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.

Version info

XI_PRM_VERSION_SELECTOR or "version_selector"

Description: Selects module/unit, which version we get.

Type: Enumerator.

Default value: 0

Usage:

```
int version_selector = 0;
xiGetParamInt(handle, XI_PRM_VERSION_SELECTOR, &version_selector);
xiSetParamInt(handle, XI_PRM_VERSION_SELECTOR, XI_VER_API);
```

Value	Description
XI_VER_API	version of API
XI_VER_DRV	version of device driver
XI_VER_MCU1	version of MCU1 firmware.
XI_VER_MCU2	version of MCU2 firmware.
XI_VER_MCU3	version of MCU3 firmware.
XI_VER_FPGA1	version of FPGA1 firmware.
XI_VER_XMLMAN	version of XML manifest.
XI_VER_HW_REV	version of hardware revision.
XI_VER_FACTORY_SET	version of factory set.

XI_PRM_VERSION or "version"

Description: Returns version of selected module/unit(XI_PRM_VERSION_SELECTOR).

Type: String.

Default value: -

Usage:

```
char value[200] = "";
xiGetParamString(handle, XI_PRM_VERSION, &value, sizeof(value));
```

XI_PRM_API_VERSION or "api_version"

Description: Returns the version of API.

Type: String.

Default value: -

Usage:

```
char value[200] = "";
xiGetParamString(handle, XI_PRM_API_VERSION, &value, sizeof(value));
```

XI_PRM_DRV_VERSION or "drv_version"

Description: Returns the version of the current device driver.

Type: String.

Default value: -

Usage:

```
char value[200] = "";
xiGetParamString(handle, XI_PRM_DRV_VERSION, &value, sizeof(value));
```

XI_PRM_MCU1_VERSION or "version_mcu1"

Description: Returns the version of the current MCU1 firmware.

Type: String.

Default value: -

Usage:

```
char value[200] = "";
xiGetParamString(handle, XI_PRM_MCU1_VERSION, &value, sizeof(value));
```

XI_PRM_MCU2_VERSION or "version_mcu2"

Description: Returns the version of the current MCU2 firmware.

Type: String.

Default value: -

Usage:

```
char value[200] = "";
xiGetParamString(handle, XI_PRM_MCU2_VERSION, &value, sizeof(value));
```

XI_PRM_MCU3_VERSION or "version_mcu3"

Description: Returns the version of the current MCU3 firmware.

Type: String.

Default value: -

Usage:

```
char value[200] = "";
xiGetParamString(handle, XI_PRM_MCU3_VERSION, &value, sizeof(value));
```

XI_PRM_FPGA1_VERSION or "version_fpga1"

Description: Returns version of FPGA firmware currently running.

Type: String.

Default value: -

Usage:

```
char value[200] = "";
xiGetParamString(handle, XI_PRM_FPGA1_VERSION, &value, sizeof(value));
```

XI_PRM_XMLMAN_VERSION or "version_xmlman"

Description: Returns version of XML manifest.

Type: String.

Default value: -

Usage:

```
char value[200] = "";
xiGetParamString(handle, XI_PRM_XMLMAN_VERSION, &value, sizeof(value));
```

XI_PRM_HW_REVISION or "hw_revision"

Description: Returns the hardware revision number of the camera.

Type: String.

Default value: -

Usage:

```
char value[200] = "";
xiGetParamString(handle, XI_PRM_HW_REVISION, &value, sizeof(value));
```

XI_PRM_FACTORY_SET_VERSION or "factory_set_version"

Description: Returns version of factory set.

Type: String.

Default value: -

Usage:

```
char value[200] = "";
xiGetParamString(handle, XI_PRM_FACTORY_SET_VERSION, &value, sizeof(value));
```

API features

XI_PRM_DEBUG_LEVEL or "debug_level"

Description: Setting the API debug level allows to select amount of messages stored to debug output.

Type: Enumerator.

