

CA2D-SDK
Reference Manual



KONICA MINOLTA

●**Note this manual abbreviates product names as follows**

Name Given in This Manual	Official Name
Windows	Microsoft® Windows®
Windows 10	Microsoft® Windows® 10 Operating System
Windows 11	Microsoft® Windows® 11 Operating System

●**Trademark Notices**

Microsoft and Windows are registered trademarks of the Microsoft Corporation, in the US and in other countries.

Other company names and product names that appear within this manual are trademarks or trademarked of their respective companies.

●**Notes of this Manual**

Copy or reproduction of all or any part of the contents of this manual without Konica Minolta's permission is strictly prohibited.

The contents of this manual are subjects to change without prior notice.

Konica Minolta will not accept any responsibility for consequences arising from the use of the software.

Contents

Introduction	5
1. System Environment.....	5
2. Installing/Uninstalling the CA2D-SDK	6
2.1 Installing the CA2D-SDK.....	6
2.2 Uninstalling the CA2D-SDK	6
3. SDK Overview	7
3.1 Function list.....	7
3.2 Basic processing flow	8
3.3 How to create programs with the SDK	10
3.3.1 Measurements.....	10
3.3.2 Getting the spot results.....	10
3.3.3 Performing user calibration.....	10
3.3.4 Switching between multiple instruments.....	11
3.4 Sample codes	12
4. SDK Reference.....	17
4.1 Initialize/terminate	17
CA2DSDK_Enable	17
CA2DSDK_Disable	18
4.2 Connect/disconnect	19
CA2DSDK_ConnectInstrument	19
CA2DSDK_DisconnectInstrument.....	20
CA2DSDK_IsConnectInstrument	21
CA2DSDK_GetInstrumentCount	22
CA2DSDK_GetInstrumentSerialNumber	23
4.3 Setup	24
CA2DSDK_SetInstrumentCondition	24
CA2DSDK_GetInstrumentCondition.....	25
CA2DSDK_SetUserCalibrationData	26
CA2DSDK_GetUserCalibrationData	27
CA2DSDK_SetUserCalibrationData_I(for LabVIEW)	28
CA2DSDK_GetUserCalibrationData_I(for LabVIEW)	29
4.4 Measure/get data	30
CA2DSDK_DoMeasurement	30
CA2DSDK_PollingMeasurement.....	31
CA2DSDK_StopMeasurement	32

CA2DSDK_GetAreaData	33
CA2DSDK_SetDataCondition	34
CA2DSDK_GetDataCondition	35
4.5 Evaluation functions	36
CA2DSDK_ClearEvaluationArea	36
CA2DSDK_SetEvaluationAreaCondition	37
CA2DSDK_GetEvaluationAreaCondition	38
CA2DSDK_AddEvaluationArea	39
CA2DSDK_GetEvaluationAreaCount	40
CA2DSDK_GetEvaluationArea	41
CA2DSDK_SetAlignedSpotCondition	42
CA2DSDK_GetAlignedSpotCondition	43
CA2DSDK_CalculateSpotValue	44
CA2DSDK_GetSpotValue	45
4.6 Others	46
CA2DSDK_DiagnosisInstrument	46
CA2DSDK_PollingDiagnosis	47
CA2DSDK_StopDiagnosis	48
CA2DSDK_GetFinderImage	49
CA2DSDK_CheckPeriodicalCalibration	50
CA2DSDK_GetSDKVersion	51
5. Definitions/Structures	52
5.1 Type definitions	52
5.2 Structure definitions	53
Measurement conditions structure [tagInstrumentCond]	53
User calibration data structure [tagUserCalData]	56
Data acquisition conditions structure [tagGetDataParam]	57
Data configuration structure [tagDataCond]	58
Evaluation area structure [tagEvaluationArea]	59
Evaluation area layout conditions structure [tagEvaluationCond]	60
Spot layout structure [tagAlignedSpotCond]	61
Spot result structure [tagSpotValue]	62
Periodic calibration information structure [tagPeriodicalCallInfo]	63
6. Error Codes	64

Introduction

The CA2D-SDK is a tool for developing PC application software for use with 2D color analyzers CA-2500. This manual describes how to use the SDK. Application developers are assumed to be using Microsoft Visual C++, so the programming methods are described using Microsoft Visual C++.

1. System Environment

The CA2D-SDK requires the following environment.

- OS: Windows 10(x86), Windows 10(x64), Windows 11
 - * Applications will run in 64-bit environments, but a 64-bit version of applications cannot be created.
- Development language: ANSI C interface can be used
- PC equipped with USB 2.0 compliant USB
- Microsoft Visual C++ 2010 SP1 Redistributable Package (x86)

The analyzers that can be controlled with the CA2D-SDK are listed below.

Analyzers: CA-2500

2. Installing/Uninstalling the CA2D-SDK**2.1 Installing the CA2D-SDK**

The CA2D-SDK does not need to be installed. Place it in any desired location on the PC.

Create a product number folder in the same folder position as the DLLs and copy the calibration data included with the product to it.

* If Microsoft Visual C++ 2010 SP1 Redistributable Package (x86) is not installed, double click on vcredist_x86.exe in vcredist_x86 folder and install it.

Installation is unnecessary if CA-S25w has been installed.

The files below are required for application development.

Dynamic link libraries	
CA2DCalculateColor.dll	Color calculation library
CA2DControl.dll	Instrument control library
CA2DEvalRange_Spot.dll	Evaluation area and spot processing library
CA2DSDK.dll	Interface library
ca2kusba.dll	USB control library
ca2kusbu.dll	USB control library
Import library (*)	
CA2DSDK.lib	Interface library
Program source files	
CA2Dcommon.h	File that defines information for using the SDK
CA2DErrorDefine.h	File that defines errors
CA2DTypeDefine.h	Type definition file
Files required in the SDK	
LenxType_Table.ini	Lens information recognition file

(*) Notes: The import library file above is created by Microsoft Visual C++.

2.2 Uninstalling the CA2D-SDK

Delete the files from your PC.

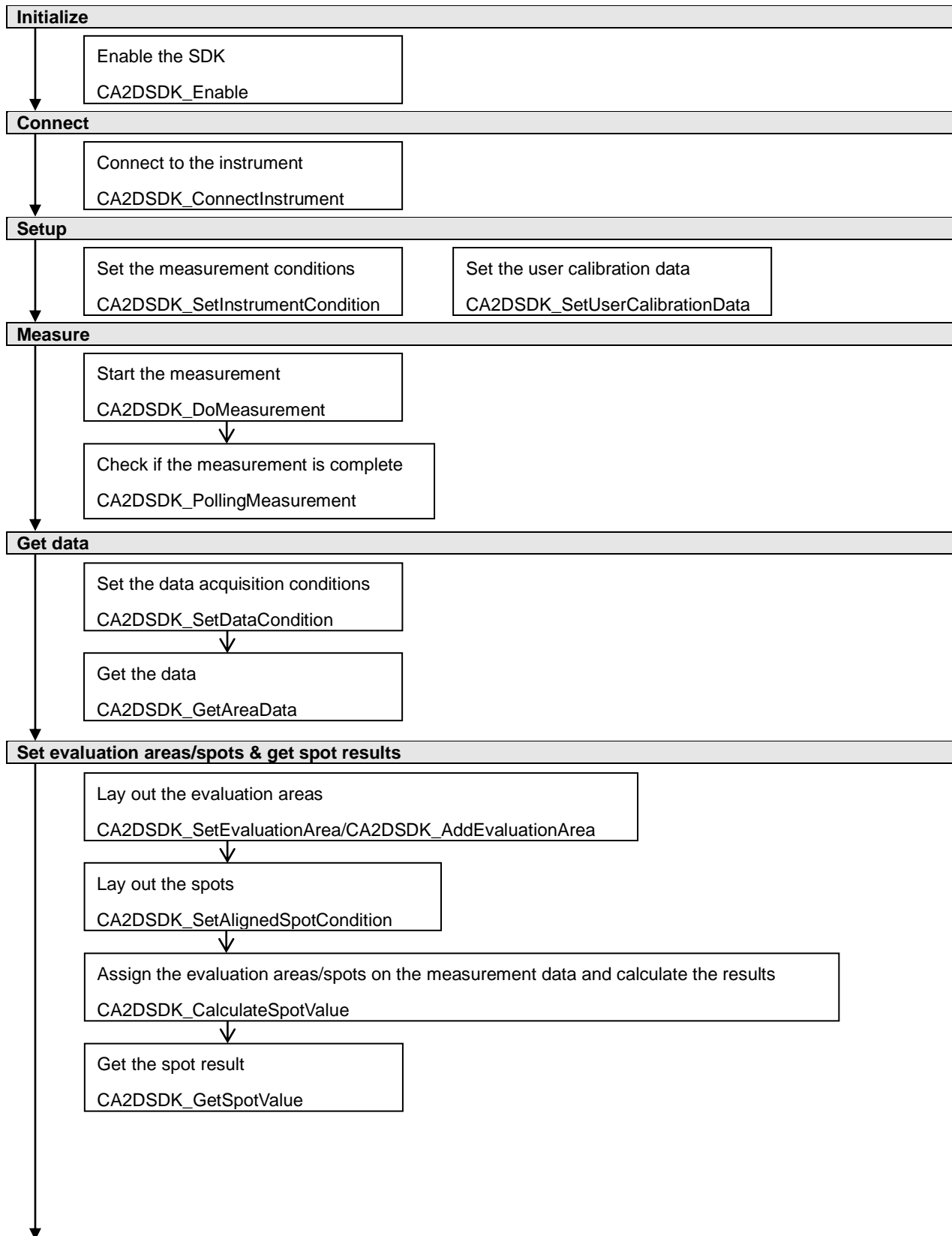
CA2D-SDK Reference Manual**3. SDK Overview****3.1 Function list**

