Writing Applications with xiAPI

Default parameters

After camera is opened by xiOpenDevice the default camera parameters are set by API. The default parameters might be different in different API versions. In order to ensure that your application will have camera in expected state with any API version - please set all parameters expected by your application to required value.

Extended Device parameters

XI_PRM_IS_DEVICE_EXIST or "isexist"

Description: Returns 1 if camera connected and works properly.

Type: Integer.

Default value: XI_OFF

Usage:

```
int value = 0;
xiGetParamInt(handle, XI_PRM_IS_DEVICE_EXIST, &value);
```

XI_PRM_ACQ_BUFFER_SIZE or "acq_buffer_size"

Description: Defines the size of the acquisition buffer in bytes(see Image below). This is a circle buffer which contains image data from sensor. This parameter can be set only when acquisition is stopped.

Note1: If the processing of this image takes more time than these 7seconds, the image data will be automatically overwritten with new image data due to the circular character of the buffer.

Note2: The maximal value for this parameter is 2147483647 because it uses the signed integer.

Type: Integer.

Default value: 100000000

Typical range: [0, 2147483647]

Usage:

```
int value = 0;
xiGetParamInt(handle, XI_PRM_ACQ_BUFFER_SIZE, &value);
xiSetParamInt(handle, XI_PRM_ACQ_BUFFER_SIZE, value);
```

Example: Sensor gives 1MB of data per image @ 10 frames/second, Application retrieves image data by xiGetImage, Application has access to the frame at most 7 seconds (70MB/(1MB*10fps))

```
xiSetParamInt(handle, XI_PRM_ACQ_BUFFER_SIZE, 70*1000*1000);
```

XI_PRM_ACQ_BUFFER_SIZE_UNIT or "acq_buffer_size_unit"

Description: Acquisition buffer size unit. Default 1. E.g. Value 1024 means that buffer_size is in KiBytes.

Type: Integer.

Default value: 1

Typical range: [1, 2147483647]

Usage:

```
// set unit to 1 MiB
xiSetParamInt(handle, XI_PRM_ACQ_BUFFER_SIZE_UNIT, 1024*1024);
// set buffer size to 200 MiB
xiSetParamInt(handle, XI_PRM_ACQ_BUFFER_SIZE, 200);
```

XI_PRM_ACQ_TRANSPORT_BUFFER_SIZE or "acq_transport_buffer_size"

Description: Size of one transport buffer in bytes (only valid for MQ,MD camera families). Frame/Field can contain multiple transport buffers. To decrease CPU load and increase system performance on committing transport buffers to kernel driver, transport buffer size has to be as high as possible. However in case of small Frame/Field size and high framerates it is necessary to decrease transport buffer size and increase queue of Frame/Field buffers (XI_PRM_BUFFERS_QUEUE_SIZE). Check out How to optimize software performance on high frame rates for more info.

Note: Whole range minimum to maximum is not guaranteed on all tested configurations. Please be aware of possible issues on some controllers.

Type: Integer.

Default value: 0

Usage:

```
xiSetParamInt(handle, XI_PRM_ACQ_TRANSPORT_BUFFER_SIZE, size_in_bytes);
```

Example: Application set transport buffer size to 128KiB for small (80KB images)

```
xiSetParamInt(handle, XI_PRM_ACQ_TRANSPORT_BUFFER_SIZE, 128*1024);
```

XI_PRM_ACQ_TRANSPORT_PACKET_SIZE or "acq_transport_packet_size"

Description: Acquisition transport packet size in bytes. (only valid for MQ,MD camera families)

Type: Integer.

Default value: 0

Usage:

```
// Get packet size
xiGetParamInt(handle, XI_PRM_ACQ_TRANSPORT_PACKET_SIZE, &packet_size);
```

XI_PRM_BUFFERS_QUEUE_SIZE or "buffers_queue_size"

Description: XI_PRM_BUFFERS_QUEUE_SIZE - 1 is the maximum number of images which can be stored in the buffers queue.

Type: Integer.

