

Technical Note

How to Use Camera Parameters in JAI SDK

How to Use Camera Parameters in SDK:

Abstract:

The JAI SDK works with JAI cameras in the GO series, SP series, SW series, and WA series. However, parameters are not always the same among cameras.

This document describes how to handle camera parameters in developing applications with the JAI SDK.

1. Direct Access Functions:

JAI SDK camera functions work regardless of the camera I/F. To Read/Write/Execute camera parameters, the following functions provide the easiest method of access.

```
J_STATUS_TYPE J_Camera_SetValueInt64(CAM_HANDLE hCam, int8_t* sNodeName, int64_t Value);
```

```
J_STATUS_TYPE J_Camera_SetValueDouble(CAM_HANDLE hCam, int8_t* sNodeName, double Value);
```

```
J_STATUS_TYPE J_Camera_SetValueString(CAM_HANDLE hCam, int8_t* sNodeName, int8_t* ValueStr);
```

```
J_STATUS_TYPE J_Camera_ExecuteCommand(CAM_HANDLE hCam, int8_t* sNodeName);
```

(Detailed descriptions of these functions can be found in "JAI SDK.pdf" which accompanies the software.)

As you can see above, the camera handle "CAM_HANDLE hCam" and string type Node Name "sNodeName" should be specified. The Node Names are defined in the XML file of each camera. Since the Node Names vary among cameras, the SDK manual does not specify them.

The following section describes how to get the Node Name.



Technical Note

How to Use Camera Parameters in JAI SDK

2. Node Name and Type:

Each camera parameter has two names: “GenICam Name (or just simply Name)” and “Display Name”.

When using the JAI Camera Control Tool, camera parameters can be get/set in the “Feature Properties” tab. The parameter shown in the “Feature Properties” tab is the “Display Name”.

On the other hand, when accessing parameters via a user’s program source code, “Name” (or “GenICam Name”) should be used. In the “Feature Tree Information” tab, camera parameters are shown in the same hierarchy as the “Feature Properties” tab. As shown in Figure 1, by selecting a parameter (“Width” for example), detailed information for that parameter is shown in right side.