Default value: XI_DL_WARNING

Usage:

```
int debug_level = 0;
xiGetParamInt(handle, XI_PRM_DEBUG_LEVEL, &debug_level);
```

```
xiSetParamInt(handle, XI_PRM_DEBUG_LEVEL, XI_DL_DETAIL);
```

Value	Description
XI_DL_DETAIL	(see Note1)
XI_DL_TRACE	Prints errors, warnings and important informations
XI_DL_WARNING	Prints all errors and warnings
XI_DL_ERROR	Prints all errors
XI_DL_FATAL	Prints only important errors
XI_DL_DISABLED	Prints no messages

Note1: Prints same as XI_DL_TRACE plus locking of resources.

Note2: In Windows use DebugView to view the current messages.

Note3: In Linux the messages are printed to stderr

XI_PRM_AUTO_BANDWIDTH_CALCULATION or "auto_bandwidth_calculation"

Description: Setting this parameter the application can control API behavior. Setting to XI_OFF - API will skip auto bandwidth measurement and calculation before opening the camera (xiOpenDevice), resulting in reducing the time to open a camera. Setting to XI_ON the measurement is enabled (default).

Note1: It is important to set this parameter to XI_OFF in case when multiple cameras are connected to one hub with enabled acquisition and new camera should be opened - to not affect overall streaming by auto bandwidth measurement.

Note2: When set to value XI_OFF, the time required to open a camera (xiOpenDevice) is reduced.

Type: Integer.

Default value: XI_ON

Usage:

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

XI_PRM_NEW_PROCESS_CHAIN_ENABLE or "new_process_chain_enable"

Description: Setting this parameter the application can control API behavior. When set to XI_OFF - API will use original processing in image pipe for cameras families MU, MQ, MD. Setting to XI_ON - API will use newer processing type.

Note: There are some differences between processing so the switching may be done with caution. For older implementation we advise to stick to original processing. Only if some features require the newer processing it might be enabled. Switching may be done before xiOpenDevice.

Type: Integer.

Default value: XI_ON

Usage:

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

XI_PRM_PROC_NUM_THREADS or "proc_num_threads"

Description: Number of threads per image processor. An application can change this number in order to optimize performance or decrease number of threads to save resources.

Note: this parameter does not work for MQ, MD camera families and for MU9Px-MH camera.

Type: Integer.

Default value: 0

Typical range: [1, 61]

Usage:

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

Camera FFS

Note: Some of XIMEA cameras contain Flash File System. It allows to store/read small customer file in each camera. For more information visit our knowledge base article How to work with FFS.

XI_PRM_READ_FILE_FFS or "read_file_ffs"

Description: File data to be read from camera flash file system.

Type: String.

Default value: -

Usage:

```
char data_buff[100] = "";
xiGetParamString(handle, XI_PRM_READ_FILE_FFS, data_buff, max_size);
```

Example: Example below reads the sensor defects file. This file exist on xiQ cameras.

```
// set filename
char filename[100] = "bad_pixel_list.txt";
stat = xiSetParamString(xiH, XI_PRM_FFS_FILE_NAME, filename, sizeof(filename));
HandleResult(stat,"xiSetParamString (XI_PRM_FFS_FILE_NAME)");
// allocate buffer
#define MAX_FILE_SIZE 1000*1000 // 1MB
char* file_content = NULL;
file_content = (char*) calloc(1,MAX_FILE_SIZE);
if (!file_content)
{
  printf("Error on memory allocation for file content.\n");
  return;
}
// read file
stat = xiGetParamString(xiH, XI_PRM_READ_FILE_FFS, file_content, MAX_FILE_SIZE);
HandleResult(stat,"xiGetParamString (XI_PRM_READ_FILE_FFS)");
```

```
// print file content
printf("Text read from FFS file:%s\n%s\n\n", filename, file_content);
free(file_content);
```

Example: For reading out Hyper-Spectral sensor calibration data from the camera - use the same code as above, but replace the filename with 'sens_calib.dat'. Code to be used for HSI Calibration:

```
// set filename for HSI Sensor Calibration
char filename[100] = "sens_calib.dat";
stat = xiSetParamString(xiH, XI_PRM_FFS_FILE_NAME, filename, sizeof(filename));
// continue like in example for reading FFS file
```

XI_PRM_WRITE_FILE_FFS or "write_file_ffs"

Description: File data to be written to camera flash file system.

