

# LabVIEW SDK Programming Reference Manual

Version: V1.0.1

Date: 2020-05-13

All rights reserved. No parts of this manual may be used or reproduced, in any forms or by any means, without prior written permission of China Daheng Group, Inc. Beijing Image Vision Technology Branch.

The right is also reserved to modify or change any parts of this manual in the future without prior notification.

All other trademarks are the properties of their respective owners.

© 2020 China Daheng Group, Inc. Beijing Image Vision Technology Branch

Web: <http://www.daheng-imaging.com/en>

Sales Email: [isales@daheng-imaging.com](mailto:isales@daheng-imaging.com)

Sales Tel: +86 10 8282 8878

Support Email: [isupport@daheng-imaging.com](mailto:isupport@daheng-imaging.com)

# Contents

<b>1. Camera Workflow .....</b>	<b>1</b>
1.1. Overall workflow .....	1
1.2. Function control flow .....	2
<b>2. Programming Guide .....</b>	<b>3</b>
2.1. Build a programming environment.....	3
2.1.1. About LabVIEW VI.....	3
2.1.2. LabVIEW program development .....	3
2.2. Quick guide .....	4
2.2.1. Initialization and de-initialization.....	4
2.2.2. Enumerate the device.....	5
2.2.3. Open/Close the device .....	5
2.2.4. Acquisition control.....	5
2.2.5. Camera control .....	6
2.2.5.1. Feature controller type.....	6
2.2.5.2. Feature data type.....	7
2.2.6. Error handling .....	9
<b>3. Module Interface Definition.....</b>	<b>11</b>
3.1. Init.vi .....	11
3.2. UnInit.vi .....	11
3.3. UpdateDeviceList.vi.....	11
3.4. OpenDeviceBySN.vi .....	12
3.5. CloseDevice.vi.....	12
3.6. OpenStream.vi .....	12
3.7. CloseStream.vi .....	13
3.8. IsImplemented.vi .....	13
3.9. GetBoolValue.vi .....	13
3.10. SetBoolValue.vi .....	14
3.11. GetIntValue.vi .....	14
3.12. SetIntValue.vi .....	14
3.13. GetFloatValue.vi .....	15
3.14. SetFloatValue.vi .....	15
3.15. GetEnumValue.vi.....	15
3.16. SetEnumValue.vi .....	16
3.17. GetStringValue.vi.....	16
3.18. SetStringValue.vi .....	16
3.19. SetCommandValue.vi .....	17

3.20. StartGrab.vi .....	17
3.21. StopGrab.vi .....	17
3.22. GetImage.vi .....	18
3.23. ImageGrabEvent.vi.....	18
3.24. GrabImageDisplay.vi .....	18
3.25. SaveImage.vi.....	19
<b>4. FAQ .....</b>	<b>20</b>
<b>5. Revision History .....</b>	<b>21</b>