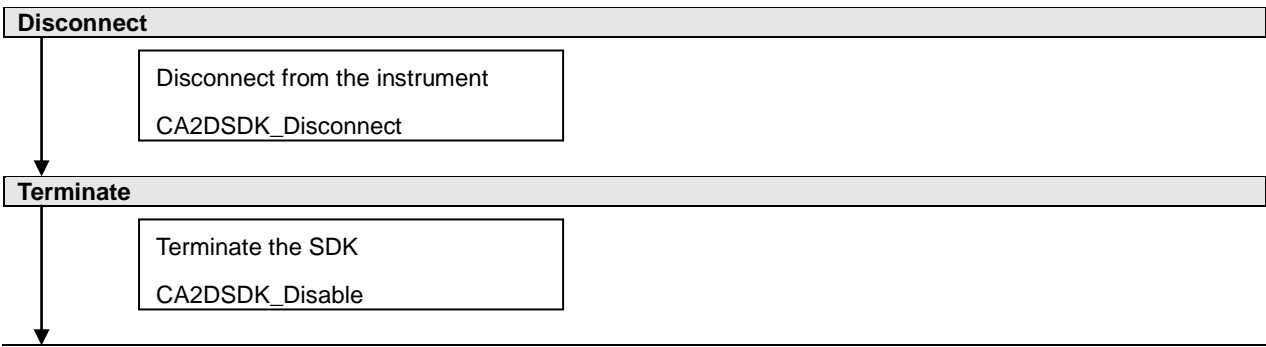
The functions are capable of performing the processing listed below.

Initialize/terminate	
CA2DSDK_Enable	Enable the SDK
CA2DSDK_Disable	End usage of the SDK
Connect/disconnect	
CA2DSDK_ConnectInstrument	Connect to the specified instrument
CA2DSDK_DisconnectInstrument	Disconnect from the connected instrument
CA2DSDK_IsConnectInstrument	Check if the instrument is connected
CA2DSDK_GetInstrumentCount	Get the number of instruments connected to the PC
CA2DSDK_GetInstrumentSerialNumber	Get the serial number of the specified instrument
Setup	
CA2DSDK_SetInstrumentCondition	Set the measurement conditions
CA2DSDK_GetInstrumentCondition	Get the measurement conditions
CA2DSDK_SetUserCalibrationData	Set the user calibration data
CA2DSDK_GetUserCalibrationData	Get the user calibration data
CA2DSDK_SetUserCalibrationData_I	Set the user calibration data(for LabVIEW)
CA2DSDK_GetUserCalibrationData_I	Get the user calibration data(for LabVIEW)
Measure/get data	
CA2DSDK_DoMeasurement	Start the measurement
CA2DSDK_PollingMeasurement	Check if the measurement is complete
CA2DSDK_StopMeasurement	Stop the measurement
CA2DSDK_GetAreaData	Get the data for the specified area
CA2DSDK_SetDataCondition	Set the conditions to calculate data
CA2DSDK_GetDataCondition	Get the conditions to calculate data
Evaluation functions	
CA2DSDK_ClearEvaluationArea	Clear the evaluation areas
CA2DSDK_SetEvaluationAreaCondition	Set the evaluation area layout conditions
CA2DSDK_GetEvaluationAreaCondition	Get the evaluation area layout conditions
CA2DSDK_AddEvaluationArea	Add an evaluation area (use when the layout condition is manual)
CA2DSDK_GetEvaluationAreaCount	Get the number of evaluation areas
CA2DSDK_GetEvaluationArea	Get the evaluation area
CA2DSDK_SetAlignedSpotCondition	Set the conditions to lay out the spots
CA2DSDK_GetAlignedSpotCondition	Get the conditions to lay out the spot
CA2DSDK_CalculateSpotValue	Calculate the spot results
CA2DSDK_GetSpotValue	Get the spot result
Others	
CA2DSDK_DiagnosisInstrument	Start the instrument diagnostic
CA2DSDK_PollingDiagnosis	Check if the instrument diagnostic is complete
CA2DSDK_StopDiagnosis	Stop the instrument diagnostic
CA2DSDK_GetFinderImage	Get the finder image
CA2DSDK_CheckPeriodicalCalibration	Check if periodic calibration is required
CA2DSDK_GetSDKVersion	Get the SDK version

3.2 Basic processing flow

The basic processing flow is listed below.





3.3 How to create programs with the SDK

3.3.1 Measurements

The basic steps are listed below. As necessary, repeat steps 3, 4, and 5.

1. Initialize	• Enable the SDK [CA2DSDK_Enable]
2. Connect	• Connect to the instrument [CA2DSDK_ConnectInstrument]
3. Setup	• Reflect the settings to the instrument [CA2DSDK_SetInstrumentCondition]
4. Measure/get data	• Run the measurement [CA2DSDK_DoMeasurement] • Check if the measurement is complete [CA2DSDK_PollingMeasurement]
5. Get data	• Set the data conditions [CA2DSDK_SetDataCondition] • Get the data [CA2DSDK_GetAreaData]
6. Disconnect	• Disconnect from the instrument [CA2DSDK_DisconnectInstrument]
7. Terminate	• End usage of the SDK [CA2DSDK_Disable]

3.3.2 Getting the spot results

To get the spot results, you must configure the settings for the evaluation areas/spots.

The evaluation areas and spots are laid out when [CA2DSDK_CalculateSpotValue] is called, so they can be configured either before or after the measurements.

3.3.2.1 Automatically laying out evaluation areas

This section describes an example of automatically laying out evaluation areas after getting the data.

5. Get spot results	<ul style="list-style-type: none"> • Clear evaluation areas [CA2DSDK_ClearEvaluationArea] • Set evaluation area layout conditions to automatic layout [CA2DSDK_SetEvaluationAreaCondition] • Set the conditions to lay out the spots [CA2DSDK_SetAlignedSpotCondition] • Calculate spot results [CA2DSDK_CalculateSpotValue] • Get spot results [CA2DSDK_GetSpotValue]
---------------------	---

3.3.2.2 Manually laying out evaluation areas

This section describes an example of manually laying out evaluation areas after getting the data.

5. Get spot results	<ul style="list-style-type: none"> • Clear evaluation areas [CA2DSDK_ClearEvaluationArea] • Set evaluation area layout conditions to manual layout [CA2DSDK_SetEvaluationAreaCondition] • Set evaluation areas [CA2DSDK_AddEvaluationArea] • Set the conditions to lay out the spots [CA2DSDK_SetAlignedSpotCondition] • Calculate spot results [CA2DSDK_CalculateSpotValue] • Get spot results [CA2DSDK_GetSpotValue]
---------------------	--

3.3.3 Performing user calibration

For user calibration, set the data to use in step 3 in 3.1.

User calibration is performed when the measurement data is acquired, so if the user calibration is changed after measuring, those changes are reflected in the next and subsequent measurements.

3. Setup	<ul style="list-style-type: none"> • Set the user calibration data [CA2DSDK_SetUserCalibrationData] • Reflect the settings to the instrument [CA2DSDK_SetInstrumentCondition]
----------	---

3.3.4 Switching between multiple instruments

To switch between multiple instruments, in the steps in 3.1, connect to the next instrument (step 2) after disconnecting from the instrument (step 6).

