image window are not executed, since the image window will contain either an image that was already processed on the last pass through the investigation, or garbage.

Interaction of options

Care must be taken when setting the **Options → Application → Halt on camera timeout** and image windows' **On camera timeout** options, as their interaction can result in unexpected results.

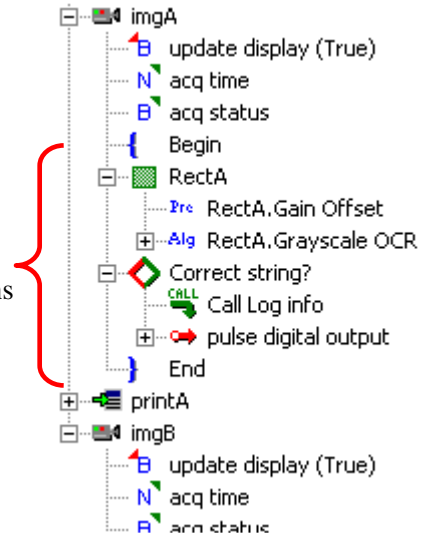
		Options → Application → Halt on camera timeout	
		False	**True**
Image window's **On camera timeout…**	**Disabled** (unchecked)	The ROIs in the image window and any instructions within the scope of the image window <u>are</u> executed. Execution of the investigation continues normally.	The investigation halts immediately. The ROIs in the image window and any instructions within the scope of the image window <u>are</u> <u>not</u> executed.
	Enabled (checked)	The ROIs in the image window and any instructions within the scope of the image window <u>are</u> <u>not</u> executed. Execution continues at the first program element outside the scope of the image window.	

Examples

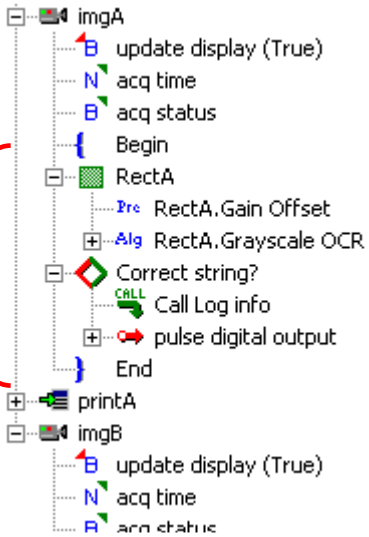
Options → Application → Halt on camera timeout is set to **False**
ImgA's **On camera timeout…** is disabled.
ImgA's acquisition times out

Because **On camera timeout…** is disabled, everything within the scope of the image window is executed (**RectA**, the **If** statement **Correct string?** and its dependant instructions **Call Log info** and **pulse digital output**)

Execution continues normally



Options → Application → Halt on camera timeout is set to **False**
ImgA's **On camera timeout…** is enabled.
ImgA's acquisition times out.

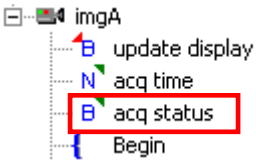


Because **On camera timeout…** is enabled, nothing within the scope of the image window is executed (**RectA**, the **If** statement **Correct string?** and its dependant instructions **Call Log info** and **pulse digital output**).

Execution continues at the instruction **printA.**

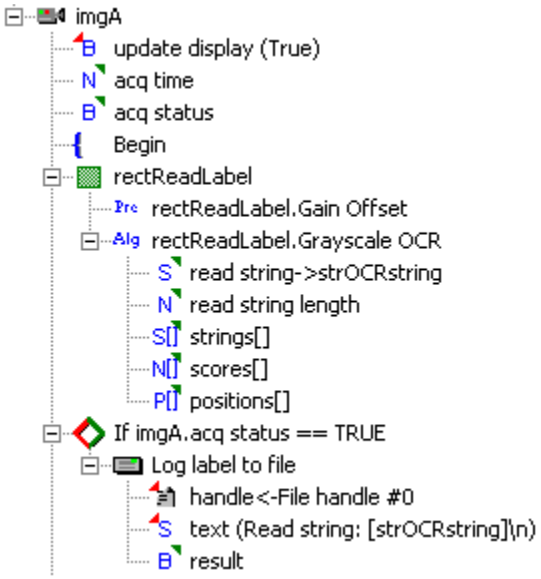
Image window acquisition status

If an image window's acquisition times out, its Boolean reading **acq status** is set to **False**; otherwise it is set to **True**.