Default value: 4

Typical range: [2, 2147483647]

Is invalidated by: XI_PRM_ACQ_BUFFER_SIZE

Usage:

```
xiSetParamInt(handle, XI_PRM_BUFFERS_QUEUE_SIZE, images_count);
```

Example: Application sets queue size to 7

Sensor acquires images at 10 frames/second. It means new image is acquired every 100ms, therefore queue can contain up to 6 images acquired during 600ms (=6*100ms) time span.

Application needs typically 40ms for processing of an image

If it takes 400ms for the application to process one of the images (e.g. saving it to a server) the subsequent 4 frames are still available since they are stored in the queue (images 17-19 on the below figure).

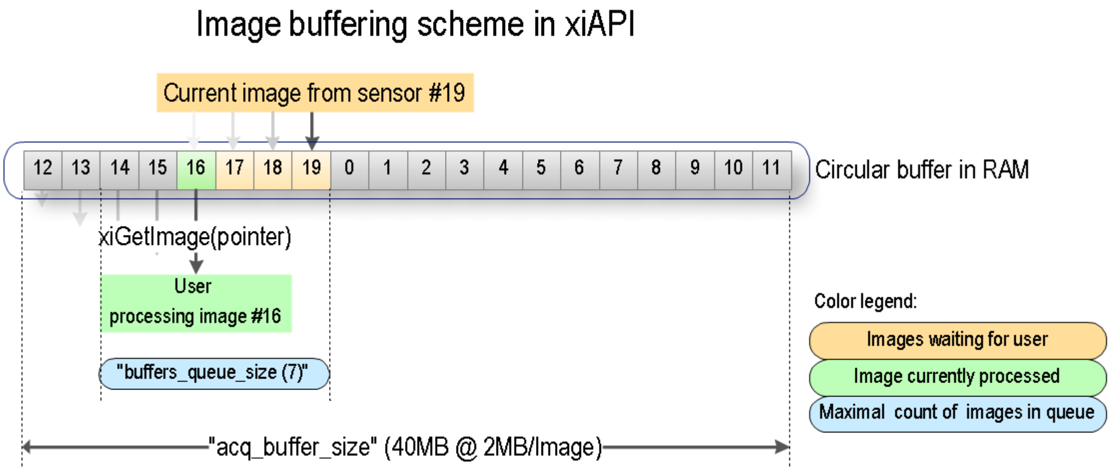


Image buffering scheme in xiAPI
Current image from sensor #19
12 13 14 15 16 17 18 19 0 1 2 3 4 5 6 7 8 9 10 11
Circular buffer in RAM
xiGetImage(pointer)
User processing image #16
"buffers_queue_size (7)"
Color legend:
Images waiting for user
Image currently processed
Maximal count of images in queue
"acq_buffer_size" (40MB @ 2MB/Image)

Illustration of normal status when application is processing one frame longer

If it takes 1000ms for the application to process one of the images, the buffer queue will fill up and some images will be skipped (images 16-20 on the below figure). The user can check for such situation by checking the image sequence number (nframe of XI_IMG structure) of each image.

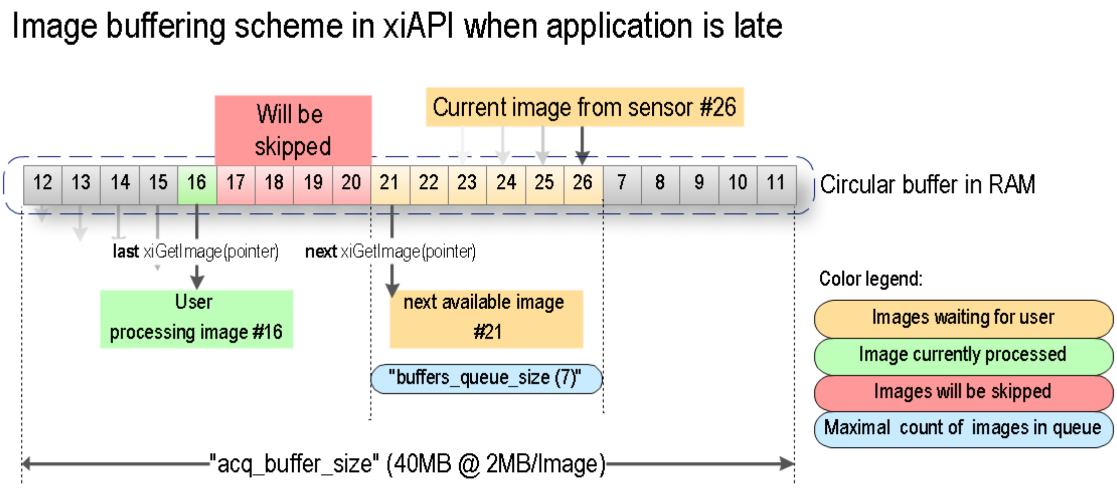


Illustration of state when application gets late. Longer than queue.