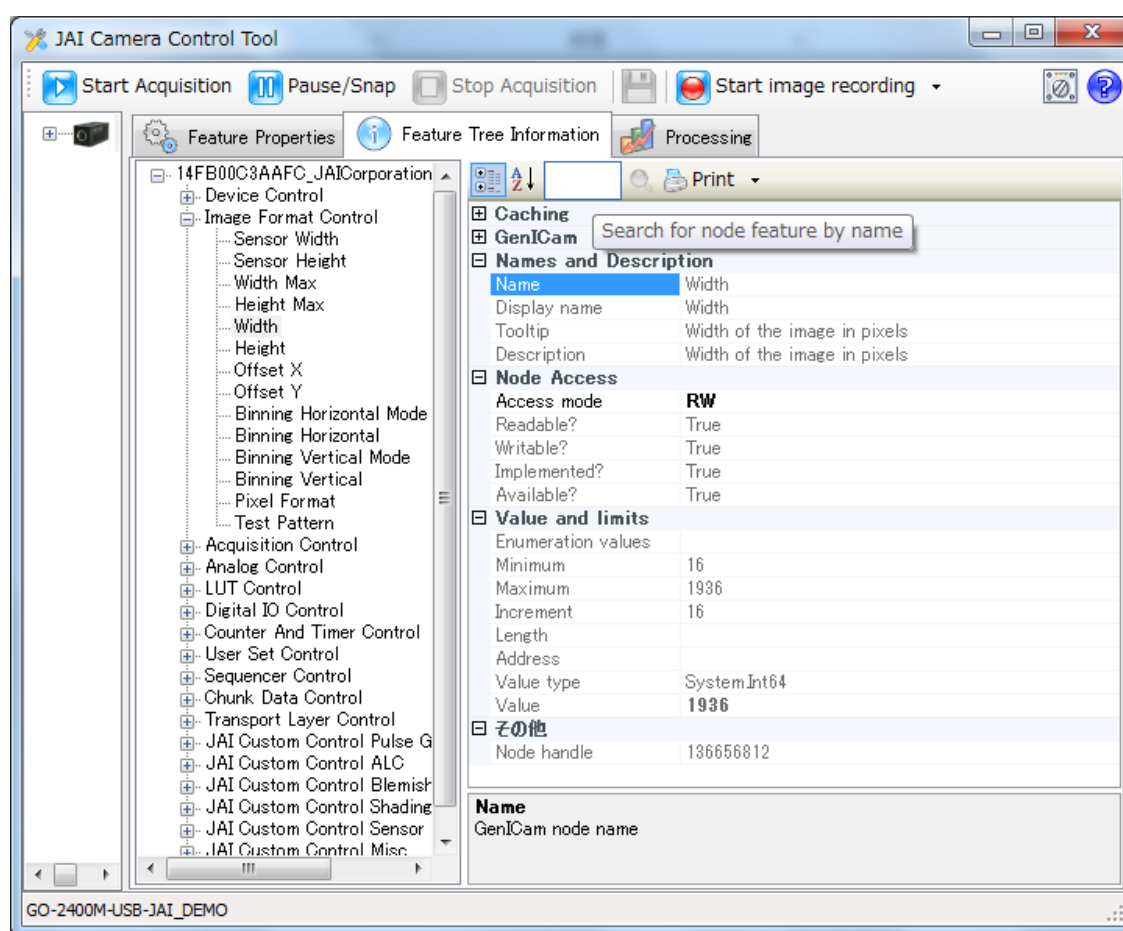


Figure 1

In this example, “Width” appears in the “Name” field. This means that in the user’s source code, “Width” should be used as the sNodeName.



Technical Note

How to Use Camera Parameters in JAI SDK

The “Value type” field in the “Value and limits” category describes the type of the parameter. From this information, the appropriate function can be determined as follows:

2.1 Value type = Int:

In Figure 1, the “Value type” of the parameter “Width” is listed as "System.Int64". Therefore, the following function can be used:

```
J_Camera_SetValueInt64(CAM_HANDLE hCam, int8_t* sNodeName, int64_t Value);
```

2.2 Value type = Double:

In Figure 2 below, “Value type” is "System.Double". Therefore, the following function can be used:

```
J_STATUS_TYPE J_Camera_SetValueDouble(CAM_HANDLE hCam, int8_t* sNodeName, double Value);
```

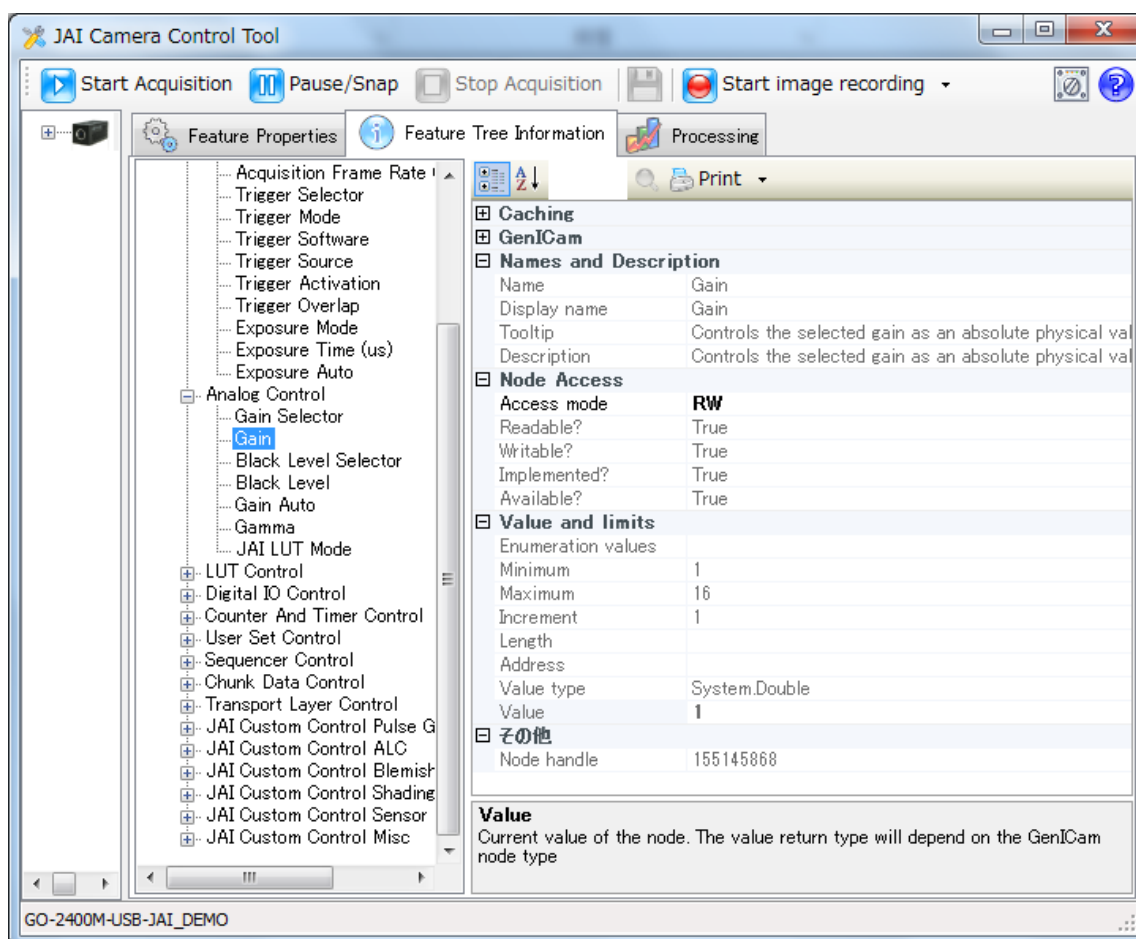


Figure 2



Technical Note

How to Use Camera Parameters in JAI SDK

2.3 Value type = Command:

In Figure 3 below, "Value type" is "System.String". In this case, the Value = "Push to Execute Command". This means the following function can be used.

```
J_STATUS_TYPE J_Camera_ExecuteCommand(CAM_HANDLE hCam, int8_t* sNodeName);
```