3.4 Sample codes

```
int ret(CA2D_OK);

//Enable the SDK
ret = CA2DSDK_Enable();
if (ret < 0){
    return;
}

//Connect to the specified instrument
ret = CA2DSDK_ConnectInstrument(0);
if (ret < 0){
    CA2DSDK_Disable();
    return;
}

//Set the measurement conditions
tagInstrumentCond cond;
cond.lensType = LENS_NORMAL;           //Normal lens
cond.lensPosition = 0;                 //0.25m
cond.exposureMode = EXPOSURE_AUTO;    //Auto exposure
cond.measurementType = SYNCMODE_OFF;  //Non-synchronized measurements
cond.syncValue = 60.0;
cond.exposureIndex = 11;
cond.additional = 1;                   //Integration times
cond.left = 0;                         //Exposure setting area
cond.top = 0;
cond.right = 979;
cond.bottom = 979;
cond.filterMeasure = FILTER_MEASURE_OFF; //XYZ measurements
cond.filterIndex = FILTER_INDEX_Y;
cond.smearIndex = SMEAR_NONE;          //Smear compensation : None
cond.userCal = USERCAL_OFF;           //User calibration : OFF
cond.rotate = ROTATION_NONE;           //Output unmodified
```

```
ret = CA2DSDK_SetInstrumentCondition(&cond);
if (ret < 0){
    CA2DSDK_DisconnectInstrument();
    CA2DSDK_Disable();
    return;
}

//Start the measurement
ret = CA2DSDK_DoMeasurement();
if (ret < 0){
    CA2DSDK_DisconnectInstrument();
    CA2DSDK_Disable();
    return;
}
while(1){
    ret = CA2DSDK_PollingMeasurement();
    if (ret == CA2D_OK){
        //When measurement is complete
        break;
    }else if (ret >= CA2D_OK){
        //When measuring
    }else{
        //When an error has occurred
        CA2DSDK_DisconnectInstrument();
        CA2DSDK_Disable();
        return;
    }
    Sleep(100);
}

//The conditions to calculate data
tagDataCond cond_d;
cond_d.lower_enable = 0; //under errors : OFF
cond_d.lower_item = 1;
cond_d.lower_threshold = 3;
```

```
cond_d.resolution = 0;                                //Resolution : 980*980

//Data acquisition area
tagGetDataParam paramArea;
paramArea.left = 0;
paramArea.top = 0;
paramArea.right = 979;
paramArea.bottom = 979;

float_km* pData = new float[MAXDATAROW * MAXDATACOL];
//Gets X/Y/Z image data
for (int j = VALTYPE_X; j <= VALTYPE_Z; j++){
    //Set the conditions to calculate data
    cond_d.valueType = j;
    ret = CA2DSDK_SetDataCondition(&cond_d);
    if (ret < 0){
        break;
    }

    //Get the data for the specified area
    ret = CA2DSDK_GetAreaData(&paramArea, pData);
    if (ret < 0){
        break;
    }

    //The image is stored in pData, so process the data as necessary
}
delete [] pData;

//Clear the evaluation areas
ret = CA2DSDK_ClearEvaluationArea();
if (ret < 0){
    CA2DSDK_DisconnectInstrument();
}
```

```
CA2DSDK_Disable();
return;
}

//Set the evaluation area layout conditions
tagEvaluationCond cond_e;
cond_e.type = 0; //Manual layout
ret = CA2DSDK_SetEvaluationAreaCondition(&cond_e);
if (ret < 0){
    CA2DSDK_DisconnectInstrument();
    CA2DSDK_Disable();
    return;
}

//Set the conditions to lay out the spots
tagAlignedSpotCond cond_spot;
cond_spot.row = 3; //Vertical layout count
cond_spot.col = 3; //Horizontal layout count
cond_spot.shape = SPOT_SHAPETYPE_CIRCLE; //Circle
cond_spot.height = 50; //Circle diameter : 50pixel
cond_spot.width = 50;
cond_spot.offset_input = 1; //Relative values
cond_spot.offset_position = 1; //Set spot center as edge
cond_spot.offset_left = 10; //Left margin : H/10
cond_spot.offset_top = 10; //Top margin : V/10
cond_spot.offset_right = 10; //Right margin : H/10
cond_spot.offset_bottom = 10; //Bottom margin V/10
ret = CA2DSDK_SetAlignedSpotCondition(&cond_spot);
if (ret < 0){
    CA2DSDK_DisconnectInstrument();
    CA2DSDK_Disable();
    return;
}

//Calculate the spot results
ret = CA2DSDK_CalculateSpotValue();
```

```
if (ret < 0){
    CA2DSDK_DisconnectInstrument();
    CA2DSDK_Disable();
    return;
}

//Get the spot result
tagSpotValue spot_val;
spot_val.color = 1;
for (int i = 0; i < 9; i++){
    ret = CA2DSDK_GetSpotValue(0, i, &spot_val);
    if (ret < 0){
        break;
    }

    //The spot values are stored in spot_val, so process the data as necessary
}

//Disconnect from the connected instrument
CA2DSDK_DisconnectInstrument();

//End usage of the SDK
CA2DSDK_Disable();
```


4. SDK Reference

4.1 Initialize/terminate

CA2DSDK_Enable

Overview:

Enable the SDK

Syntax:

CA2DSDK_ER CA2DSDK_Enable()

[Arguments]

None

[Return values]

CA2D_OK	Completed normally
CA2D_ER_ENABLE	Initialization failed
CA2D_ER_FILE_NOTFOUND_LENSTYPE	The lens type file does not exist
CA2D_ER_COMMUNICATION	Could not communicate with the instrument
CA2D_ER_COM_MEMORY	Insufficient memory
CA2D_ER_COM_DETECT	The instrument was disconnected
CA2D_ER_USBDLL	USB driver control is not operating correctly

Description:

This API enables the SDK.

When using the SDK, always call this API.

When this API is used, the instruments connected to the PC can be connected to with [CA2DSDK_ConnectInstrument](#). Use this API when the instruments are connected to the PC.

After using this API, the instrument information can be updated by calling this API when changing the instruments connected to the PC.

However, if this API cannot be used because an instrument is currently connected, it returns CA2D_ER_ENABLE.

To terminate the SDK, always call [CA2DSDK_Disable](#).

CA2DSDK_Disable

Overview:

End usage of the SDK

Syntax:

CA2DSDK_ER CA2DSDK_Disable()

[Arguments]

None

[Return values]

CA2D_OK	Completed normally
---------	--------------------

Description:

This function ends usage of the SDK.
To terminate the SDK, always call this API.

4.2 Connect/disconnect

CA2DSDK_ConnectInstrument

Overview:

Connect to the specified instrument

Syntax:

CA2DSDK_ER CA2DSDK_ConnectInstrument(int32_km index)

[Arguments]

index	Specify the instrument number to connect to See CA2DSDK_GetInstrumentCount for the number to specify
-------	---

[Return values]

CA2D_OK	Completed normally
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_INSTRUMENT	The instrument information is incorrect
CA2D_ER_CONNECT	Connection failed
CA2D_ER_PARAM_INDEX	Invalid instrument specification
CA2D_ER_COMMUNICATION	Could not communicate with the instrument
CA2D_ER_COM_FILTER	Filter control is not operating correctly
CA2D_ER_COM_SHUTTERSPEED	Shutter speed control is not operating correctly
CA2D_ER_COM_GAIN	Gain control is not operating correctly
CA2D_ER_COM_ADDITIONAL	Cumulative number control is not operating correctly
CA2D_ER_COM_FAN	Fan control is not operating correctly
CA2D_ER_COM_MOTOR	Motor control is not operating correctly
CA2D_ER_COM_MEMORY	Insufficient memory
CA2D_ER_COM_DETECT	The instrument was disconnected
CA2D_ER_USBDLL	USB driver control is not operating correctly

Description:

This function connects to the specified instrument.

When multiple instruments are connected to the PC, use [CA2DSDK_GetInstrumentNumber](#) to check the instrument to use before connecting to it.

Only one instrument can be connected to at one time. Simultaneous connections to multiple instruments are not possible.

To switch the instrument, first disconnect from the instrument with [CA2DSDK_DisconnectInstrument](#), and then connect to the next instrument.

The connection state can be checked with [CA2DSDK_IsConnectInstrument](#).

CA2DSDK_DisconnectInstrument**Overview:**

Disconnect from the connected instrument

Syntax:

CA2DSDK_ER CA2DSDK_DisconnectInstrument()

[Arguments]

None

[Return values]

CA2D_OK	Completed normally
CA2D_ER_ENABLE	Initialization is not complete

Description:

This function disconnects from the connected instrument.

CA2DSDK_IsConnectInstrument**Overview:**

Check the connection state

Syntax:

CA2DSDK_ER CA2DSDK_IsConnectInstrument()

[Arguments]

None

[Return values]

Positive value	A connected instrument is present and the return value is the serial number for that instrument
CA2D_ER_ENABLE	Initialization is not complete

Description:

This function checks the connection state.

If a connected instrument is present, this function returns the instrument's serial number.

If the number of instruments connected to the PC changes while an instrument is connected, the correct value may not be obtainable.

In this situation, we recommend you terminate with [CA2DSDK_Disable](#) and start again from [CA2DSDK_Enable](#).

For example, consider the following cases.

- When the cable is prematurely disconnected
- When the connection of the instruments is switched in the connection state

CA2DSDK_GetInstrumentCount**Overview:**

Get the number of instruments connected to the PC

Syntax:

CA2DSDK_ER CA2DSDK_GetInstrumentCount()

[Arguments]

None

[Return values]

0 or higher value	The number of instruments connected to the PC
CA2D_ER_ENABLE	Initialization is not complete

Description:

This function gets the number of instruments connected to the PC.

The number of instruments is fixed when [CA2DSDK_Enable](#) is called.

The numbers to specify in [CA2DSDK_ConnectInstrument](#) are determined by the number of instruments in this API. For example, if the number of instruments that was acquired is 2, then the numbers that can be specified are 0 and 1.

Specify the instrument by first checking the serial number with [CA2DSDK_GetInstrumentSerialNumber](#).

CA2DSDK_GetInstrumentSerialNumber**Overview:**

Get the serial number of the specified instrument

Syntax:

CA2DSDK_ER CA2DSDK_GetInstrumentSerialNumber(int32_km index)

[Arguments]

index	Specify the instrument number to connect to See CA2DSDK_GetInstrumentCount for the number to specify
-------	---

[Return values]

Positive value	Instrument serial number
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_PARAM_INDEX	Invalid instrument specification

Description:

This function gets the serial number of the specified instrument.

The instrument information is the information that was acquired when [CA2DSDK_Enable](#) was called.

If the connection state is changed, you must use [CA2DSDK_Enable](#) to update the information.