Type: String.

Default value: -

Usage:

```
char value[200] = "";
xiGetParamString(handle, XI_PRM_WRITE_FILE_FFS, &value, sizeof(value));
xiSetParamString(handle, XI_PRM_WRITE_FILE_FFS, value, strlen(value));
```

Example: Write file:

```
xiSetParamString(xiH, XI_PRM_FFS_FILE_NAME, "User1", strlen("User1"));
char* file_content = "ABCDEFGH";
xiSetParamString(xiH, XI_PRM_WRITE_FILE_FFS, file_content, strlen(file_content));
```

Example: Delete file:

```
xiSetParamString(xiH, XI_PRM_FFS_FILE_NAME, "filename.txt", strlen("filename.txt"));
xiSetParamString(xiH, XI_PRM_WRITE_FILE_FFS, NULL, 0);
```

XI_PRM_FFS_FILE_NAME or "ffs_file_name"

Description: Name of file to be written/read from camera FFS.

Note: On MX,MC,CB,MT cameras family, there is limited set of filenames. User's application can use filenames: User1, User2, User3 to store some application specific data.

Type: String.

Default value: -

Usage:

```
char filename[100] = "User1";
xiSetParamString(handle, XI_PRM_FFS_FILE_NAME, filename, size);
```

_FILE_ID or "ffs_file_id"

Description: File number(id) in camera FFS.

Type: Integer.

Default value: 0

Usage:

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

Example: Get list of files:

```
int max_file_id = 0;
        xiGetParamInt(xiH, XI_PRM_FFS_FILE_ID XI_PRM_INFO_MAX, &max_file_id);
    char file_name[MAX_PATH];

    for(int i = 0; i <= max_file_id; i++)
    {
        memset(file_name, 0, MAX_PATH);
        stat = xiSetParamInt(xiH, XI_PRM_FFS_FILE_ID, i);
        stat = xiGetParamString(xiH, XI_PRM_FFS_FILE_NAME, file_name, MAX_PATH);
    }
```

XI_PRM_FFS_FILE_SIZE or "ffs_file_size"

Description: Size of a file specified with parameter XI_PRM_FFS_FILE_IDin bytes.

Type: Integer.

Default value: 0

Usage:

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

XI_PRM_FREE_FFS_SIZE or "free_ffs_size"

Description: Size of free camera flash file system space in bytes.

Type: Unsigned integer 64 bit.

Default value: 0

Usage:

```
uint64_t value = 0;
DWORD size = sizeof(value);
XI_PRM_TYPE type = xiTypeInteger64;
xiGetParam(handle, XI_PRM_FREE_FFS_SIZE, &value, &size, &type);
```

XI_PRM_USED_FFS_SIZE or "used_ffs_size"

Description: Size of used camera flash file system space in bytes.

Type: Unsigned integer 64 bit.

Default value: 0

Usage:

```
uint64_t value = 0;
DWORD size = sizeof(value);
XI_PRM_TYPE type = xiTypeInteger64;
xiGetParam(handle, XI_PRM_USED_FFS_SIZE, &value, &size, &type);
```

XI_PRM_FFS_ACCESS_KEY or "ffs_access_key"

Description: Setting of the key enables file operations on some cameras. It is required to set before usage of XI_PRM_WRITE_FILE_FFS.

Type: Integer.

Default value: 0

Usage:

```
xiSetParamInt(handle, XI_PRM_FFS_ACCESS_KEY, 0x12345678);
```