## 1. Camera Workflow

### 1.1. Overall workflow

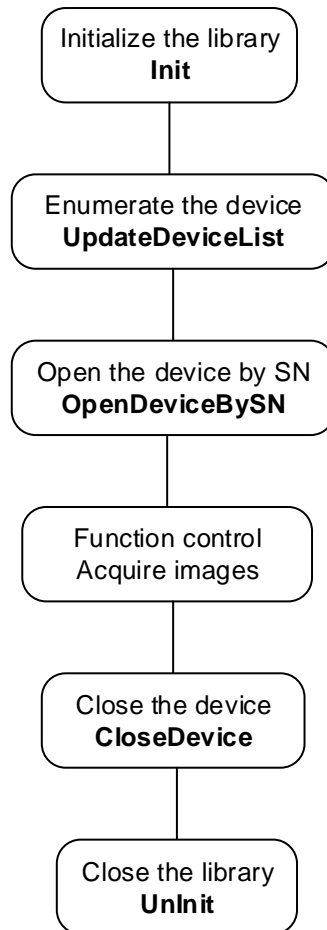


Figure 1-1 Overall workflow

## 1.2. Function control flow

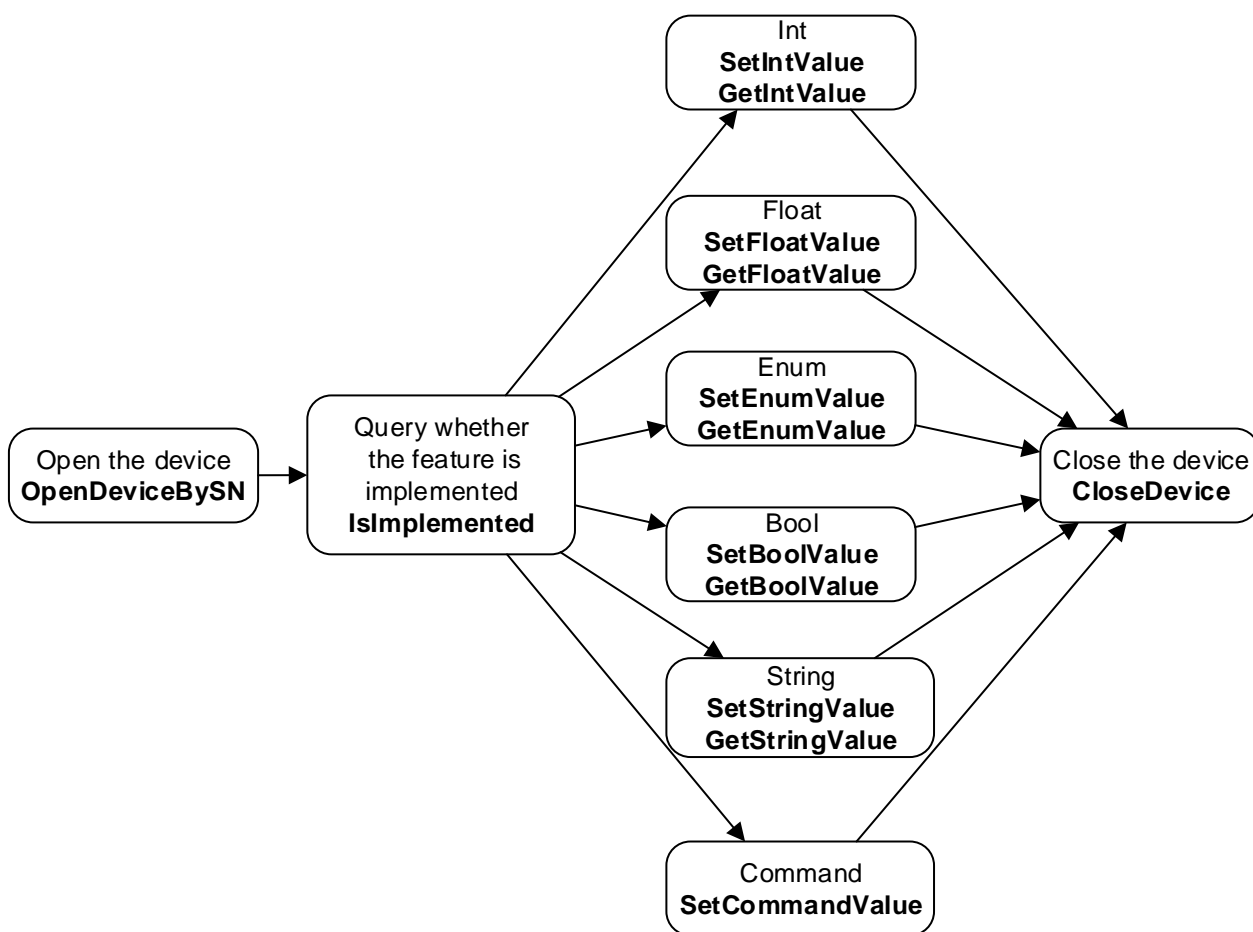


Figure 1-2 Function Control flow

## 2. Programming Guide

### 2.1. Build a programming environment

Support LabVIEW2013 or above version. In addition to installing the LabVIEW program, you need to install the corresponding VISION library.

#### 2.1.1. About LabVIEW VI

In order to make it easy for the user to call SDKs of the.NET interface on the LabVIEW platform, we have encapsulated sub-VIs on the LabVIEW platform. There are 25 sub-VIs in GxLVLib.lvlib. These VIs are secondary encapsulation of part of SDKs of the.NET interface, which internally calls dynamic link libraries of SDKs. And the user can directly call the.NET interface.

#### 2.1.2. LabVIEW program development

First create a new project. In the Project Explorer, right-click **My Computer**, click **Add » File**, and add GxLVLib.lvlib to the project. Create a new VI and name it Demo.vi. Then open Demo.vi, you can add VIs in GxLVLib.lvlib for camera operation in the Block Diagram.

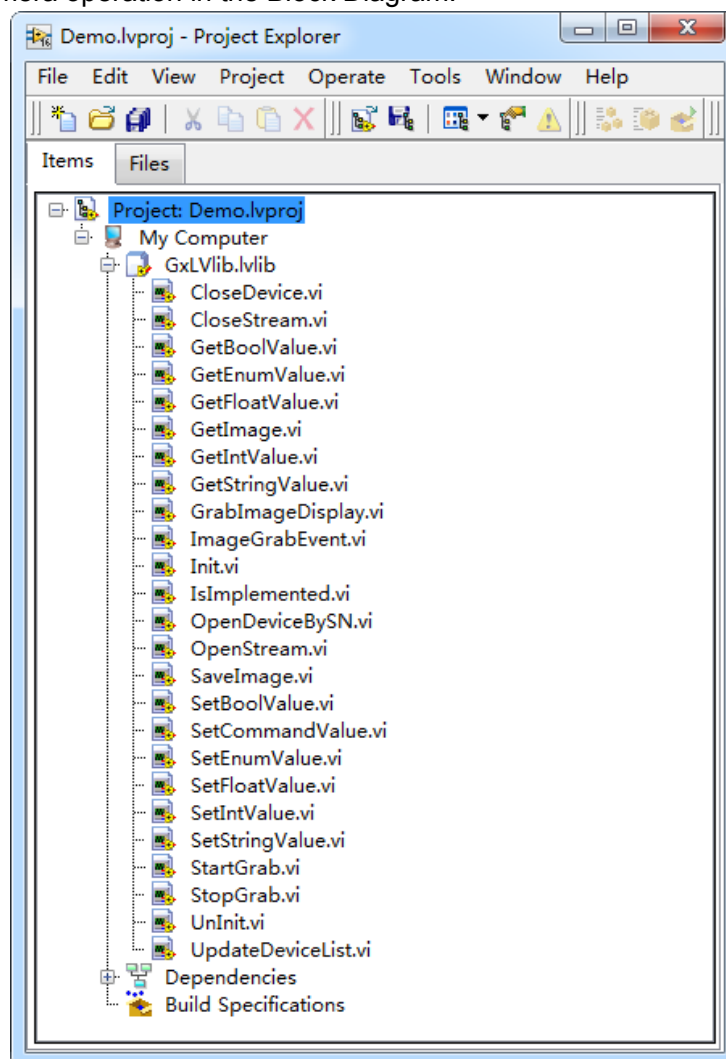


Figure 2-1 VI interface list



During the initialization of VI, the Figure 2-2 may appear.