4.3 Setup

CA2DSDK_SetInstrumentCondition

Overview:

Set the measurement conditions

Syntax:

CA2DSDK_ER CA2DSDK_SetInstrumentCondition(const tagInstrumentCond* pInstCond)

[Arguments]

pInstCond	Pointer to the measurement conditions structure
-----------	---

[Return values]

CA2D_OK	Completed normally
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_CONNECT	Connection is not complete
CA2D_ER_MEASURING	Cannot be processed because the measurement is running
CA2D_ER_RUNNING_DIAGNOSIS	Cannot be processed because the instrument diagnostic is running
CA2D_ER_USERDATA_NONE	User calibration data has not been set
CA2D_ER_FILE_NOTFOUND_CALIBRATION	Calibration data file does not exist
CA2D_ER_PARAM	Invalid argument specification
CA2D_ER_PARAM_LENSTYPE	Invalid lens type specification (lens cannot be used)
CA2D_ER_PARAM_LENSPOS	Invalid focus ring distance indicator specification
CA2D_ER_PARAM_EXPOSURE_MODE	Invalid exposure mode specification
CA2D_ER_PARAM_EXPOSURE_INDEX	Invalid exposure table index number specification
CA2D_ER_PARAM_EXPOSURE_AREA	Invalid exposure area setting
CA2D_ER_PARAM_SYNC_MODE	Invalid synchronized measurement mode specification
CA2D_ER_PARAM_SYNC_VALUE	Invalid synchronization frequency specification
CA2D_ER_PARAM_ADDITIONAL	Invalid cumulative number specification
CA2D_ER_PARAM_FILTER	Invalid filter measurement specification
CA2D_ER_PARAM_FILTER_INDEX	Invalid filter type specification
CA2D_ER_PARAM_SMEAR	Invalid smear compensation specification
CA2D_ER_PARAM_USERCAL	Invalid user calibration specification
CA2D_ER_PARAM_ROTATION	Invalid image orientation specification
CA2D_ER_COMMUNICATION	Could not communicate with the instrument
CA2D_ER_COM_FILTER	Filter control is not operating correctly
CA2D_ER_COM_SHUTTERSPEED	Shutter speed control is not operating correctly
CA2D_ER_COM_GAIN	Gain control is not operating correctly
CA2D_ER_COM_ADDITIONAL	Cumulative number control is not operating correctly
CA2D_ER_COM_MOTOR	Motor control is not operating correctly
CA2D_ER_COM_MEMORY	Insufficient memory
CA2D_ER_COM_DETECT	The instrument was disconnected
CA2D_ER_USBDLL	USB driver control is not operating correctly

Description:

This function sets the measurement conditions.

To use user calibration, you must set the user calibration data by using [CA2DSDK_SetUserCalibrationData](#).

If this function returns CA2D_ER_FILE_NOTFOUND_CALIBRATION, check if the specified path to the calibration data included with the product actually exists.

CA2DSDK_GetInstrumentCondition**Overview:**

Get the measurement conditions

Syntax:

CA2DSDK_ER CA2DSDK_GetInstrumentCondition(tagInstrumentCond* pInstCond)

[Arguments]

pInstCond	Pointer to the measurement conditions structure
-----------	---

[Return values]

CA2D_OK	Completed normally
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_CONNECT	Connection is not complete
CA2D_ER_MEASURING	Cannot be processed because the measurement is running
CA2D_ER_RUNNING_DIAGNOSIS	Cannot be processed because the instrument diagnostic is running
CA2D_ER_PARAM	Invalid argument specification

Description:

This function gets the measurement conditions.

CA2DSDK_SetUserCalibrationData**Overview:**

Set the user calibration data

Syntax:

CA2DSDK_ER CA2DSDK_SetUserCalibrationData(const tagUserCalData* pUserCal)

[Arguments]

pUserCal	Pointer to the user calibration data structure
----------	--

[Return values]

CA2D_OK	Completed normally
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_MEASURING	Cannot be processed because the measurement is running
CA2D_ER_RUNNING_DIAGNOSIS	Cannot be processed because the instrument diagnostic is running
CA2D_ER_USERDATA_CALC	The user calibration data is invalid (calibration factor could not be calculated)
CA2D_ER_PARAM	Invalid argument specification
CA2D_ER_PARAM_CALTYPE	Invalid calibration type specification

Description:

This function sets the user calibration data.

To enable user calibration data, turn on the setting with [CA2DSDK_SetInstrumentCondition](#).

* Use [CA2DSDK_SetUserCalibrationData](#) when using with LabVIEW-.

CA2DSDK_GetUserCalibrationData**Overview:**

Get the user calibration data

Syntax:

CA2DSDK_ER CA2DSDK_GetUserCalibrationData(tagUserCalData* pUserCal)

[Arguments]

pUserCal	Pointer to the user calibration data structure
----------	--

[Return values]

CA2D_OK	Completed normally
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_MEASURING	Cannot be processed because the measurement is running
CA2D_ER_RUNNING_DIAGNOSIS	Cannot be processed because the instrument diagnostic is running
CA2D_ER_USERDATA_NONE	User calibration data has not been set
CA2D_ER_PARAM	Invalid argument specification

Description:

This function gets the user calibration data.

* Use [CA2DSDK_GetUserCalibrationData](#) | when using with LabVIEW.

CA2D-SDK Reference Manual**CA2DSDK_SetUserCalibrationData_I(for LabVIEW)****Overview:**

Set the user calibration data

Syntax:

CA2DSDK_ER CA2DSDK_SetUserCalibrationData_I(int16_km caltype, const double_km* pR, const double_km* pG, const double_km* pB, const double_km* pW)

[Arguments]

caltype	Calibration type		
	Value	Definition	Description
	0	CALTYPE_NONE	No calibration
	1	CALTYPE_NORMAL	One-color calibration
	2	CALTYPE_RGB	RGB calibration
	3	CALTYPE_WRGB	WRGB calibration
pR	Red data (used for RGB calibration and WRGB calibration) pR[3][2] [3]: Lv, x · y order [2]: Pre-calibration, post-calibration order		
pG	Green data (used for RGB calibration and WRGB calibration) pG[3][2] [3]: Lv, x · y order [2]: Pre-calibration, post-calibration order		
pB	Blue data (used for RGB calibration and WRGB calibration) pB[3][2] [3]: Lv, x · y order [2]: Pre-calibration, post-calibration order		
pW	White data (used for one-color calibration, RGB calibration, and WRGB calibration) pW[3][2] [3]: Lv, x · y order [2]: Pre-calibration, post-calibration order		

[Return values]

CA2D_OK	Completed normally
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_MEASURING	Cannot be processed because the measurement is running
CA2D_ER_RUNNING_DIAGNOSIS	Cannot be processed because the instrument diagnostic is running
CA2D_ER_USERDATA_CALC	The user calibration data is invalid (calibration factor could not be calculated)
CA2D_ER_PARAM	Invalid argument specification
CA2D_ER_PARAM_CALTYPE	Invalid calibration type specification

Description:

This function sets the user calibration data.

To enable user calibration data, turn on the setting with [CA2DSDK_SetInstrumentCondition](#).

CA2D-SDK Reference Manual**CA2DSDK_GetUserCalibrationData_I(for LabVIEW)****Overview:**

Get the user calibration data

Syntax:

CA2DSDK_ER CA2DSDK_GetUserCalibrationData_I(int16_km* pCaltype, double_km* pR, double_km * pG, double_km * pB, double_km * pW)

[Arguments]

pCaltype	Calibration type		
	Value	Definition	Description
	0	CALTYPE_NONE	No calibration
	1	CALTYPE_NORMAL	One-color calibration
	2	CALTYPE_RGB	RGB calibration
	3	CALTYPE_WRGB	WRGB calibration
pR	Red data (used for RGB calibration and WRGB calibration) pR[3][2] [3]: Lv, x · y order [2]: Pre-calibration, post-calibration order		
pG	Green data (used for RGB calibration and WRGB calibration) pG[3][2] [3]: Lv, x · y order [2]: Pre-calibration, post-calibration order		
pB	Blue data (used for RGB calibration and WRGB calibration) pB[3][2] [3]: Lv, x · y order [2]: Pre-calibration, post-calibration order		
pW	White data (used for one-color calibration, RGB calibration, and WRGB calibration) pW[3][2] [3]: Lv, x · y order [2]: Pre-calibration, post-calibration order		

[Return values]

CA2D_OK	Completed normally
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_MEASURING	Cannot be processed because the measurement is running
CA2D_ER_RUNNING_DIAGNOSIS	Cannot be processed because the instrument diagnostic is running
CA2D_ER_USERDATA_NONE	User calibration data has not been set
CA2D_ER_PARAM	Invalid argument specification

Description:

This function gets the user calibration data.

4.4 Measure/get data**CA2DSDK_DoMeasurement****Overview:**

Start the measurement

Syntax:

CA2DSDK_ER CA2DSDK_DoMeasurement()

[Arguments]

None

[Return values]

CA2D_OK	Completed normally
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_CONNECT	Connection is not complete
CA2D_ER_RUNNING_DIAGNOSIS	Cannot be processed because the instrument diagnostic is running
CA2D_ER_TEMPERATURE	Temperature error
CA2D_ER_FILE_NOTFOUND_CALIBRATION	Calibration data file does not exist
CA2D_ER_COMMUNICATION	Could not communicate with the instrument
CA2D_ER_COM_TEMPERATURE	Temperature control is not operating correctly
CA2D_ER_COM_FAN	Fan control is not operating correctly
CA2D_ER_COM_MEMORY	Insufficient memory
CA2D_ER_COM_DETECT	The instrument was disconnected
CA2D_ER_USBDLL	USB driver control is not operating correctly

Description:

This function starts the measurement.

It returns a return value when the measurement starts.

For example, even when a measurement error occurs due to overexposure, this function completes normally.

Check for completion of the measurement and errors during the measurement by using [CA2DSDK_PollingMeasurement](#).

