The World's Sixth Sense®

Feature Locking in Spinnaker API

Revised 5/2/2017

Applicable products

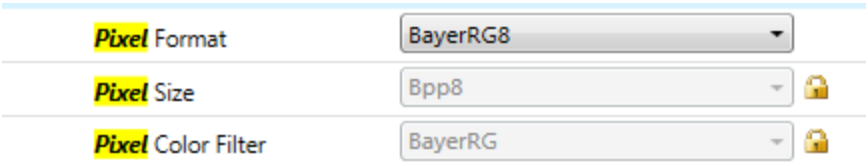
Spinnaker® SDK

Application note description

This application note describes the reasons why some camera features are inaccessible to the API.

Overview

Some camera features cannot be modified. These are identified in SpinView by a lock icon.

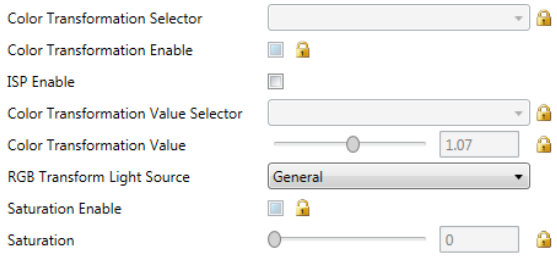


When a lock appears next to a node, it indicates the API cannot write to that node. There are three reasons this might happen:

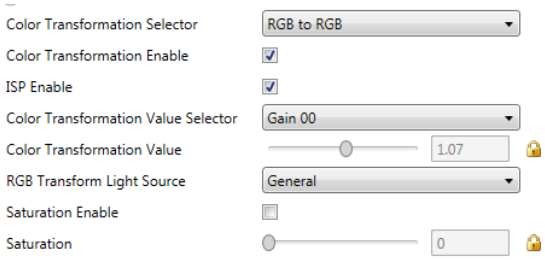
- The node is dependent on another node
- The node is read-only
- The camera is locked because another process is writing to the camera

Dependent nodes

Sometimes the ability to write to a node is dependent on the state of other nodes. For example, the *ISP Enable* node controls several other nodes.



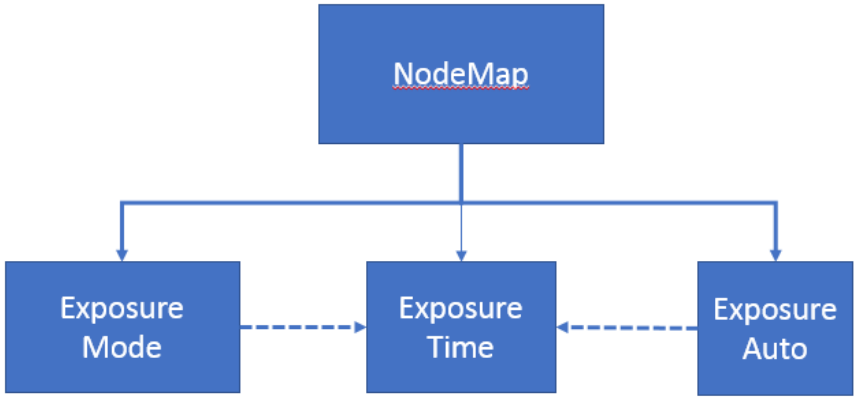
ISP Disabled - Locks Color Transformation Selectors



ISP Enabled - Allows Color Transformation Selectors to be Modified

It is challenging to determine what nodes lock other features as a node can be locked for more than one reason.

In the example below, the solid lines show the inheritance and the dashed lines show dependencies.



Exposure Time is dependent on both Exposure Mode and Exposure Auto, but neither are parents of Exposure Time. Therefore, all the nodes in the NodeMap must be searched to determine the dependencies of Exposure Time.

You can find the dependencies in the camera's XML file, located:
C:\ProgramData\Spinnaker\XML

Commonly locked dependent nodes

The following is a non-exhaustive list of commonly locked nodes with dependencies.