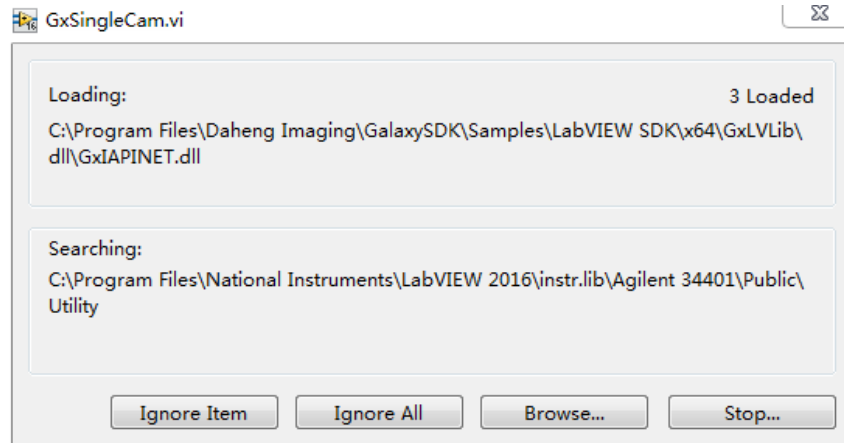


Figure 2-2 Loading VI

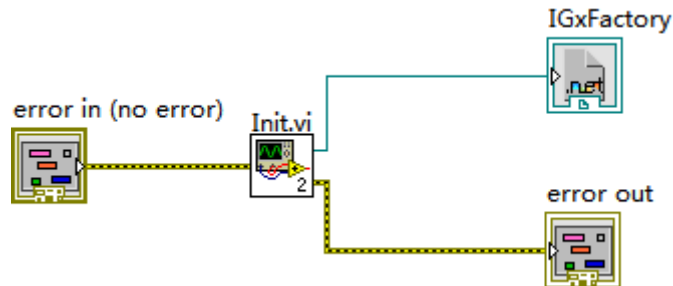
The GxIAPINET.dll called by the VI will be searched automatically. If it is not found, please click **Browse** manually to add the path of GxIAPINET.dll in the SDK.

## 2.2. Quick guide

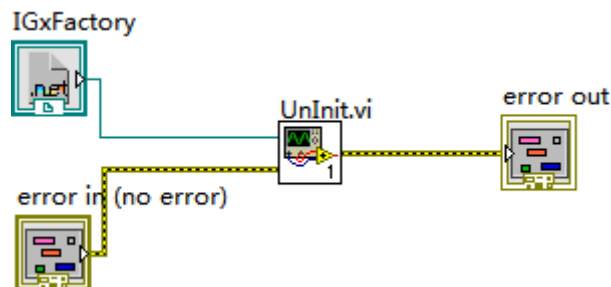
### 2.2.1. Initialization and de-initialization

GxLVLlib.lvlib is a secondary encapsulation of the .NET interface library, and initialization and de-initialization must be performed when use GxIAPINET.

You must call Init.vi to perform initialization before calling other interfaces.



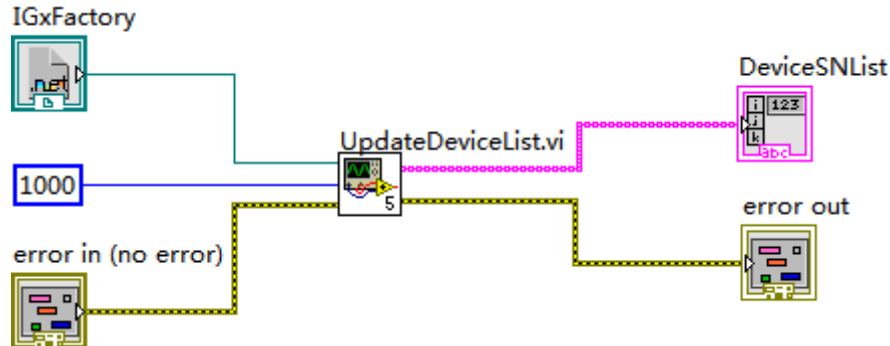
Before exiting the process, you must call UnInit.vi to release all resources requested by GxIAPINET.





### 2.2.2. Enumerate the device

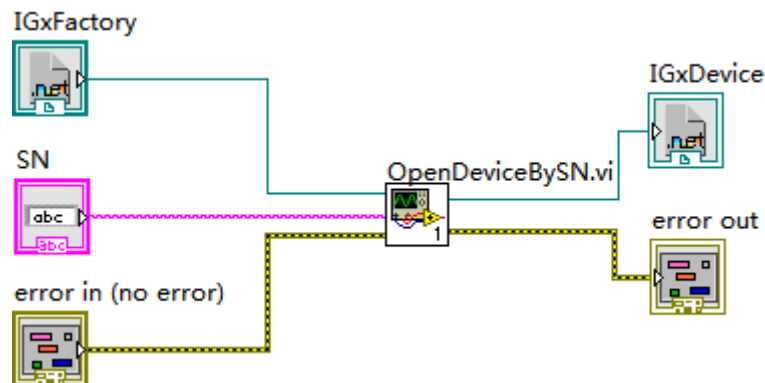
The user can enumerate all available devices by calling UpdateDeviceList.vi, and the function return value is device SN list (DeviceSNList). The number of elements in the device information list is the number of devices enumerated, and the data type of the elements in the list is string.