CA2DSDK_PollingMeasurement**Overview:**

Check the measurement

Syntax:

CA2DSDK_ER CA2DSDK_PollingMeasurement()

[Arguments]

None

[Return values]

CA2D_CANCEL	Processing was canceled
CA2D_OK	Completed normally
CA2D_OK_MEASURING	Measuring
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_CONNECT	Connection is not complete
CA2D_ER_RUNNING_DIAGNOSIS	Cannot be processed because the instrument diagnostic is running
CA2D_ER_AUTOEXPO_OVER	Failed to calculate the exposure time (overexposed)
CA2D_ER_AUTOEXPO_UNDER	Failed to calculate the exposure time (underexposed)
CA2D_ER_FILE_NOTFOUND_CALIBRATION	Calibration data file does not exist
CA2D_ER_MEASURE_CALC	Measurement calculation failed
CA2D_ER_COMMUNICATION	Could not communicate with the instrument
CA2D_ER_COM_FILTER	Filter control is not operating correctly
CA2D_ER_COM_STATUS	Get status is not operating correctly
CA2D_ER_COM_MEASURE	Measurement control is not operating correctly
CA2D_ER_COM_SHUTTERSPEED	Shutter speed control is not operating correctly
CA2D_ER_COM_GAIN	Gain control is not operating correctly
CA2D_ER_COM_ADDITIONAL	Cumulative number control is not operating correctly
CA2D_ER_COM_GETDATA	Data acquisition control is not operating correctly
CA2D_ER_COM_MOTOR	Motor control is not operating correctly
CA2D_ER_COM_MEMORY	Insufficient memory
CA2D_ER_COM_DETECT	The instrument was disconnected
CA2D_ER_USBDLL	USB driver control is not operating correctly

Description:

This function checks the measuring and measurement complete states.

When measuring, this function returns CA2D_OK_MEASURING. In all other cases, it returns CA2D_OK or an error.

For CA2D_ER_AUTOEXPO_OVER or CA2D_ER_AUTOEXPO_UNDER, consider the following causes.

- 1) The measurement subject is intermittent light
- 2) Instrument diagnostics have not been run

For 1), the problem may be resolved by performing synchronized measurements or measuring with a manual exposure.

If instrument diagnostics have not been run, we recommend running diagnostics.

CA2DSDK_StopMeasurement**Overview:**

Stop the completion of the measurement

Syntax:

CA2DSDK_ER CA2DSDK_StopMeasurement()

[Arguments]

None

[Return values]

CA2D_OK	Completed normally
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_CONNECT	Connection is not complete
CA2D_ER_RUNNING_DIAGNOSIS	Cannot be processed because the instrument diagnostic is running
CA2D_ER_COMMUNICATION	Could not communicate with the instrument
CA2D_ER_COM_MEMORY	Insufficient memory
CA2D_ER_COM_DETECT	The instrument was disconnected
CA2D_ER_USBDLL	USB driver control is not operating correctly

Description:

This function stops the measurement.

When the measurement finishes stopping, [CA2DSDK_PollingMeasurement](#) returns CA2D_CANCEL.

If no measurement is running, it returns CA2D_OK.

CA2DSDK_GetAreaData**Overview:**

Get the data

Syntax:

CA2DSDK_ER CA2DSDK_GetAreaData(const tagGetDataParam* pDataParam, float_km* pData)

[Arguments]

pDataParam	Pointer to the data acquisition conditions structure
pData	Pointer to the acquisition data The data is stored from the upper left to the lower right

[Return values]

CA2D_OK	Completed normally
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_CONNECT	Connection is not complete
CA2D_ER_MEASURING	Cannot be processed because the measurement is running
CA2D_ER_RUNNING_DIAGNOSIS	Cannot be processed because the instrument diagnostic is running
CA2D_ER_DATA_NONE	No measurement data (not measured or measurement error)
CA2D_ER_PARAM	Invalid argument specification
CA2D_ER_PARAM_GETAREA	Invalid data acquisition area specification

Description:

This function gets the data.

The data is calculated with the conditions specified with [CA2DSDK_SetDataCondition](#).

[Notes]

The data image may contain values that indicate errors.

- Over error pixel (set when an extremely bright image is measured)
-3.3028231e+38
- Under error pixel (set when an extremely dark image is measured)
-2.4028230e+38
- Calculation error pixel (set when calculating the color specification value)
-1.4028230e+38

Since the values above are error pixels when included in the image, do not use them.

The error value may also include a margin of error, so we recommend the error be determined in the following manner.

```

if (val < -3.0E+38){
    // Over error pixel
}else if ( (-3.0E+38 <= val)&&(val < -2.0E+38) ){
    // Under error pixel
}else if (val < -1.0E+38){
    // Calculation error pixel
}else{
    // Normal pixel
}

```

CA2DSDK_SetDataCondition

Overview:

Set the conditions for calculating data

Syntax:

CA2DSDK_ER CA2DSDK_SetDataCondition(const tagDataCond* pDataCond)

[Arguments]

pDataCond	Pointer to the data configuration structure
-----------	---

[Return values]

CA2D_OK	Completed normally
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_CONNECT	Connection is not complete
CA2D_ER_PARAM	Invalid argument specification
CA2D_ER_PARAM_VALTYPE	Invalid color specification value specification
CA2D_ER_PARAM_RESOLUTION	Invalid resolution specification
CA2D_ER_PARAM_LOWER_LEVEL	Invalid threshold value specification to process as an under error

Description:

This function sets the conditions for calculating data.

The data acquired by using [CA2DSDK_GetAreaData](#) is determined by the settings of this API.

If you want to get multiple types of color value data for one item of measurement data, repeatedly use this API and [CA2DSDK_GetAreaData](#).

The data to use in "[4.5 Evaluation functions](#)" is executed using the conditions set here.

CA2DSDK_GetDataCondition**Overview:**

Get the data settings

Syntax:

CA2DSDK_ER CA2DSDK_GetDataCondition(tagDataCond* pDataCond)

[Arguments]

pDataCond	Pointer to the data configuration structure
-----------	---

[Return values]

CA2D_OK	Completed normally
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_CONNECT	Connection is not complete
CA2D_ER_PARAM	Invalid argument specification

Description:

This function gets the conditions to calculate data.

4.5 Evaluation functions

CA2DSDK_ClearEvaluationArea

Overview:

Clear the evaluation areas

Syntax:

CA2DSDK_ER CA2DSDK_ClearEvaluationArea()

[Arguments]

None

[Return values]

CA2D_OK	Completed normally
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_MEASURING	Cannot be processed because the measurement is running
CA2D_ER_RUNNING_DIAGNOSIS	Cannot be processed because the instrument diagnostic is running

Description:

This function clears the evaluation areas.

After this function is called, the evaluation area becomes the entire area.

The spot results are also cleared, so you must calculate the spot results by using [CA2DSDK_CalculateSpotValue](#).

CA2DSDK_SetEvaluationAreaCondition**Overview:**

Set the evaluation area layout conditions.

Syntax:

CA2DSDK_ER CA2DSDK_SetEvaluationAreaCondition(const tagEvaluationCond* pCond)

[Arguments]

pCond	Pointer to the evaluation area layout conditions structure
-------	--

[Return values]

CA2D_OK	Completed normally
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_MEASURING	Cannot be processed because the measurement is running
CA2D_ER_RUNNING_DIAGNOSIS	Cannot be processed because the instrument diagnostic is running
CA2D_ER_PARAM	Invalid argument specification
CA2D_ER_PARAM_EVAL_TYPE	Invalid layout type specification
CA2D_ER_PARAM_EVAL_AREA	Invalid detection area specification
CA2D_ER_PARAM_EVAL_COUNT	Invalid detection count specification
CA2D_ER_PARAM_EVAL_LEVEL	Invalid detection threshold value specification

Description:

This function lays out the evaluation areas with the specified conditions.

When this API is used, the evaluation areas that were previously set are cleared (same operation as [CA2DSDK_ClearEvaluationArea](#)).

The spot results are also cleared, so you must calculate the spot results by using [CA2DSDK_CalculateSpotValue](#).

Evaluation areas are either automatically laid out or manually laid out.

To automatically lay out evaluation areas, set the detection conditions in the argument. Detection is performed when [CA2DSDK_CalculateSpotValue](#) is used.

To manually lay out evaluation areas, use [CA2DSDK_AddEvaluationArea](#) to register the evaluation areas.

CA2DSDK_GetEvaluationAreaCondition**Overview:**

Get the evaluation area layout conditions.

Syntax:

CA2DSDK_ER CA2DSDK_SetEvaluationAreaCondition(const tagEvaluationCond* pCond)

[Arguments]

pCond	Pointer to the evaluation area layout conditions structure
-------	--

[Return values]

CA2D_OK	Completed normally
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_MEASURING	Cannot be processed because the measurement is running
CA2D_ER_RUNNING_DIAGNOSIS	Cannot be processed because the instrument diagnostic is running
CA2D_ER_PARAM	Invalid argument specification

Description:

This function gets the evaluation area layout conditions.

CA2DSDK_AddEvaluationArea**Overview:**

Add an evaluation area

Syntax:

CA2DSDK_ER CA2DSDK_AddEvaluationArea(const tagEvaluationArea* pArea)

[Arguments]

pArea	Pointer to the evaluation area structure
-------	--

[Return values]

CA2D_OK	Completed normally
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_MEASURING	Cannot be processed because the measurement is running
CA2D_ER_RUNNING_DIAGNOSIS	Cannot be processed because the instrument diagnostic is running
CA2D_ER_PARAM	Invalid argument specification
CA2D_ER_PARAM_EVAL_TYPE	Invalid layout type specification
CA2D_ER_PARAM_EVAL_AREA	Invalid evaluation area specification

Description:

This function adds an evaluation area.

Adding evaluation areas is only possible for manual layout in [CA2DSDK_SetEvaluationAreaCondition](#).

This function returns CA2D_ER_PARAM_EVAL_TYPE for automatic layout.

The spot results are cleared when an evaluation area is added, so you must calculate the spot results by using [CA2DSDK_CalculateSpotValue](#).