The application can work with the image data at most acq_buffer_size/(frame_rate_FPS*bytes_per_frame) seconds before the data is overwritten due to the circular character of the buffer. If the application needs more time to process the image, XI_PRM_BUFFER_POLICY
must be set to XI_BP_SAFE. In this case API copies the image to a user/API allocated memory where it can be accessed without the risk of being overwritten. This copying however takes extra CPU time.

XI_PRM_ACQ_TRANSPORT_BUFFER_COMMIT or "acq_transport_buffer_commit"

Description: Defines number of buffers to be committed to transport layer. (only valid for USB 3.0 camera families)

Type: Integer.

Default value: 1

Typical range: [1, 256]

Usage:

```
int value = 0;
xiGetParamInt(handle, XI_PRM_ACQ_TRANSPORT_BUFFER_COMMIT, &value);
xiSetParamInt(handle, XI_PRM_ACQ_TRANSPORT_BUFFER_COMMIT, value);
```

XI_PRM_RECENT_FRAME or "recent_frame"

Description: This parameter changes the behavior of xiGetImage.

Note1: possible value: 0 - Retrieves next available image from buffer

Note2: possible value: 1 - Retrieves the most recent image from buffer

Type: Integer.

Default value: XI_OFF

Usage:

```
int value = 0;
xiGetParamInt(handle, XI_PRM_RECENT_FRAME, &value);
xiSetParamInt(handle, XI_PRM_RECENT_FRAME, XI_ON);
```

XI_PRM_DEVICE_RESET or "device_reset"

Description: Resets the camera firmware. From the functional view, it is the same as disconnection and connection of the camera.It is typically followed by an enumeration of the operating system which might take some time (e.g. 10 seconds). Application shall wait some time after the reset and then use xiGetNumberDevices in order to enumerate the camera again. It should be used on the device with the stopped acquisition. After re-enumeration closing of the old device to release its handler is recommended. A new handler should be used for further image acquisition.

Note: currently supported only for xiQ camera family

Type: Integer.

Default value: 0

Usage:

```
xiSetParamInt(old_handle, XI_PRM_DEVICE_RESET, 1);
Sleep(10000); // wait 10 seconds for enumeration by OS
DWORD devices_count = 0;
xiGetNumberDevices(&devices_count);
xiCloseDevice(old_handle);
HANDLE new_handle = NULL;
xiOpenDevice(0, &new_handle);
```

XI_PRM_CONCAT_IMG_MODE or "concat_img_mode"

Description: Enable/disable the Concatenated Images in One Buffer feature

Type: Integer.