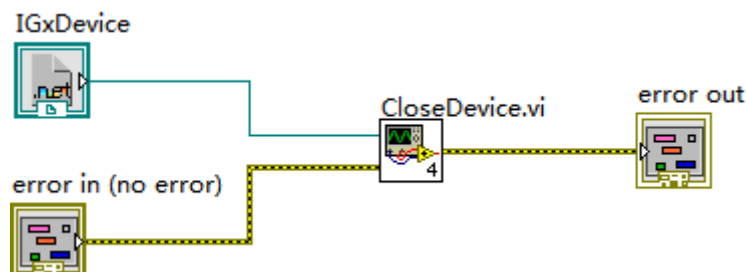
### 2.2.3. Open/Close the device

The user can open the device by OpenDeviceBySN.vi. The input parameter of the function is the SN, and the type is string.

Before opening the device, the user should call UpdateDeviceList.vi to update the device list inside the GxIAPINET library. Otherwise, the device may open failed.



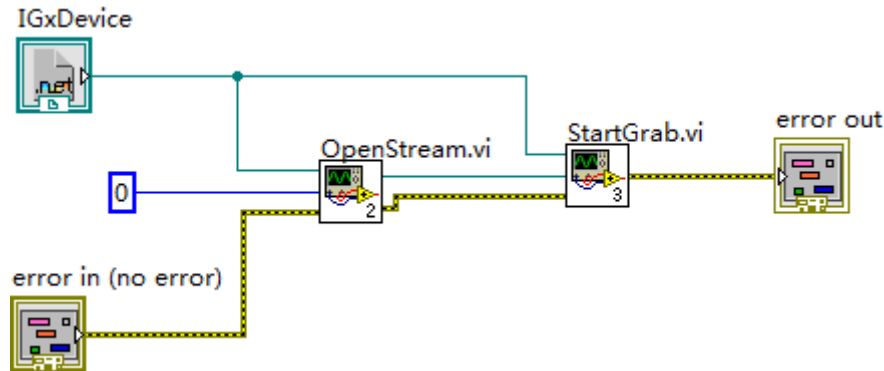
The user can call CloseDevice.vi to close the device and release all device resources.



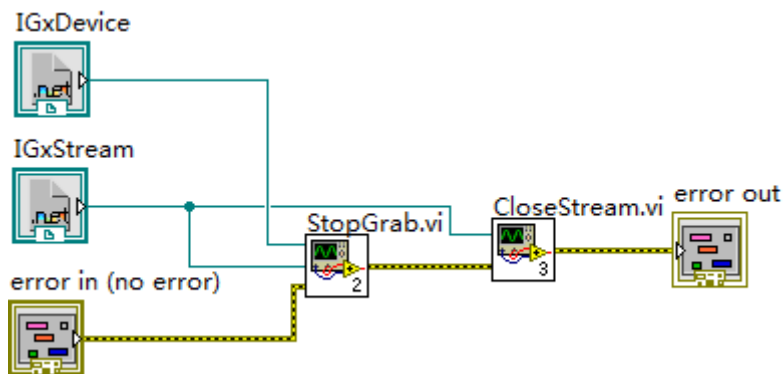
### 2.2.4. Acquisition control

The user can call StartGrab.vi and StopGrab.vi to start and stop acquisition after the device is opened successfully and the camera acquisition parameters are set.

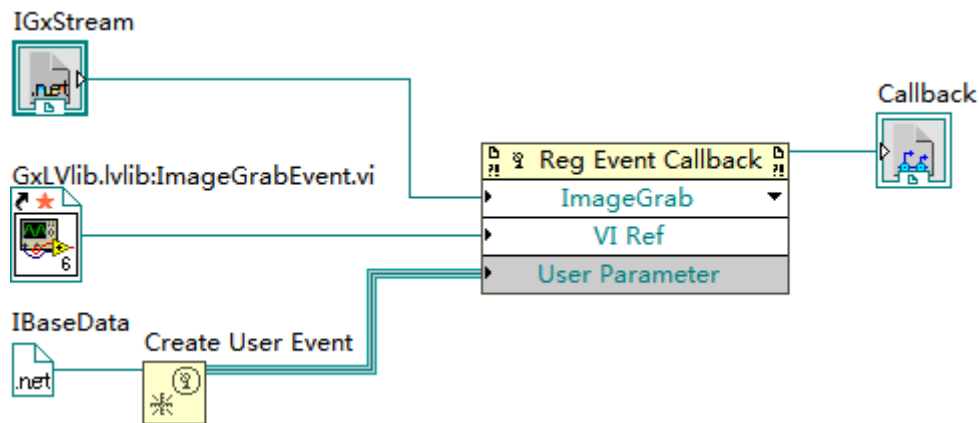
Before starting acquisition, you need to open the data stream of the stream channel. Currently, only the data stream of channel 0 is supported. And then you can execute StartGrab.vi.



Before stopping acquisition, you need to execute StopGrab.vi, and then close the stream channel.



The user can get images through GetImage or callback method. If the callback method is adopted, the callback event needs to be registered by Register Event Callback before start acquisition.