The number is assigned in the order added starting from 0.

The entire area is registered for the initial value, but when the first area is added, it is newly registered from 0.

Up to 144 evaluation areas can be added.

CA2DSDK_GetEvaluationAreaCount**Overview:**

Get the number of evaluation areas

Syntax:

CA2DSDK_ER CA2DSDK_GetEvaluationAreaCount(int32_k* pCount)

[Arguments]

pCount	Evaluation area count
--------	-----------------------

[Return values]

CA2D_OK	Completed normally
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_MEASURING	Cannot be processed because the measurement is running
CA2D_ER_RUNNING_DIAGNOSIS	Cannot be processed because the instrument diagnostic is running
CA2D_ER_PARAM	Invalid argument specification

Description:

This function gets the number of set evaluation areas.

For manual layout, you can get the number of evaluation areas that are currently registered.

For automatic layout, the number is 1 until [CA2DSDK_CalculateSpotValue](#) is used.

After [CA2DSDK_CalculateSpotValue](#) is used, you can get the detected number of evaluation areas.

CA2DSDK_GetEvaluationArea**Overview:**

Get the evaluation area

Syntax:

CA2DSDK_ER CA2DSDK_GetEvaluationArea(int num, tagEvaluationArea* pArea)

[Arguments]

num	Evaluation area number (specify from 0)
pArea	Pointer to the evaluation area structure

[Return values]

CA2D_OK	Completed normally
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_MEASURING	Cannot be processed because the measurement is running
CA2D_ER_RUNNING_DIAGNOSIS	Cannot be processed because the instrument diagnostic is running
CA2D_ER_PARAM	Invalid argument specification
CA2D_ER_PARAM_NUM	Invalid evaluation area number specification

Description:

This function gets the evaluation area with the specified number.

For automatic layout, please note that you cannot get the correct count until [CA2DSDK_CalculateSpotValue](#) is used.

CA2DSDK_SetAlignedSpotCondition**Overview:**

Set the spot aligned layout conditions

Syntax:

CA2DSDK_ER CA2DSDK_SetAlignedSpotCondition(const tagAlignedSpotCond* pCond)

[Arguments]

pCond	Pointer to the spot aligned layout structure
-------	--

[Return values]

CA2D_OK	Completed normally
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_MEASURING	Cannot be processed because the measurement is running
CA2D_ER_RUNNING_DIAGNOSIS	Cannot be processed because the instrument diagnostic is running
CA2D_ER_PARAM	Invalid argument specification
CA2D_ER_PARAM_SPOT_COUNT	Invalid layout count specification
CA2D_ER_PARAM_SPOT_SHAPE	Invalid shape specification
CA2D_ER_PARAM_SPOT_SIZE	Invalid size specification
CA2D_ER_PARAM_SPOT_OFFSET_INPUT	Invalid offset input method specification
CA2D_ER_PARAM_SPOT_OFFSET_POS	Invalid offset layout position specification
CA2D_ER_PARAM_SPOT_OFFSET_AREA	Invalid offset margin specification

Description:

This function sets the conditions to align the spots.

The spots are laid out when [CA2DSDK_CalculateSpotValue](#) is used.

The spot results are cleared when the conditions are changed, so you must calculate the spot results by using [CA2DSDK_CalculateSpotValue](#).

The number of spots that can be laid out is limited by the number of evaluation areas.

Evaluation areas: 4 or less	Spot count upper limit: 50 x 50 (2500)
Evaluation areas: 25 or less	Spot count upper limit: 20 x 20 (400)
Evaluation areas: 100 or less	Spot count upper limit: 10 x 10 (100)
Evaluation areas: 144 or less	Spot count upper limit: 5 x 5 (25)

CA2DSDK_GetAlignedSpotCondition**Overview:**

Get the spot aligned layout conditions

Syntax:

CA2DSDK_ER CA2DSDK_GetAlignedSpotCondition(tagAlignedSpotCond* pCond)

[Arguments]

pCond	Pointer to the spot aligned layout structure
-------	--

[Return values]

CA2D_OK	Completed normally
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_MEASURING	Cannot be processed because the measurement is running
CA2D_ER_RUNNING_DIAGNOSIS	Cannot be processed because the instrument diagnostic is running
CA2D_ER_PARAM	Invalid argument specification

Description:

This function gets the conditions to align the spots.

CA2DSDK_CalculateSpotValue**Overview:**

Calculate the spot results.

Syntax:

CA2DSDK_ER CA2DSDK_CalculateSpotValue()

[Arguments]

None

[Return values]

CA2D_OK	Completed normally
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_MEASURING	Cannot be processed because the measurement is running
CA2D_ER_RUNNING_DIAGNOSIS	Cannot be processed because the instrument diagnostic is running
CA2D_ER_DATA_NONE	No measurement data (not measured or measurement error)
CA2D_ER_EVAL_CALC	Could not correctly lay out the evaluation areas
CA2D_ER_SPOT_CALC	Could not correctly lay out the spots

Description:

This function calculates the spot results.

The processing flow for this API is listed below.

- [1] Lay out the evaluation areas (for automatic layout only)
- [2] Lay out the spots
- [3] Calculate the spot results for each evaluation area

To get the calculated results, use [CA2DSDK_GetSpotValue](#).

CA2DSDK_GetSpotValue**Overview:**

Get the spot result

Syntax:

CA2DSDK_ER CA2DSDK_GetSpotValue(int32_k num, int32_k point, tagSpotValue* pResult)

[Arguments]

num	Evaluation area number (specify from 0)
point	Spot number (specify from 0)
pResult	Pointer to the spot result structure

[Return values]

CA2D_OK	Completed normally
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_MEASURING	Cannot be processed because the measurement is running
CA2D_ER_RUNNING_DIAGNOSIS	Cannot be processed because the instrument diagnostic is running
CA2D_ER_DATA_NONE	No data (Recalculation required)
CA2D_ER_PARAM	Invalid argument specification
CA2D_ER_PARAM_NUM	Invalid evaluation area number specification
CA2D_ER_PARAM_POINT	Invalid spot area specification

Description:

This function gets the spot result.

The spot results are not automatically calculated each time a measurement is run. You must always use [CA2DSDK_CalculateSpotValue](#) to calculate the results before getting them.

If this function returns CA2D_ER_DATA_NONE, there may be no measured data or it may not have been calculated with [CA2DSDK_CalculateSpotValue](#).

4.6 Others

CA2DSDK_DiagnosisInstrument

Overview:

Start the instrument diagnostic

Syntax:

CA2DSDK_ER CA2DSDK_DiagnosisInstrument()

[Arguments]

None

[Return values]

CA2D_OK	Completed normally
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_CONNECT	Connection is not complete
CA2D_ER_MEASURING	Cannot be processed because the measurement is running
CA2D_ER_RUNNING_DIAGNOSIS	Cannot be processed because the instrument diagnostic is running
CA2D_ER_FILE_NOTFOUND_CALIBRATION	Calibration data file does not exist
CA2D_ER_FILE_ACCESS	Could not access the file
CA2D_ER_COMMUNICATION	Could not communicate with the instrument
CA2D_ER_COM_FAN	Fan control is not operating correctly
CA2D_ER_COM_MEMORY	Insufficient memory
CA2D_ER_COM_DETECT	The instrument was disconnected
CA2D_ER_USBDLL	USB driver control is not operating correctly

Description:

This function starts the instrument diagnostic.

Before running the instrument diagnostic, always attach the lens cap to the instrument.

Before running the instrument diagnostic, set measurement conditions using [CA2DSDK_SetInstrumentCondition](#).

It returns a return value when the measurement starts.

Check for the completion of the diagnostic and errors during the diagnostic by using [CA2DSDK_PollingDiagnosis](#).

CA2D-SDK Reference Manual**CA2DSDK_PollingDiagnosis****Overview:**

Check the instrument diagnostic

Syntax:

CA2DSDK_ER CA2DSDK_PollingDiagnosis()

[Arguments]

None

[Return values]

CA2D_CANCEL	Processing was canceled
CA2D_OK	Completed normally
CA2D_OK_RUNNING_DIAGNOSIS	The diagnostic is running
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_CONNECT	Connection is not complete
CA2D_ER_MEASURING	Cannot be processed because the measurement is running
CA2D_ER_DIAGNOSIS	Stopped due to an error that occurred during the diagnostic
CA2D_ER_DIAGNOSIS_AREA	There is a problem with the diagnostic results
CA2D_ER_DIAGNOSIS_CAUTION	The diagnostic results have reached the caution level
CA2D_ER_DIAGNOSIS_WARNING	The diagnostic results have reached the warning level
CA2D_ER_COMMUNICATION	Could not communicate with the instrument
CA2D_ER_COM_FILTER	Filter control is not operating correctly
CA2D_ER_COM_STATUS	Get status is not operating correctly
CA2D_ER_COM_MEASURE	Measurement control is not operating correctly
CA2D_ER_COM_SHUTTERSPEED	Shutter speed control is not operating correctly
CA2D_ER_COM_GAIN	Gain control is not operating correctly
CA2D_ER_COM_ADDITIONAL	Cumulative number control is not operating correctly
CA2D_ER_COM_GETDATA	Data acquisition control is not operating correctly
CA2D_ER_COM_MOTOR	Motor control is not operating correctly
CA2D_ER_COM_MEMORY	Insufficient memory
CA2D_ER_COM_DETECT	The instrument was disconnected
CA2D_ER_USBDLL	USB driver control is not operating correctly

Description:

This function checks the state during the diagnostic and during standby.

This API is also used for getting the error that occurred during the diagnostic.