Default value: XI_OFF

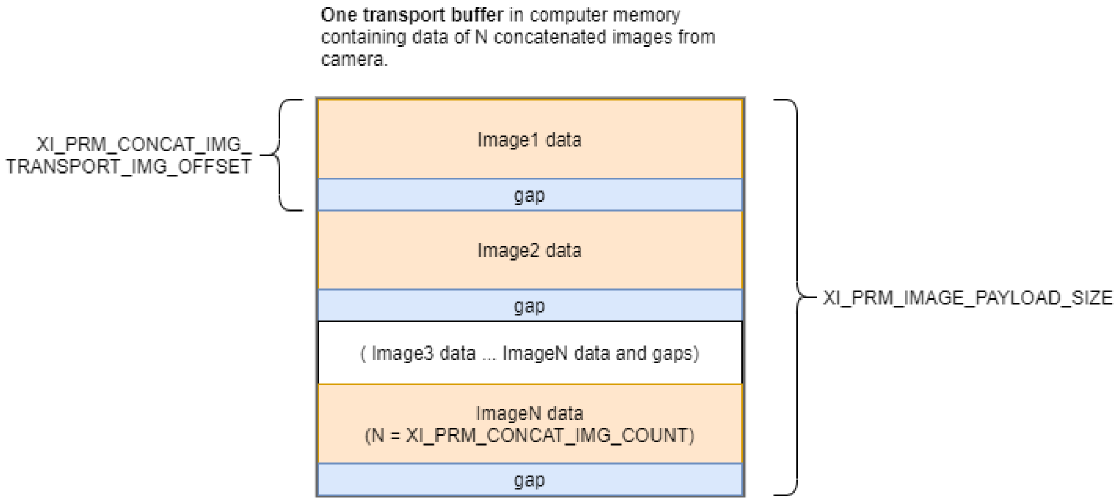
Usage:

```
int value = 0;
xiGetParamInt(handle, XI_PRM_CONCAT_IMG_MODE, &value);
xiSetParamInt(handle, XI_PRM_CONCAT_IMG_MODE, XI_ON);
```

XI_PRM_CONCAT_IMG_COUNT or "concat_img_count"

Description: Number of Concatenated Images in One Buffer.

Note: Read more at Concatenated Images in One Buffer feature.


One transport buffer in computer memory containing data of N concatenated images from camera.
XI_PRM_CONCAT_IMG_TRANSPORT_IMG_OFFSET
Image1 data
gap
Image2 data
gap
(Image3 data ... ImageN data and gaps)
ImageN data
(N = XI_PRM_CONCAT_IMG_COUNT)
gap
XI_PRM_IMAGE_PAYLOAD_SIZE

Type: Integer.

Default value: 1

Typical range: [1, 0]

Usage:

```
int value = 0;
xiGetParamInt(handle, XI_PRM_CONCAT_IMG_COUNT, &value);
xiSetParamInt(handle, XI_PRM_CONCAT_IMG_COUNT, value);
```

XI_PRM_CONCAT_IMG_TRANSPORT_IMG_OFFSET or "concat_img_transport_img_offset"

Description: Offset between images data in transport buffer when feature Concatenated Images in One Buffer is enabled

Type: Integer.

Default value: 1

Usage:

```
int value = 0;
xiGetParamInt(handle, XI_PRM_CONCAT_IMG_TRANSPORT_IMG_OFFSET, &value);
```

XI_PRM_PROBE_SELECTOR or "probe_selector"

Description: Select Probe

Type: Enumerator.

Default value: XI_PROBE_SELECTOR_CURRENT_MAINBOARD_VCC_IN

Usage:

```
int probe_selector = 0;
xiGetParamInt(handle, XI_PRM_PROBE_SELECTOR, &probe_selector);
xiSetParamInt(handle, XI_PRM_PROBE_SELECTOR,
XI_PROBE_SELECTOR_CURRENT_MAINBOARD_VCC_IN);
```

Value	Description
XI_PROBE_SELECTOR_CURRENT_MAINBOARD_VCC_IN	Current probe on Main Board at VCC_IN power supply
XI_PROBE_SELECTOR_VOLTAGE_MAINBOARD_VCC_IN	Voltage probe on Main Board at VCC_IN power supply
XI_PROBE_SELECTOR_CURRENT_MAINBOARD_VCC_ADJ2	Current probe on Main Board at VCC_ADJ2 power supply
XI_PROBE_SELECTOR_VOLTAGE_MAINBOARD_VCC_ADJ2	Voltage probe on Main Board at VCC_ADJ2 power supply
XI_PROBE_SELECTOR_CURRENT_MAINBOARD_VCC_ADJ1	Current probe on Main Board at VCC_ADJ1 power supply
XI_PROBE_SELECTOR_VOLTAGE_MAINBOARD_VCC_ADJ1	Voltage probe on Main Board at VCC_ADJ1 power supply
XI_PROBE_SELECTOR_CURRENT_MAINBOARD_VCC_PLT	Current probe on Main Board at VCC_PLT power supply
XI_PROBE_SELECTOR_VOLTAGE_MAINBOARD_VCC_PLT	Voltage probe on Main Board at VCC_PLT power supply
XI_PROBE_SELECTOR_VOLTAGE_SENSORBOARD_VCC_ADJ1	Voltage probe on Sensor Board at VCC_ADJ1
XI_PROBE_SELECTOR_VOLTAGE_SENSORBOARD_VCC_ADJ2	Voltage probe on Sensor Board at VCC_ADJ2
XI_PROBE_SELECTOR_VOLTAGE_SENSORBOARD_VCC_5V0	Voltage probe on Sensor Board at VCC_5V0
XI_PROBE_SELECTOR_VOLTAGE_SENSORBOARD_VCC_3V3	Voltage probe on Sensor Board at VCC_3V3
XI_PROBE_SELECTOR_VOLTAGE_DATA_CON_INPUT	Voltage probe on device input, if device is bus powered
XI_PROBE_SELECTOR_VOLTAGE_PELTIER1	Voltage probe on peltier #1
XI_PROBE_SELECTOR_CURRENT_PELTIER1	Current probe on peltier #1
XI_PROBE_SELECTOR_VOLTAGE_PELTIER2	Voltage probe on peltier #2
XI_PROBE_SELECTOR_CURRENT_PELTIER2	Current probe on peltier #2