Example

Options → Application → Halt on camera timeout is set to **False**
ImgA's **On camera timeout…** is disabled.
ImgA's acquisition times out.



rectReadLabel is executed.

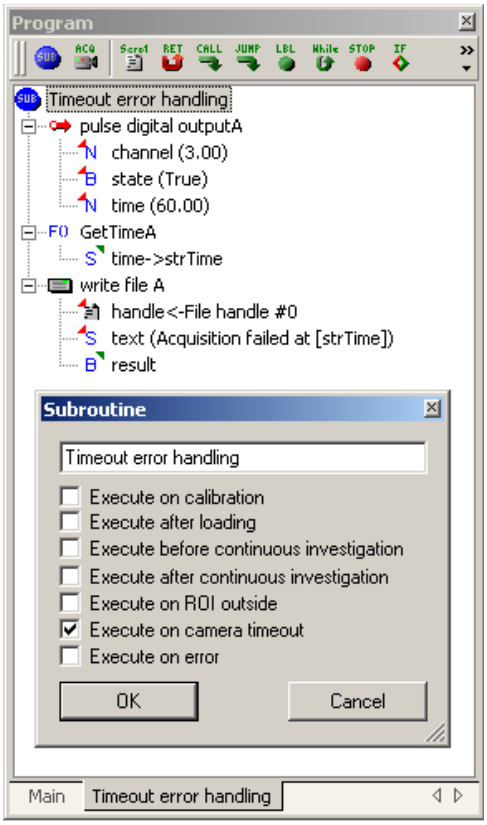
ImgA's **acq status** is **False**, so the instruction to write a line to a text file is <u>not</u> executed.

Automatic subroutine execution

A subroutine that is set to **Execute on camera timeout** is executed as soon as a timeout occurs, before the investigation is terminated by the **Options → Application → Halt on camera timeout → True** option.

If multiple subroutines are set to **Execute on camera timeout,** they are executed in the order in which they were created.

An image window whose **On camera timeout…** is enabled does not generate a timeout error if its acquisition times out, and therefore does not cause automatic subroutine execution.



Example scenarios

For all scenarios, the image windows' acquisition timeouts are set to values other than –1 (infinity).

For scenarios 1 thru 3, **Options → Application → Halt on camera timeout** is set to **False.**

1. Calling a subroutine and continuing the investigation

A subroutine **subTimeout** is marked to **Execute on camera timeout.**
Image window **imgA**'s **On camera timeout…** is disabled.

- **ImgA**'s acquisition times out.
- **SubTimeout** is called.
- **ImgA**'s ROIs are executed.
- Execution continues normally.

2. Suppressing the timeout error

A subroutine **subTimeout** is marked to **Execute on camera timeout**.
Image window **imgA**'s **On camera timeout…** is disabled.
Image window **imgB**'s **On camera timeout…** is enabled.

- **ImgA**'s acquisition times out.
- **SubTimeout** is called.
- **ImgA's** ROIs are executed.
- **ImgB**'s acquisition times out.
- **SubTimeout** is not called, because **imgB**'s acquisition timeout error was suppressed.

- **ImgB**'s ROIs are not executed.
- Execution continues at the first program element outside the scope of **imgB**.

3. Checking the image window status

Image window **imgA'**s **On camera timeout…** is disabled.

- **ImgA**'s acquisition times out.
- **ImgA'**s ROIs are executed.
- An **If** instruction outside the scope of **imgA** checks the value of **imgA.acq status**; if it is **True**, instructions that analyze the readings from the ROIs in **imgA** are executed; otherwise, they are skipped. (If **imgA.acq status** is **False**, a new image was not acquired, so any readings from the ROIs will be invalid.)

For scenario 4, **Options → Application → Halt on camera timeout** is set to **True.**

4. Calling a subroutine and terminating the investigation

A subroutine **subTimeout** is marked to **Execute on camera timeout**.

- **ImgA**'s acquisition times out.
- **SubTimeout** is called.
- Execution of the investigation is terminated.

(Whether **imgA**'s **On camera timeout…** is disabled or enabled is irrelevant.)

 If acquisition timeout is set to –1 (infinity) and an acquisition never completes, you will have to terminate Sherlock with the **Windows Task Manager.**