## 2.2.5. Camera control

### 2.2.5.1. Feature controller type

Three feature controllers are available:

Remote device feature controller: such as width, height, exposure, gain, etc. We provide VIs to access and control the remote device feature controller.

Local feature controller: different types of devices have different functions. The user can directly use the IGXDevice::GetFeatureControl function of the IGxAPINET library for control.

Stream object feature controller: a feature access controller for acquisition control and acquisition data statistics. The user can directly use the IGxStream::GetFeatureControl function of the IGxAPINET library for control.

### 2.2.5.2. Feature data type

Six data types of VI interface are available:

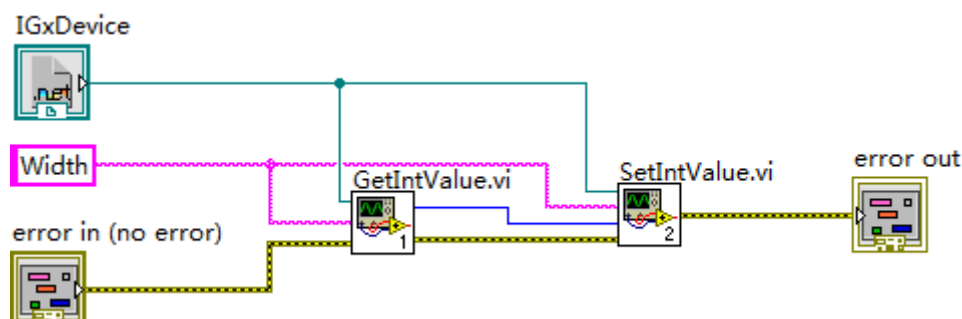
#### Int:

Related interfaces:

SetIntValue.vi //Set

GetIntValue.vi //Get

Sample code:



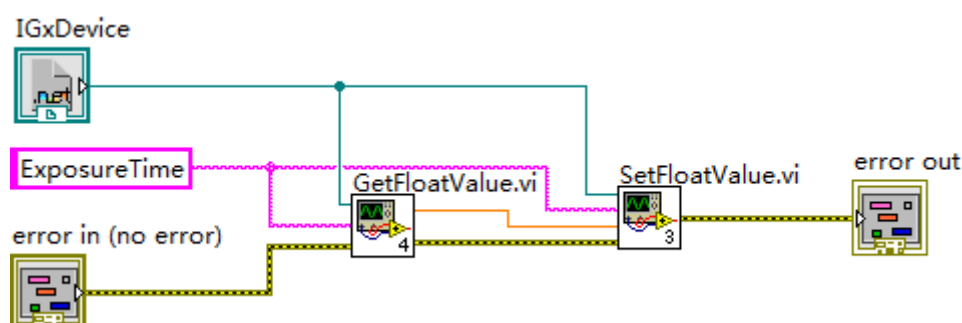
#### Float:

Related interfaces:

SetFloatValue.vi //Set

GetFloatValue.vi //Get

Sample code:



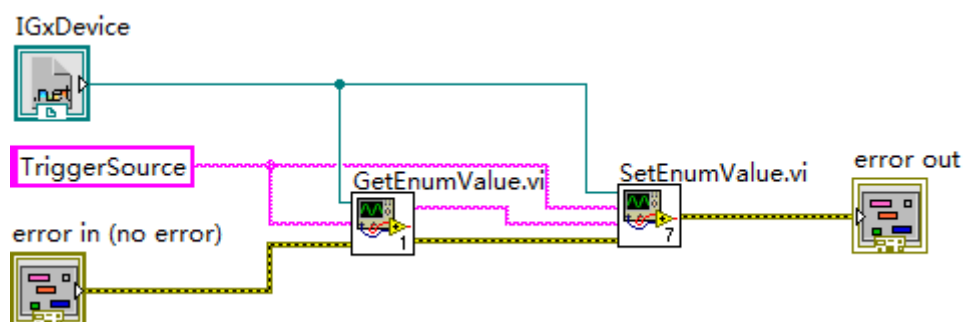
#### Enum:

Related interfaces:

SetEnumValue.vi //Set

GetEnumValue.vi //Get

Sample code:



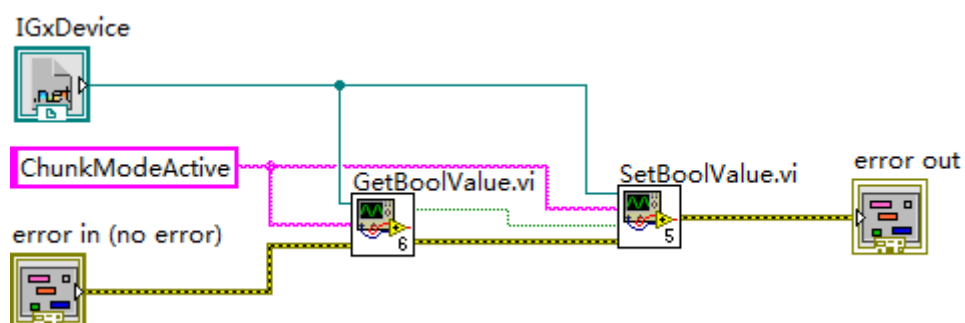
**Bool:**

Related interfaces:

SetBoolValue.vi //Set

GetBoolValue.vi //Get

Sample code:



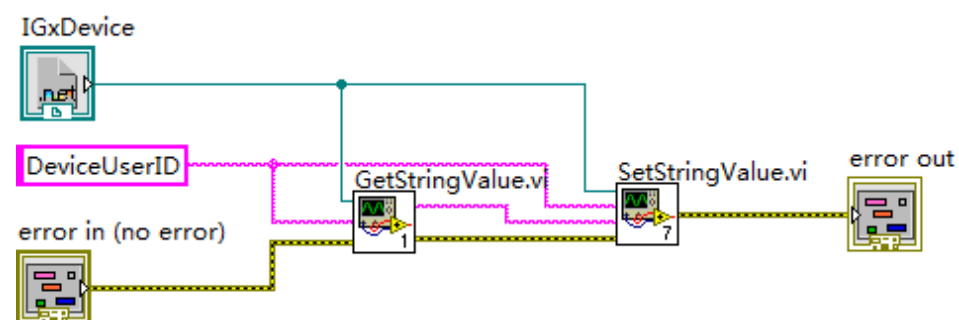
**String:**

Related interfaces:

SetStringValue.vi //Set

GetStringValue.vi //Get

Sample code:

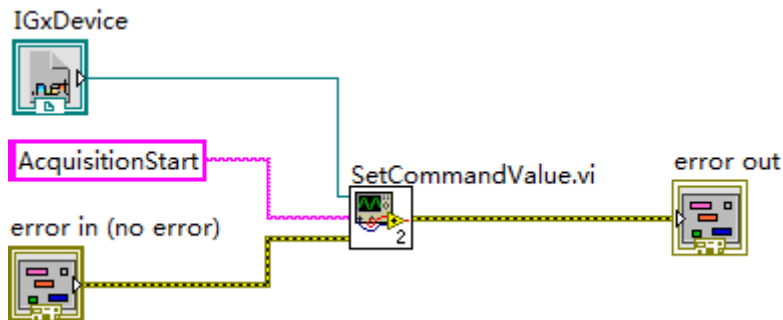


## Command:

Related interfaces:

SetCommandValue.vi //Set

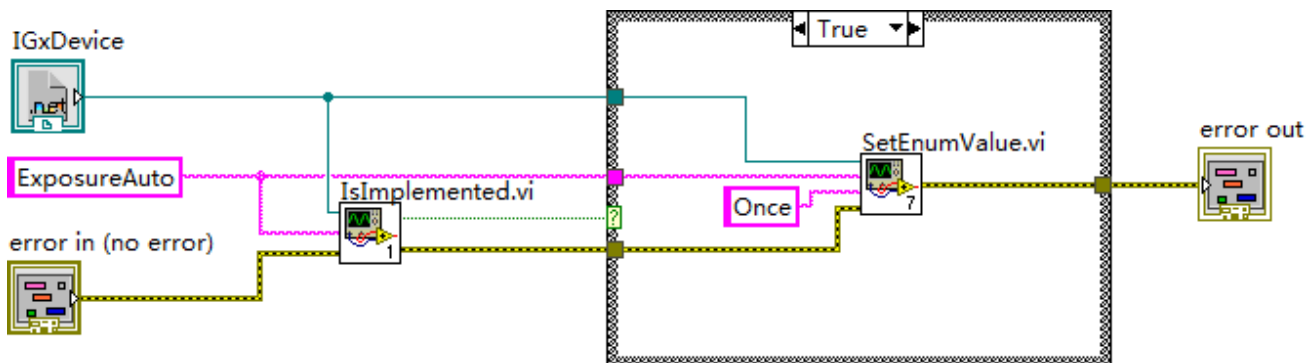
Sample code:



It is recommended that the user query whether the feature is implemented before operating the feature:

Related interfaces (Query whether the feature is implemented):

IsImplemented.vi //Is it implemented



## 2.2.6. Error handling

The error information of LabVIEW is represented by two error cluster widgets: error in and error out. The clusters are composed of a bool (status) representing the status, a 32-bit int (code) representing the error code, and a string (source) representing the source of the error. The error clusters are used to transmit error messages during the execution of LabVIEW code.

The user can use the **Error Code Editor** to user-define error codes as follows.

- 1) Click **Tools » Advanced » Edit Error Codes** to open the **Error Code Editor**.
- 2) In the prompt box that appears, click **New** to create a new error code file or click **Existing** to find the existing error code file.
- 3) Enter comments about the error code file in the **Comments about this file**.

- 4) Click error codes to edit, click the text field to add a text description for the error. Click **Add Error Code** to add another error code to the error code file.
- 5) To edit error codes and descriptions, select the error codes or texts you want to modify, and then enter your changes. Click **Error Code Sort** to sort the error codes in ascending order. To delete an error code, select the error code and click **Delete Selected Error Code**.
- 6) After editing the error code file, click **Save** to save the error code file to the labview\user.lib\errors directory. The error code file must be kept as xxx-errors.txt, where xxx is user-defined.

The error code file we provided is GxLVLib-errors.txt, which is placed in the \GxLVLib\res directory. When you use it, you must copy it to the labview\user.lib\errors directory.

Two types of error are currently provided. You can add and modify them:

Error codes	Descriptions
5000	Invalid param
5001	Not found device

Table 2-1 List of error codes

When the GxIAPINET library reports an error, it is expressed in the form of throwing an exception. And the LabVIEW will directly pop up the exception message, as shown in the Figure 2-3.