XI_PRM_PROBE_VALUE or "probe_value"

Description: Returns Value of the selected Probe

Type: Float.

Default value: 0.0

Usage:

```
float value = 0.0;
xiGetParamFloat(handle, XI_PRM_PROBE_VALUE, &value);
```

Sensor Defects Correction

XI_PRM_COLUMN_FPN_CORRECTION or "column_fpn_correction"

Description: Correction of column fpn.

Type: Enumerator.

Default value: XI_OFF

Usage:

```
int column_fpn_correction = 0;
xiGetParamInt(handle, XI_PRM_COLUMN_FPN_CORRECTION, &column_fpn_correction);
xiSetParamInt(handle, XI_PRM_COLUMN_FPN_CORRECTION, XI_OFF);
```

Value	Description
XI_OFF	Turn parameter off
XI_ON	Turn parameter on

XI_PRM_ROW_FPN_CORRECTION or "row_fpn_correction"

Description: Correction of row fpn.

Type: Enumerator.

Default value: XI_OFF

Usage:

```
int row_fpn_correction = 0;
xiGetParamInt(handle, XI_PRM_ROW_FPN_CORRECTION, &row_fpn_correction);
xiSetParamInt(handle, XI_PRM_ROW_FPN_CORRECTION, XI_OFF);
```

Value	Description
XI_OFF	Turn parameter off
XI_ON	Turn parameter on

XI_PRM_COLUMN_BLACK_OFFSET_CORRECTION or "column_black_offset_correction"

Description: Correction of column black offset.

Type: Enumerator.

Default value: XI_OFF

Usage:

```
int column_black_offset_correction = 0;

xiGetParamInt(handle, XI_PRM_COLUMN_BLACK_OFFSET_CORRECTION, &column_black_offset_correction);

xiSetParamInt(handle, XI_PRM_COLUMN_BLACK_OFFSET_CORRECTION, XI_OFF);
```

Value	Description
XI_OFF	Turn parameter off
XI_ON	Turn parameter on

XI_PRM_ROW_BLACK_OFFSET_CORRECTION or "row_black_offset_correction"

Description: Correction of row black offset.

Type: Enumerator.

Default value: XI_OFF

Usage:

```
int row_black_offset_correction = 0;

xiGetParamInt(handle, XI_PRM_ROW_BLACK_OFFSET_CORRECTION, &row_black_offset_correction);

xiSetParamInt(handle, XI_PRM_ROW_BLACK_OFFSET_CORRECTION, XI_OFF);
```

Value	Description
XI_OFF	Turn parameter off
XI_ON	Turn parameter on

Sensor features

XI_PRM_SENSOR_MODE or "sensor_mode"

Description: Current sensor mode. Allows to select sensor mode by one integer. Setting of this parameter affects: image dimensions and downsampling.

Type: Enumerator.