CA2DSDK_StopDiagnosis**Overview:**

Stop the completion of the instrument diagnostic

Syntax:

CA2DSDK_ER CA2DSDK_StopDiagnosis()

[Arguments]

None

[Return values]

CA2D_OK	Completed normally
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_CONNECT	Connection is not complete
CA2D_ER_MEASURING	Cannot be processed because the measurement is running
CA2D_ER_COMMUNICATION	Could not communicate with the instrument
CA2D_ER_COM_MEMORY	Insufficient memory
CA2D_ER_COM_DETECT	The instrument was disconnected
CA2D_ER_USBDLL	USB driver control is not operating correctly

Description:

This function stops the instrument diagnostic.

CA2D-SDK Reference Manual**CA2DSDK_GetFinderImage****Overview:**

Get the finder image

Syntax:

CA2DSDK_ER CA2DSDK_GetFinderImage(int16_km* pData)

[Arguments]

pData	Pointer to the data to get The data is stored from the upper left to the lower right
-------	---

[Return values]

CA2D_OK	Completed normally
CA2D_OK_MEASURING	Measuring
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_CONNECT	Connection is not complete
CA2D_ER_MEASURING	Cannot be processed because the measurement is running
CA2D_ER_RUNNING_DIAGNOSIS	Cannot be processed because the instrument diagnostic is running
CA2D_ER_DATA_NONE	No measurement data (not measured or measurement error)
CA2D_ER_PARAM	Invalid argument specification (pointer argument or other argument)
CA2D_ER_COMMUNICATION	Could not communicate with the instrument
CA2D_ER_COM_FILTER	Filter control is not operating correctly
CA2D_ER_COM_STATUS	Get status is not operating correctly
CA2D_ER_COM_MEASURE	Measurement control is not operating correctly
CA2D_ER_COM_FINDER	Finder measurement is not operating correctly
CA2D_ER_COM_SHUTTERSPEED	Shutter speed control is not operating correctly
CA2D_ER_COM_GAIN	Gain control is not operating correctly
CA2D_ER_COM_ADDITIONAL	Cumulative number control is not operating correctly
CA2D_ER_COM_GETDATA	Data acquisition control is not operating correctly
CA2D_ER_COM_MOTOR	Motor control is not operating correctly
CA2D_ER_COM_MEMORY	Insufficient memory
CA2D_ER_COM_DETECT	The instrument was disconnected
CA2D_ER_USBDLL	USB driver control is not operating correctly

Description:

This function gets the finder image data.
The resolution is 490*490 fixation.

Brightness is able to be adjusted by using the number of "exposureIndex" in [CA2DSDK_SetInstrumentCondition](#).

CA2DSDK_CheckPeriodicalCalibration**Overview:**

Check if periodic calibration is required

Syntax:

int32_km CA2DSDK_CheckPeriodicalCalibration(tagPeriodicalCallInfo* plInfo)

[Arguments]

plInfo	Pointer to the periodic calibration information structure
--------	---

[Return values]

CA2D_OK	Completed normally
CA2D_ER_ENABLE	Initialization is not complete
CA2D_ER_CONNECT	Connection is not complete
CA2D_ER_MEASURING	Cannot be processed because the measurement is running
CA2D_ER_RUNNING_DIAGNOSIS	Cannot be processed because the instrument diagnostic is running
CA2D_ER_PERIODICAL_CAL	Periodic calibration is required
CA2D_ER_PARAM	Invalid argument specification

Description:

This function checks if periodic calibration is required.

If periodic calibration is required, this function returns CA2D_ER_PERIODICAL_CAL. Please contact the dealer where the instrument was purchased.

CA2DSDK_GetSDKVersion

Overview:

Get the SDK version

Syntax:

`int32_km CA2DSDK_GetSDKVersion()`

[Arguments]

None

[Return values]

Version number (for example, this function returns 1000000 for version 1.00.0000)

Description:

This function gets the SDK version.

5. Definitions/Structures

5.1 Type definitions

Definition	Bytes	Description	C/C++	VB.NET	VBA (VB6)
int8_km	1	Signed byte type	char	SByte	(Byte)
uint8_km	1	Byte type	unsigned char	Byte	Byte
int16_km	2	Short integer type	short	Short	Integer
uint16_km	2	Short integer type (unsigned)	unsigned short	UShort	(Integer)
int32_km	4	Integer type	long(int)	Integer	Long
uint32_km	4	Integer type (unsigned)	unsigned long	UInteger	(Long)
int64_km	8	Long integer type	long long	Long	
uint64_km	8	Long integer type (unsigned)	unsigned long long	ULong	
float_km	4	Single precision floating point type	float	Single	Single
double_km	8	Double precision floating point type	double	Double	Double
CA2DSDK_ER	4	int32_km			

5.2 Structure definitions

Measurement conditions structure [tagInstrumentCond]

```
typedef struct tagInstrumentCond{
    int16_km          lensType;
    int16_km          lensPosition;
    int16_km          exposureMode;
    int16_km          measurementType;
    double_km         syncValue;
    int16_km          exposureIndex;
    int16_km          additional;
    int16_km          left;
    int16_km          top;
    int16_km          right;
    int16_km          bottom;
    int16_km          filterMeasure;
    int16_km          filterIndex;
    int16_km          smearIndex;
    int16_km          userCal;
    int16_km          rotate;
}
```

Parameter:

meter.

lensType	Lens type					
	Value	Definition		Description		
	0	LENS_NORMAL		Normal lens		
	1	LENS_WIDE		Wide lens		
	2	LENS_TELE		Telephoto lens		
	3	LENS_MACRO1		Macro 1 (low magnification)		
	4	LENS_MACRO2		Macro 2 (high magnification)		
lensPosition	Lens position					
	Value	Normal	Wide	Telephoto	Macro 1	Macro 2
	0	0.25 m	0.20 m	0.90 m	0.50 m	0.30 m
	1	0.25 m + 1/2	0.24 m	0.90 m + 1/2		
	2	0.30 m	0.30 m	1.00 m		
	3	0.30 m + 1/2	0.50 m	1.00 m + 1/3		
	4	0.50 m	1.00 m	1.00 m + 2/3		
	5	0.50 m + 1/2	∞	1.50 m		
	6	1.00 m		1.50 m + 1/3		
	7	1.00 m + 1/2		1.50 m + 2/3		
	8	∞		3.00 m		
	9			3.00 m + 1/3		
	10			3.00 m + 2/3		
	11			∞		
exposureMode	Exposure mode					
	Value	Definition		Description		
	0	EXPOSURE_MANUAL		Manual exposure		
	1	EXPOSURE_AUTO		Auto exposure		
	2	EXPOSURE_MULTI		Multiple exposure		
For manual exposure, measurements are taken with the exposure time set by exposureIndex.						
For auto exposure and multiple exposure, the exposureIndex value is ignored.						

⋮

CA2D-SDK Reference Manual

measurementType	Measurement type		
	Value	Definition	Description
	0	SYNCMODE_OFF	Non-synchronized measurements
	1	SYNCMODE_ON	Synchronized measurements
	For synchronized measurements, the exposure time is calculated using the frequency set for syncValue and exposureIndex.		
syncValue	Synchronized measurement frequency (4.0000 <= configurable range <= 2000.0000) This value can be set up to four digits after the decimal point, the fifth and subsequent digits are discarded		
exposureIndex	Exposure time table		
	Value	Non-synchronized measurements	
		Time	ND Gain
	0	1/2048	1.5% normal
	1	1/1024	1.5% normal
	2	1/512	1.5% normal
	3	1/256	1.5% normal
	4	1/128	1.5% normal
	5	1/64	1.5% normal
	6	1/64	3.0% normal
	7	1/64	6.0% normal
	8	1/64	12.5% normal
	9	1/64	25.0% normal
	10	1/64	50.0% normal
	11	1/64	100.0% normal
	12	1/32	100.0% normal
	13	1/16	100.0% normal
	14	1/8	100.0% normal
	15	1/4	100.0% normal
	16	1/2	100.0% normal
	17	1/1	100.0% normal
	18	2/1	100.0% normal
	* Please note that for synchronized measurements, the exposure time changes according to the set frequency. The table numbers also vary according to the frequency setting. We recommend first checking the table count in the application before specifying this value.		
additional	Cumulative number (1 <= configurable range <=256)		
left top right bottom	Exposure setting area (0 <= configurable range <= 979) left: left coordinate, top: top coordinate, right: right coordinate, bottom: bottom coordinate Set these members so that left < right and top < bottom holds true		
filterMeasure	On/off for individual X, Y, Z measurements		
	Value	Definition	Description
	0	FILTER_MEASURE_OFF	Off
	1	FILTER_MEASURE_ON	On
filterIndex	Filter index number Valid when filterMeasure is on		
	Value	Definition	Description
	0	FILTER_INDEX_X	X
	1	FILTER_INDEX_Y	Y
	2	FILTER_INDEX_Z	Z

⋮

CA2D-SDK Reference Manual

smearIndex	Smear compensation		
	Value	Definition	Description
	0	SMEAR_NONE	None
	1	SMEAR_SIMPLE	Simple compensation
	2	SMEAR_DETAIL	Approximate compensation
userCal	Apply user calibration		
	Value	Definition	Description
	0	USERCAL_OFF	User calibration off
	1	USERCAL_ON	User calibration on
	When on, data must be set.		
rotate	Image orientation		
	Value	Definition	Description
	0	ROTATION_NONE	Output unmodified
	1	ROTATION_RIGHT	Rotate clockwise 90°
	2	ROTATION_INVERSION	Rotate clockwise 180°
	3	ROTATION_LEFT	Rotate clockwise 270°