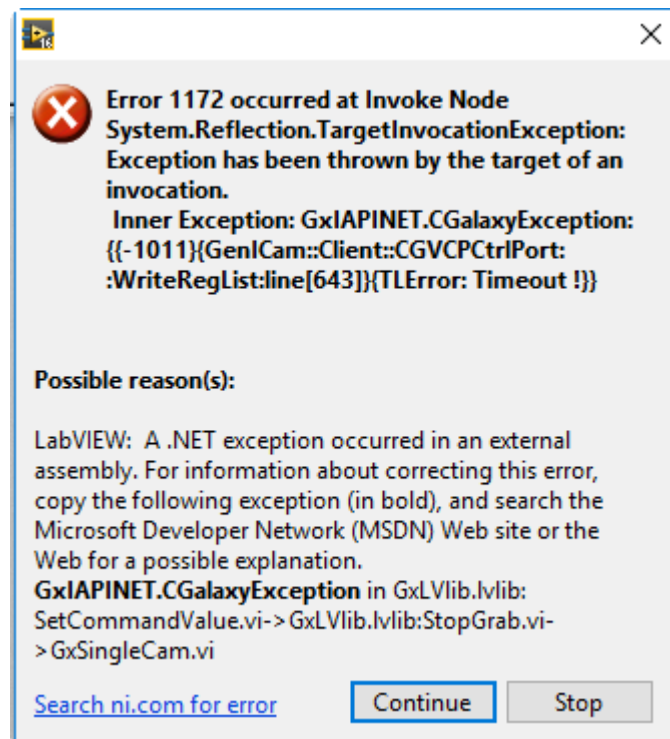


Figure 2-3 Error

## 3. Module Interface Definition

### 3.1. Init.vi

#### Descriptions:

Initialize library resources

#### Parameters:

error in	Error in
IGxFactory	IGXFactory object instance
error out	Error out

### 3.2. UnInit.vi

#### Descriptions:

Release library resources

#### Parameters:

IGxFactory	IGXFactory object instance
error in	Error in
error out	Error out

### 3.3. UpdateDeviceList.vi

#### Descriptions:

Enumerate devices (for GigE devices: subnet enumeration)

#### Parameters:

IGxFactory	IGXFactory object instance
Time out	Enumeration timeout, unit: ms
error in	Error in
DeviceSNList	Device SN list
error out	Error out

### 3.4. OpenDeviceBySN.vi

**Descriptions:**

Open the device by SN

**Parameters:**

IGxFactory	IGXFactory object instance
SN	Serial number of the device
error in	Error in
IGxDevice	IGxDevice object instance
error out	Error out

### 3.5. CloseDevice.vi

**Descriptions:**

Close the device

**Parameters:**

IGxDevice	IGxDevice object instance
error in	Error in
error out	Error out

### 3.6. OpenStream.vi

**Descriptions:**

The user specifies the StreamID to open a stream and get the stream channel object

**Parameters:**

IGxDevice	IGxDevice object instance
StreamID	Stream channel ID
error in	Error in
IGxStream	IGxStream object instance
error out	Error out



### 3.7. CloseStream.vi

**Descriptions:**

Close the stream channel

**Parameters:**

IGxStream	IGxStream object instance
error in	Error in
error out	Error out

### 3.8. IsImplemented.vi

**Descriptions:**

Test whether the feature is implemented

**Parameters:**

IGxDevice	IGxDevice object instance
Feature Name	The name of the feature
error in	Error in
IsImplement	Is it implemented
error out	Error out

### 3.9. GetBoolValue.vi

**Descriptions:**

Get the value of bool

**Parameters:**

IGxDevice	IGxDevice object instance
Feature Name	The name of the feature
error in	Error in
Bool Value	The value of bool
error out	Error out

### 3.10. SetBoolValue.vi

**Descriptions:**

Set the value of bool

**Parameters:**

IGxDevice	IGxDevice object instance
Feature Name	The name of the feature
Bool Value	The value of bool
error in	Error in
error out	Error out

### 3.11. GetIntValue.vi

**Descriptions:**

Get the value of int

**Parameters:**

IGxDevice	IGxDevice object instance
Feature Name	The name of the feature
error in	Error in
Int Value	The value of int
error out	Error out

### 3.12. SetIntValue.vi

**Descriptions:**

Set the value of int

**Parameters:**

IGxDevice	IGxDevice object instance
Feature Name	The name of the feature
Int Value	The value of int
error in	Error in

error out	Error out
-----------	-----------

### 3.13. GetFloatValue.vi

#### Descriptions:

Get the value of float

#### Parameters:

IGxDevice	IGxDevice object instance
Feature Name	The name of the feature
error in	Error in
Float Value	The value of float
error out	Error out

### 3.14. SetFloatValue.vi

#### Descriptions:

Set the value of float

#### Parameters:

IGxDevice	IGxDevice object instance
Feature Name	The name of the feature
Float Value	The value of float
error in	Error in
error out	Error out

### 3.15. GetEnumValue.vi

#### Descriptions:

Get the value of enum

#### Parameters:

IGxDevice	IGxDevice object instance
Feature Name	The name of the feature

error in	Error in
Enum Value	The value of enum
error out	Error out

### 3.16. SetEnumValue.vi

#### Descriptions:

Set the value of enum

#### Parameters:

IGxDevice	IGxDevice object instance
Feature Name	The name of the feature
Enum Value	The value of enum
error in	Error in
error out	Error out

### 3.17. GetStringValue.vi

#### Descriptions:

Get the value of string

#### Parameters:

IGxDevice	IGxDevice object instance
Feature Name	The name of the feature
error in	Error in
String Value	The value of string
error out	Error out

### 3.18. SetStringValue.vi

#### Descriptions:

Set the value of string

#### Parameters:

IGxDevice	IGxDevice object instance
Feature Name	The name of the feature
String Value	The value of string
error in	Error in
error out	Error out