Default value: 0

Usage:

```
int sensor_mode = 0;
xiGetParamInt(handle, XI_PRM_SENSOR_MODE, &sensor_mode);
xiSetParamInt(handle, XI_PRM_SENSOR_MODE, XI_SENS_MD0);
```

Value	Description

XI_SENS_MD0	Sensor mode number 0
XI_SENS_MD1	Sensor mode number 1
XI_SENS_MD2	Sensor mode number 2
XI_SENS_MD3	Sensor mode number 3
XI_SENS_MD4	Sensor mode number 4
XI_SENS_MD5	Sensor mode number 5
XI_SENS_MD6	Sensor mode number 6
XI_SENS_MD7	Sensor mode number 7
XI_SENS_MD8	Sensor mode number 8
XI_SENS_MD9	Sensor mode number 9
XI_SENS_MD10	Sensor mode number 10
XI_SENS_MD11	Sensor mode number 11
XI_SENS_MD12	Sensor mode number 12
XI_SENS_MD13	Sensor mode number 13
XI_SENS_MD14	Sensor mode number 14
XI_SENS_MD15	Sensor mode number 15

XI_PRM_HDR or "hdr"

Description: Enable High Dynamic Range sensor feature.

Note1: enables HDR mode for certain type of sensors. For more information see HDR mode support page.

Type: Integer.

Default value: XI_OFF

Usage:

```
int value = 0;
xiGetParamInt(handle, XI_PRM_HDR, &value);
xiSetParamInt(handle, XI_PRM_HDR, XI_ON);
```

XI_PRM_HDR_KNEEPOINT_COUNT or "hdr_kneepoint_count"

Description: number of kneepoints.

Note1: Defines the number of kneepoints in the Piecewise Linear Response (PWLR) curve.

Note2: In case of one kneepoint, the kneepoint is defined by parameters (T2,SL2). In case of two kneepoints define both (T1,SL1), (T2,SL2).

Type: Integer.

Default value: 1

Typical range: [1, 2]

Usage:

```
int value = 0;
xiGetParamInt(handle, XI_PRM_HDR_KNEEPOINT_COUNT, &value);
xiSetParamInt(handle, XI_PRM_HDR_KNEEPOINT_COUNT, value);
```

XI_PRM_HDR_T1 or "hdr_t1"

Description: Exposure time (T1) of 1st kneepoint in % of XI_PRM_EXPOSURE

Type: Integer.

Default value: 60

Typical range: [0, 100]

Usage:

```
int value = 0;
xiGetParamInt(handle, XI_PRM_HDR_T1, &value);
xiSetParamInt(handle, XI_PRM_HDR_T1, value);
```

XI_PRM_HDR_T2 or "hdr_t2"

Description: Exposure time (T2) of 2nd kneepoint in % of

Type: Integer.

Default value: 80

Typical range: [0, 100]

Usage:

```
int value = 0;
xiGetParamInt(handle, XI_PRM_HDR_T2, &value);
xiSetParamInt(handle, XI_PRM_HDR_T2, value);
```

XI_PRM_KNEEPOINT1 or "hdr_kneepoint1"

Description: Saturation level (SL1) of 1st kneepoint in % of sensor saturation.

Type: Integer.

Default value: 40

Typical range: [0, 100]

Usage:

```
int value = 0;
xiGetParamInt(handle, XI_PRM_KNEEPOINT1, &value);
xiSetParamInt(handle, XI_PRM_KNEEPOINT1, value);
```

XI_PRM_KNEEPOINT2 or "hdr_kneepoint2"

Description: Saturation level (SL2) of 2nd kneepoint in % of sensor saturation.

Type: Integer.

Default value: 60

Typical range: [0, 100]

Usage:

```
int value = 0;
xiGetParamInt(handle, XI_PRM_KNEEPOINT2, &value);
xiSetParamInt(handle, XI_PRM_KNEEPOINT2, value);
```

XI_PRM_IMAGE_BLACK_LEVEL or "image_black_level"

Description: Black level is calculated level (in pixel counts) that should reflect the value of pixels without light. It should be the same as XI_IMG.black_level from last image get using xiGetImage. Setting of this parameter does not affect the data from sensor or API when camera is connected. It can be used for setting black level only for Offline Processing.

Type: Integer.

