

Sherlock 7 Technical Resource

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

August 9, 2012

Handling runtime errors

A runtime error is generated when an element of the investigation cannot perform its intended task. There are three types of runtime errors:

- Acquisition timeout
- ROI outside of an image window
- General

This document explains the options available for trapping and handling ROI errors and general errors. The options for handling acquisition timeout errors are explained in a separate document.

ROI outside of image window

An ROI can be positioned partially or completely outside an image window as the result of an alignment, or by a call to **IO : ROI : Offset**, **: Set Coord**, or **: Set Rotation.** (Or, less likely, by the user manually moving the ROI.) This generates an error because the ROI cannot process pixels that don't exist.

The **Options** \rightarrow **Application** dialog includes the entry **Halt on ROI outside of image window**. If this entry is set to **True**, an investigation aborts when an ROI that is positioned outside of an image window executes; if it is set to **False**, the investigation continues to execute.



In either case, subroutines marked **Execute on ROI** outside are called as soon as the error occurs, before the investigation either aborts or continues. You can add code to these subroutines to log errors, to try to correct the problem that caused the ROI to fall outside the image window, etc.



General errors

Runtime errors other than acquisition timeouts or ROIs being positioned outside of image windows fall into the category of general errors. These include mathematical errors (e.g., divide by 0, square root of a negative number), missing parameters (e.g., "<EMPTY>" for the **calibration** parameter of the **IO : Calibrate : Calibrate Using Point Arrays** instruction), and others.

Handling runtime errors Page 1

v1.1 August 9, 2012

By default, an investigation aborts on a general error. Unlike **Halt on ROI outside of image** window and **Halt on camera timeout**, there is no entry in **Options** \rightarrow **Application** that you can set to **False** to change this behavior.

Subroutines marked **Execute on error** are executed before the investigation aborts.

| Program | | | | | _ | | _ | _ | - |
|---------------------------------------|--------|------|--------|------|------------|------|-----|--------|---|
| 🛛 💷 🚳 | Scrot | U | | JUHP | Nhile U | STOP | ۰. | ↓ ↓ | T |
| 🕮 error G | ienera | al | | | | | | | |
| ^{©.} Subro | utin | e | | | | | | × | 1 |
| erro | r Gen | eral | | | | | | | |
| , | | | | | | | | | |
| I I I I I I I I I I I I I I I I I I I | Execu | te o | n erro | n | | | | | |
| | OK | | | | | Can | cel | | |
| | | | | | | | | | 1 |

IO : System : Set error handling strategy

The error-handling behaviors discussed above are set at the Sherlock application level, and are applied to an investigation as it is loaded. These settings can be overridden within an investigation with the **IO** : System : Set error handling strategy instruction. (Note: This instruction was implemented in Sherlock 7.1.4.0.) This instruction has three options:

- Halt Call subroutines marked Execute on error and Execute on ROI outside as appropriate, then abort the investigation.
- **Ignore error** Call subroutines marked **Execute on error** and **Execute on ROI outside** as appropriate, then return control to the instruction after the instruction that generated the error.
- **Restart** Call subroutines marked **Execute on error** and **Execute on ROI outside** as appropriate, then return control to the **Main** routine and a start new iteration of the program. **This option is meaningful only if the investigation was launched in continuous mode**. The investigation is restarted in continuous mode. Subroutines marked **Execute before continuous investigation** are NOT called automatically when the investigation is restarted. If the investigation was launched to run once, subroutines marked **Execute on error** and **Execute on ROI outside** are called as appropriate, and the investigation aborts.

If an ROI is positioned outside of an image window, only subroutines marked Execute on ROI outside are called; subroutines marked Execute on error are not.

Halt on ROI outside of image window → False and IO : System : Set error handling strategy : Ignore error should be used with great care, since the investigation continues to run as if no error occurred. It is up to you to design the investigation to "behave well." For example, if you set Halt on ROI outside of image window to False and an ROI is positioned outside the image window, the ROI's algorithms will not extract fresh information from the image, and the algorithms' readings will maintain their values from the last successful execution of the ROI; instructions further down in the program that use the readings as input will receive old values.

Handling runtime errors Page 2

v1.1 August 9, 2012

Handling runtime errors Page 3

v1.1 August 9, 2012