To unlock this node...	Try...
Exposure Time	Setting Exposure Auto to Off
Gain	Setting Gain Auto to Off
Auto Exposure Control Priority	Setting Gain Auto to Continuous Setting Exposure Auto to Continuous
Balance Ratio	Setting Balance White Auto to Off
Frame Rate	Selecting Acquisition Frame Rate Enable
ISP Enable	Setting Pixel Format to Mono, Raw, or Bayer When Pixel Format is set to YCbCr, RGB, YUV, or BGR ISP is enabled and locked.
ISP Enable	Stopping acquisition
Width / Height	
Acquisition Mode	
Binning Mode	
Pixel Format	
Chunk Enable	
Color Transformation Selector	Selecting ISP Enable
Color Transformation Enable	
Color Transformation Value Selector	
Saturation Enable	
Sharpening Enable	
ROI Offset X / ROI Offset Y	Selecting ROI Enable
ROI Width / ROI Height	
Lighting Mode	Setting Target Grey Value Auto to Continuous
Metering Mode	
EV Compensation	

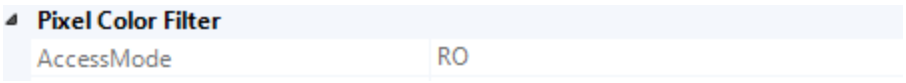
Note: When Sequencer Configuration Mode is set to On, many features become locked, even features not included in the sequencer configuration. After configuring sequencer, set the Sequencer Configuration Mode to Off.

Read-only nodes

For some features, the node is read-only and cannot be unlocked under any circumstances. The node is provided for information purposes.

To determine if a node is read-only:

- In SpinView, double-click on the feature to open the Node Information window. Look for the AccessMode row.



Access	Description
RO	Read-only
RW	Read / Write
N/A	Not Available

Using Spinnaker API, you can poll the Access Mode with the following:

```
//! Checks if a node is readable
inline bool IsReadable (const IBase* p)
{
   return (p != NULL) && IsReadable (p->GetAccessMode());
}
```

Single process writing

Only one process can write to the camera at any one time. When an application is writing to the camera, all other nodes are not available and will not be available until the application finishes writing to the camera. The read and write time to each node is generally very fast and running commands sequentially does not lead to issues, only when writing commands to the camera in parallel does this become important.

Useful API calls

In Spinnaker, every node can be polled to see if they are readable / writeable / available. These states are helpful in catching exceptions. Below are examples of C++ node class functions.

```
//! Checks if a node is readable
inline bool IsReadable(const IBase* p)
{
   return (p != NULL) && IsReadable(p->GetAccessMode());
}
//! Checks if a node is writable
inline bool IsWritable(const IBase* p)
{
   return (p != NULL) && IsWritable(p->GetAccessMode());
}
//! Checks if a node is implemented
inline bool IsImplemented(const IBase* p)
{
   return (p != NULL) && IsImplemented(p->GetAccessMode());
}
//! Checks if a node is available
inline bool IsAvailable(const IBase* p)
{
   return (p != NULL) && IsAvailable(p->GetAccessMode());
}
```

Downloads and support

FLIR endeavors to provide the highest level of technical support possible to our customers. Most support resources can be accessed through the Support section of our website.

The first step in accessing our technical support resources is to obtain a customer login account. This requires a valid name and email address. To apply for a customer login account go to our downloads page.

Customers with a customer login account can access the latest **software** and **firmware** for their cameras from our website. We encourage our customers to keep their software and firmware up-to-date by downloading and installing the latest versions.

Finding information

Spinnaker SDK—The Spinnaker SDK provides API examples and the SpinView camera evaluation application. Available from our Downloads page.

API Documentation—The installation of the Spinnaker SDK comes with API references for C++, C#, and C code. A Programmer's Guide is included in each of the references. Available from:

- Start Menu→All Programs→Point Grey Spinnaker SDK→Documentation
- The SpinView application Help menu

Getting Started with SpinView—A quick guide to using the SpinView camera evaluation application provided in the Spinnaker SDK. Available from:

- Start Menu→All Programs→Point Grey Spinnaker SDK→Documentation
- The SpinView application Help menu
- Camera Reference zip package

Camera Reference—A zip package containing PDF and HTML copies of the camera's references, including: Installation Guide, Technical Reference, and Getting Started. Available from our downloads page.

Knowledge Base—A database of articles and application notes with answers to common questions as well as articles and tutorials about hardware and software systems. Available from our knowledge base.

Learning Center—Our Learning Center contains links to many resources including videos, case studies, popular topics, other application notes, and information on sensor technology.

Contacting technical support

Before contacting Technical Support, have you:

1. Read the product documentation?
2. Searched the knowledge base?
3. Downloaded and installed the latest version of software and/or firmware?

If you have done all the above and still can't find an answer to your question, contact our technical support team.

Additional resources

GenICam—A programming interface for cameras and devices. More information available on the EMVA.org website.

USB3 Vision—A vision standard for the USB 3.1 interface that uses GenICam. More information available on the AIA Vision Online website.