Default value: 0

Usage:

```
int value = 0;
xiGetParamInt(handle, XI_PRM_IMAGE_BLACK_LEVEL, &value);
```

XI_PRM_IMAGE_AREA or "image_area"

Description: Defines image area of sensor as output.

Type: Enumerator.

Default value: XI_IMAGE_AREA_ACTIVE

Usage:

```
int image_area = 0;
xiGetParamInt(handle, XI_PRM_IMAGE_AREA, &image_area);
xiSetParamInt(handle, XI_PRM_IMAGE_AREA, XI_IMAGE_AREA_ACTIVE);
```

Value	Description
XI_IMAGE_AREA_ACTIVE	All light sensitive pixels suggested by image vendor.
XI_IMAGE_AREA_ACTIVE_AND_MASKED	All Active pixels plus masked pixels surrounding the Active area.

XI_PRM_DUAL_ADC_MODE or "dual_adc_mode"

Description: Sets DualADC Mode

Type: Enumerator.

Default value: XI_DUAL_ADC_MODE_OFF

Usage:

```
int dual_adc_mode = 0;
xiGetParamInt(handle, XI_PRM_DUAL_ADC_MODE, &dual_adc_mode);
xiSetParamInt(handle, XI_PRM_DUAL_ADC_MODE, XI_DUAL_ADC_MODE_OFF);
```

Value	Description
XI_DUAL_ADC_MODE_OFF	Disable DualADC feature
XI_DUAL_ADC_MODE_COMBINED	Set Combined mode
XI_DUAL_ADC_MODE_NON_COMBINED	Set NonCombined mode

XI_PRM_DUAL_ADC_GAIN_RATIO or "dual_adc_gain_ratio"

Description: Sets DualADC Gain Ratio in dB

Type: Float.

Default value: 0.0

Typical range: [0.0, 24.0]

Usage:

```
float value = 0.0;
xiGetParamFloat(handle, XI_PRM_DUAL_ADC_GAIN_RATIO, &value);
xiSetParamFloat(handle, XI_PRM_DUAL_ADC_GAIN_RATIO, value);
```

RESHOLD or "dual_adc_threshold"

Description: Sets DualADC Threshold value

Type: Integer.

Default value: 50

Typical range: [0, 100]

Is invalidated by: XI_PRM_DUAL_ADC_MODE

Usage:

```
int value = 0;
xiGetParamInt(handle, XI_PRM_DUAL_ADC_THRESHOLD, &value);
xiSetParamInt(handle, XI_PRM_DUAL_ADC_THRESHOLD, value);
```

XI_PRM_COMPRESSION_REGION_SELECTOR or "compression_region_selector"

Description: Sets Compression Region Selector

Type: Integer.

Default value: 1

Typical range: [1, 2]

Is invalidated by: XI_PRM_DUAL_ADC_MODE

Usage:

```
int value = 0;
xiGetParamInt(handle, XI_PRM_COMPRESSION_REGION_SELECTOR, &value);
xiSetParamInt(handle, XI_PRM_COMPRESSION_REGION_SELECTOR, value);
```

XI_PRM_COMPRESSION_REGION_START or "compression_region_start"

Description: Sets Compression Region Start

Type: Float.

Default value: 0.0

Typical range: [0.0, 50.0]

Is invalidated by: XI_PRM_DUAL_ADC_MODE, XI_PRM_DUAL_ADC_GAIN_RATIO, XI_PRM_COMPRESSION_REGION_SELECTOR

Usage:

```
float value = 0.0;
xiGetParamFloat(handle, XI_PRM_COMPRESSION_REGION_START, &value);
xiSetParamFloat(handle, XI_PRM_COMPRESSION_REGION_START, value);
```

XI_PRM_COMPRESSION_REGION_GAIN or "compression_region_gain"

Description: Sets Compression Region Gain

Type: Float.

Default value: 0.0

Typical range: [-90.0, 0.0]

Is invalidated by: XI_PRM_DUAL_ADC_MODE

Usage:

```
float value = 0.0;
xiGetParamFloat(handle, XI_PRM_COMPRESSION_REGION_GAIN, &value);
xiSetParamFloat(handle, XI_PRM_COMPRESSION_REGION_GAIN, value);
```