### 3.19. SetCommandValue.vi

#### Descriptions:

Execute the command

#### Parameters:

IGxDevice	IGxDevice object instance
Feature Name	The name of the feature
error in	Error in
error out	Error out

### 3.20. StartGrab.vi

#### Descriptions:

Start the acquisition

#### Parameters:

IGxDevice	IGxDevice object instance
IGxStream	IGxStream object instance
error in	Error in
error out	Error out

### 3.21. StopGrab.vi

#### Descriptions:

Stop the acquisition

#### Parameters:

IGxDevice	IGxDevice object instance
IGxStream	IGxStream object instance
error in	Error in
error out	Error out

### 3.22. GetImage.vi

#### Descriptions:

After the acquisition is started, the image can be directly got through this interface. Note that this interface cannot be used with the callback acquisition method

#### Parameters:

IGxStream	IGxStream object instance
Time out	Timeout, unit: ms
error in	Error in
ImageData	The image data object got
error out	Error out

### 3.23. ImageGrabEvent.vi

#### Descriptions:

In callback acquisition method, callback events are triggered by this interface

#### Parameters:

Event Common Data	Event common data
Callback Param	Image data and so on output by GxIAPINET library
.Control ref	System object
Callback Event	LabVIEW callback event triggered
error out	Error out

### 3.24. GrabImageDisplay.vi

#### Descriptions:

Display the image

**Parameters:**

IBaseData	Object instance of IBaseData image data
Image Src	Allocate memory for saving image data
error in	Error in
Image out	The output image format is IMAQImage
error out	Error out

### 3.25. SaveImage.vi

**Descriptions:**

Save the image to specified path

**Parameters:**

Image Src	IMAQImage image data to be saved
Path	The path for saving
error in	Error in
error out	Error out

## 4. FAQ

Question 1: LabVIEW pops up the searching interface (for .dll) during loading VI.

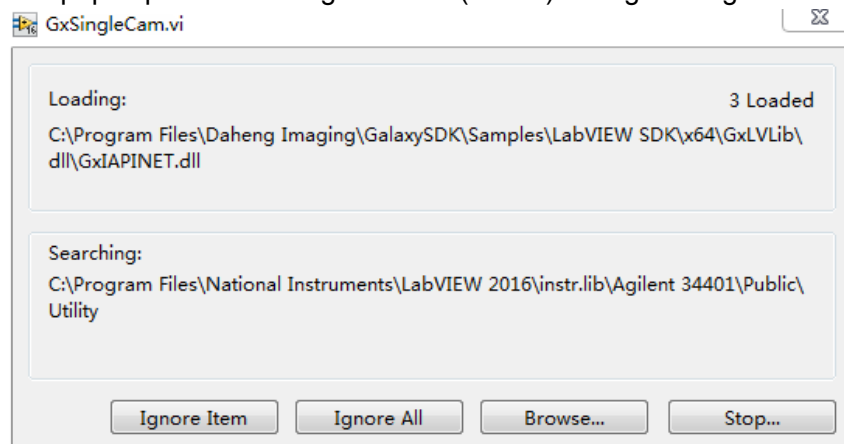


Figure 4-1 Loading VI

Solution:

The GxIAPINET.dll called by the VI will be searched automatically. If it is not found, please click **Browse** manually to add the path of GxIAPINET.dll in the SDK.

Question 2: After browsing and selecting the path of GxIAPINET.dll, it prompts "An error occurred trying to load the assembly".

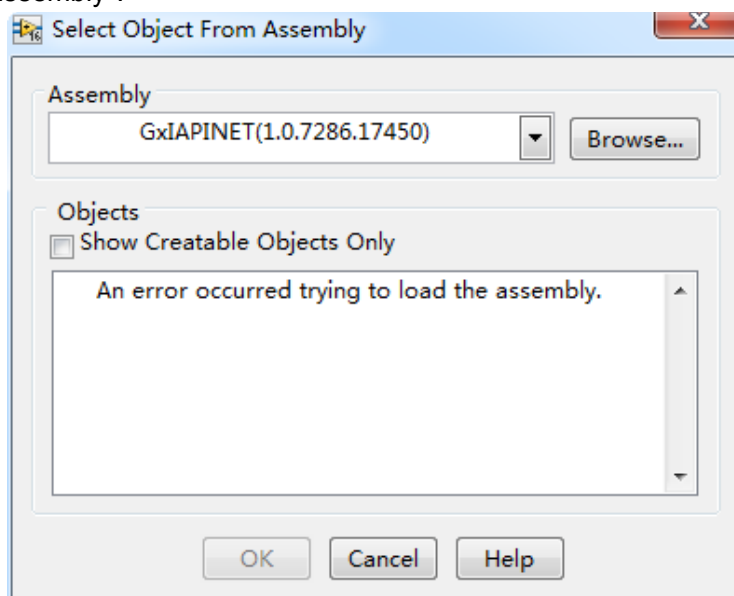


Figure 4-2 An error occurred trying to load the assembly

Solution:

The user may select the 32-bit GxIAPINET library when using 64-bit LabVIEW, or select the 64-bit GxIAPINET library when using 32-bit LabVIEW. When using LabVIEW, please select the appropriate version of the library.



## 5. Revision History

No.	Version	Changes	Date
1	V1.0.0	Initial release	2020-01-19
2	V1.0.1	Modify section 4	2020-05-13