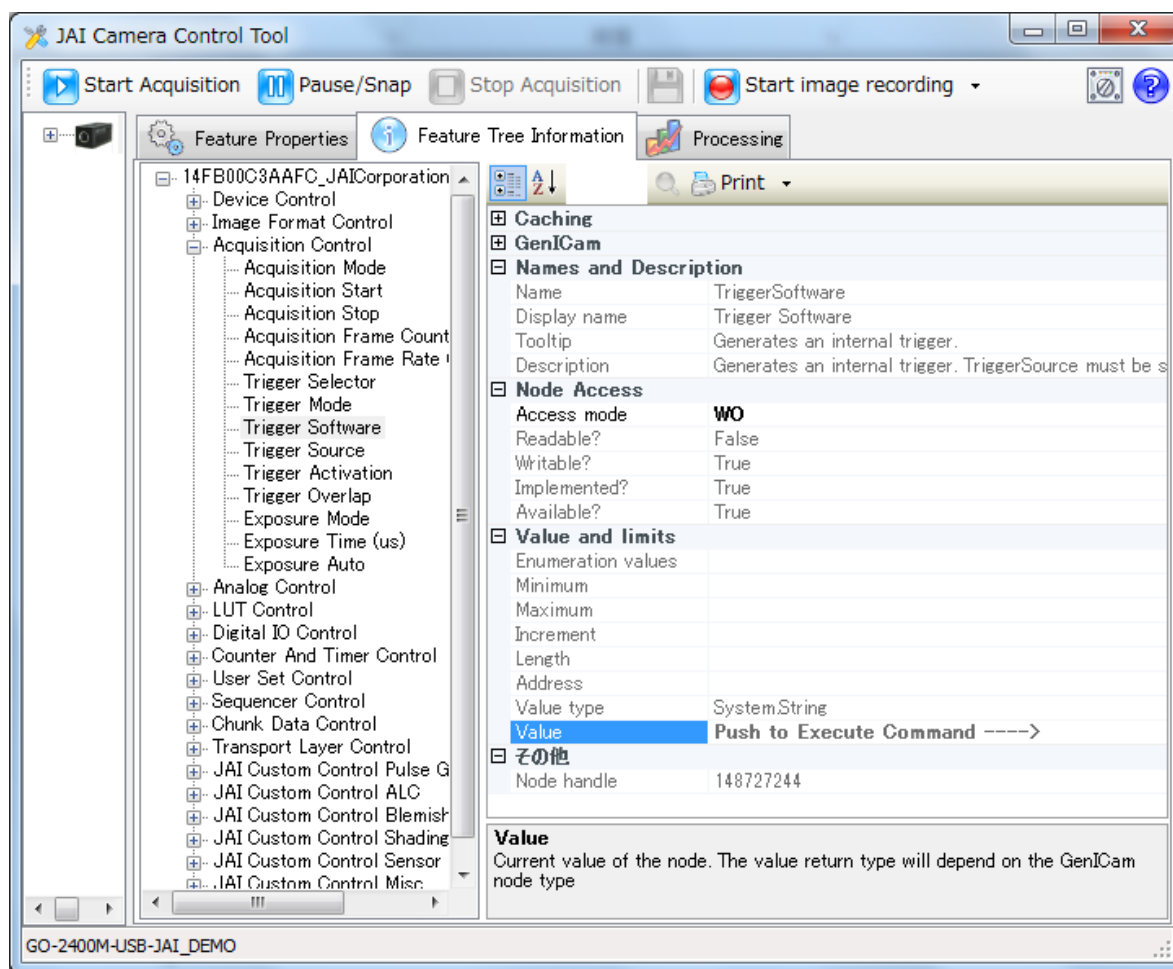


Figure 3



Technical Note

How to Use Camera Parameters in JAI SDK

2.4 Value type = String:

In Figure 4 below, the “Value type” is again “System.String”. Because the Value is something other than “Push to Execute Command”, the following function can be used.

```
J_STATUS_TYPE J_Camera_SetValueString(CAM_HANDLE hCam, int8_t* sNodeName, int8_t* ValueStr);
```

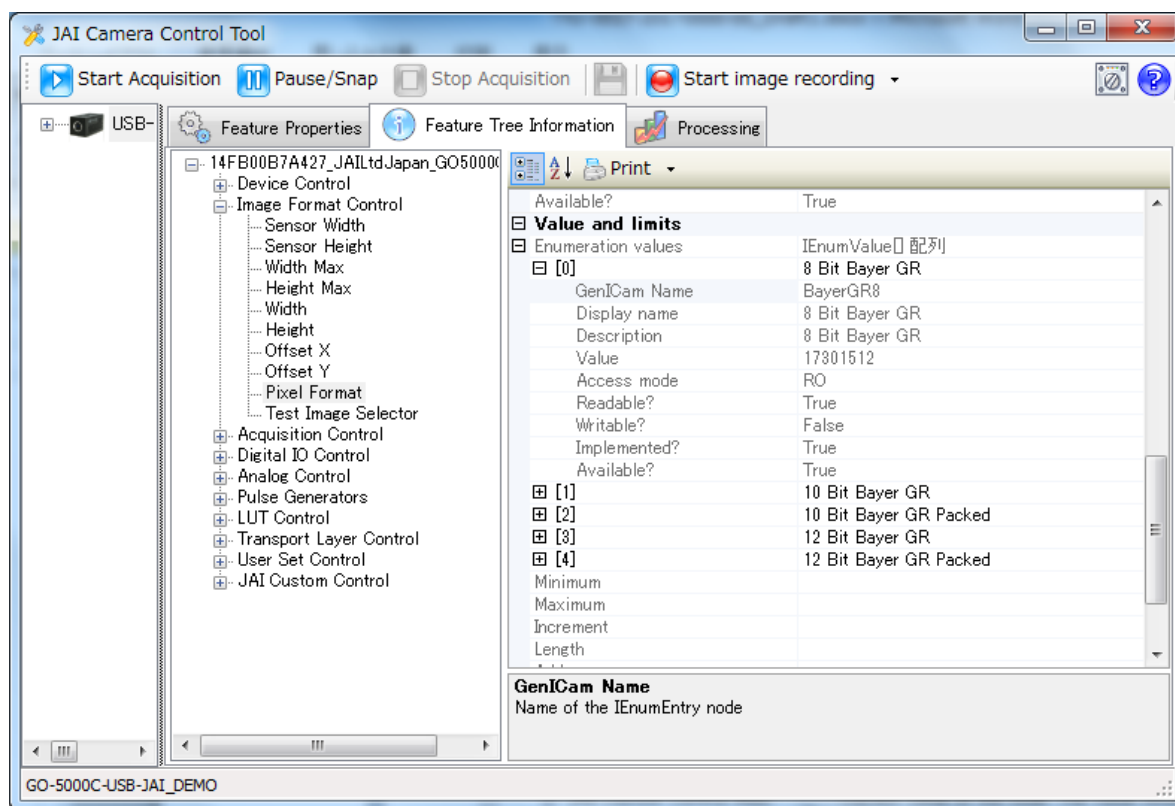


Figure 4

When “Value type” is String, available strings are listed in “Enumeration values” in “Value and limits”. The ValueStr variable (the third parameter of function J_Camera_SetValueString()), can be set to the “GenICam Name” listed in the Enumeration values list. In Figure 4, this would be “BayerGR8”.

3. Handling parameters as Nodes:

In some cases, the range of a parameter could be changed by other parameters. Such parameters should be handled as Nodes.



Technical Note

How to Use Camera Parameters in JAI SDK

In the function below, you can "get" the Node handle by using its GenICam Name (in place of "sNodeName").

```
J_STATUS_TYPE J_Camera_GetNodeByName(CAM_HANDLE hCam, int8_t* sNodeName, NODE_HANDLE *
phNode);
```

By using the above Node handle, the following functions can be used to get related information:

```
J_STATUS_TYPE J_Node_GetValueInt64(NODE_HANDLE hNode, bool Verify, int64_t* pValue); //Value
```

```
J_STATUS_TYPE J_Node_GetMinInt64(NODE_HANDLE hNode, int64_t* pValue); //Min of Int
```

```
J_STATUS_TYPE J_Node_GetMaxInt64(NODE_HANDLE hNode, int64_t* pValue); //Max of Int
```

```
J_STATUS_TYPE J_Node_GetInc(NODE_HANDLE hNode, int64_t *pValue); //Step of Int
```

```
J_STATUS_TYPE J_Node_GetMinDouble(NODE_HANDLE hNode, double* pValue); //Min of Double
```

```
J_STATUS_TYPE J_Node_GetMaxDouble(NODE_HANDLE hNode, double* pValue); //Max of Double
```

...

Setting values should be in range of Max and Min. You can refer to section 1 for how to set values.

End.