User calibration data structure [tagUserCalData]

```
typedef struct tagUserCalData{
    int16_km      calType;
    double_km     R[3][2];
    double_km     G[3][2];
    double_km     B[3][2];
    double_km     W[3][2];
}
```

Parameter:

calType	Calibration type		
	Value	Definition	Description
	0	CALTYPE_NONE	No calibration
	1	CALTYPE_NORMAL	One-color calibration
	2	CALTYPE_RGB	RGB calibration
	3	CALTYPE_WRGB	WRGB calibration
R[3][2]	Red data (used for RGB calibration and WRGB calibration) [3]: Lv, x · y order [2]: Pre-calibration, post-calibration order		
G[3][2]	Green data (used for RGB calibration and WRGB calibration) [3]: Lv, x · y order [2]: Pre-calibration, post-calibration order		
B[3][2]	Blue data (used for RGB calibration and WRGB calibration) [3]: Lv, x · y order [2]: Pre-calibration, post-calibration order		
W[3][2]	White data (used for one-color calibration, RGB calibration, and WRGB calibration) [3]: Lv, x · y order [2]: Pre-calibration, post-calibration order		

Data acquisition conditions structure [tagGetDataParam]

```
typedef struct tagGetDataParam{
    int16_km      left;
    int16_km      top;
    int16_km      right;
    int16_km      bottom;
}
```

Parameter:

left	Data acquisition area (0 <= configurable range <= 979)
top	left: left coordinate, top: top coordinate, right: right coordinate, bottom: bottom coordinate
right	Set these members so that left < right and top < bottom holds true
bottom	

Data configuration structure [tagDataCond]

```
typedef struct tagDataCond{
    int16_km      valueType;
    int16_km      resolution;
    int16_km      lower_enable;
    int16_km      lower_item;
    double_km     lower_threshold;
}
```

Parameter:

valueType	<div>Color specification value to calculate</div> <div>The color specification value to get is also determined by this member.</div> <table><tr><th>Value</th><th>Definition</th><th>Description</th></tr><tr><td>0</td><td>VALTYPE_X</td><td>X</td></tr><tr><td>1</td><td>VALTYPE_Y</td><td>Y</td></tr><tr><td>2</td><td>VALTYPE_Z</td><td>Z</td></tr><tr><td>3</td><td>VALTYPE_LV</td><td>Lv</td></tr><tr><td>4</td><td>VALTYPE_SX</td><td>x</td></tr><tr><td>5</td><td>VALTYPE_SY</td><td>y</td></tr><tr><td>6</td><td>VALTYPE_UD</td><td>u'</td></tr><tr><td>7</td><td>VALTYPE_VD</td><td>v'</td></tr><tr><td>8</td><td>VALTYPE_TCP</td><td>Correlated color temperature Tcp</td></tr><tr><td>9</td><td>VALTYPE_DUV</td><td>Δuv</td></tr><tr><td>10</td><td>VALTYPE_TCP_JIS</td><td>Correlated color temperature Tcp (JIS)</td></tr><tr><td>11</td><td>VALTYPE_DUV_JIS</td><td>Δuv (JIS)</td></tr><tr><td>12</td><td>VALTYPE_DWL</td><td>Dominant wavelength</td></tr><tr><td>13</td><td>VALTYPE_PL</td><td>Excitation purity</td></tr></table>	Value	Definition	Description	0	VALTYPE_X	X	1	VALTYPE_Y	Y	2	VALTYPE_Z	Z	3	VALTYPE_LV	Lv	4	VALTYPE_SX	x	5	VALTYPE_SY	y	6	VALTYPE_UD	u'	7	VALTYPE_VD	v'	8	VALTYPE_TCP	Correlated color temperature Tcp	9	VALTYPE_DUV	Δuv	10	VALTYPE_TCP_JIS	Correlated color temperature Tcp (JIS)	11	VALTYPE_DUV_JIS	Δuv (JIS)	12	VALTYPE_DWL	Dominant wavelength	13	VALTYPE_PL	Excitation purity
Value	Definition	Description																																												
0	VALTYPE_X	X																																												
1	VALTYPE_Y	Y																																												
2	VALTYPE_Z	Z																																												
3	VALTYPE_LV	Lv																																												
4	VALTYPE_SX	x																																												
5	VALTYPE_SY	y																																												
6	VALTYPE_UD	u'																																												
7	VALTYPE_VD	v'																																												
8	VALTYPE_TCP	Correlated color temperature Tcp																																												
9	VALTYPE_DUV	Δuv																																												
10	VALTYPE_TCP_JIS	Correlated color temperature Tcp (JIS)																																												
11	VALTYPE_DUV_JIS	Δuv (JIS)																																												
12	VALTYPE_DWL	Dominant wavelength																																												
13	VALTYPE_PL	Excitation purity																																												
resolution	<div>Resolution</div> <table><tr><th>Value</th><th>Definition</th><th>Description</th></tr><tr><td>0</td><td>RESOLUTION_980</td><td>980*980</td></tr><tr><td>1</td><td>RESOLUTION_490</td><td>490*490</td></tr><tr><td>2</td><td>RESOLUTION_196</td><td>196*196</td></tr></table>	Value	Definition	Description	0	RESOLUTION_980	980*980	1	RESOLUTION_490	490*490	2	RESOLUTION_196	196*196																																	
Value	Definition	Description																																												
0	RESOLUTION_980	980*980																																												
1	RESOLUTION_490	490*490																																												
2	RESOLUTION_196	196*196																																												
lower_enable	<div>Turn on/off under errors</div> <div>* Even when this setting is off, under errors may occur when the measurement subject includes light/dark</div>																																													
lower_item	<table><tr><th>Value</th><th>Definition</th><th>Description</th></tr><tr><td>0</td><td>VALTYPE_X</td><td>X</td></tr><tr><td>1</td><td>VALTYPE_Y</td><td>Y</td></tr><tr><td>2</td><td>VALTYPE_Z</td><td>Z</td></tr></table>	Value	Definition	Description	0	VALTYPE_X	X	1	VALTYPE_Y	Y	2	VALTYPE_Z	Z																																	
Value	Definition	Description																																												
0	VALTYPE_X	X																																												
1	VALTYPE_Y	Y																																												
2	VALTYPE_Z	Z																																												
lower_threshold	<div>The threshold value to process as an under error</div> <div>Set this member as a relative ratio from the maximum value [0.0 <= <= 100.0]</div> <div>* If 0.0 is specified, only the minimum value is an under error</div>																																													

Evaluation area structure [tagEvaluationArea]

```
typedef struct tagEvaluationArea{
    int16_km      left;
    int16_km      top;
    int16_km      right;
    int16_km      bottom;
}
```

Parameter:

left	Evaluation area (0 <= configurable range <= 979)
top	left: left coordinate, top: top coordinate, right: right coordinate, bottom: bottom coordinate
right	Set these members so that left < right and top < bottom holds true
bottom	

Evaluation area layout conditions structure [tagEvaluationCond]

```
typedef struct tagEvaluationCond{
    int16_km      type;
    int16_km      left;
    int16_km      top;
    int16_km      right;
    int16_km      bottom;
    int16_km      row;
    int16_km      col;
    double_km     thresholdValue
}
```

Parameter:

type	Layout method		
	Value	Definition	Description
	0	EXTTYPE_MANU	Manual layout
	1	EXTTYPE_AUTO	Automatic layout
left top right bottom	Area to detect (0 <= configurable range <= 979) left: left coordinate, top: top coordinate, right: right coordinate, bottom: bottom coordinate Set these members so that left < right and top < bottom holds true * Use when automatic layout		
row	Vertical layout count [1 <= <= 12] * Use when automatic layout		
col	Horizontal layout count [1 <= <= 12] * Use when automatic layout		
thresholdValue	Threshold value for detection [0.0 <= <= 100.0] * Use when automatic layout		

Spot layout structure [tagAlignedSpotCond]

```
typedef struct tagAlignedSpotCond{
    int16_km      row;
    int16_km      col;
    int16_km      shape;
    float_km      height;
    float_km      width;
    int16_km      offset_input;
    int16_km      offset_position;
    float_km      offset_left;
    float_km      offset_top;
    float_km      offset_right;
    float_km      offset_bottom;
}
```

Parameter:

row	Vertical layout count [1 <= <= 50]		
col	Horizontal layout count [1 <= <= 50]		
shape	Shape		
	Value	Definition	Description
	0	SPOT_SHAPETYPE_CIRCLE	Circle
	1	SPOT_SHAPETYPE_RECT	Rectangle
height	Rectangle height (or circle diameter)		
width	Rectangle width (not used for circle)		
offset_input	Offset input method		
	Value	Definition	Description
	0	SPOT_OFFSET_ABSOLUTE	Absolute values
	1	SPOT_OFFSET_RELATIVE	Relative values
offset_position	Offset position		
	Value	Definition	Description
	0	SPOT_OFFSET_CORNER	Set spot edge as edge
	1	SPOT_OFFSET_CENTER	Set spot center as edge
offset_left offset_top offset_right offset_bottom	<p>When absolute value input Offset margin (0 <= configurable range <= 979) left: Left margin top: Top margin right: Right margin bottom: Bottom margin</p> <p>When relative value input Offset margin (2 <= configurable range <= 100) left: Left margin ratio top: Top margin ratio right: Right margin ratio bottom: Bottom margin ratio</p> <p>The input value is used as the denominator, and the margin is determined as a ratio of the evaluation area length. Taking the evaluation area width as H and height as V, the margin is found with the equations below</p> <p>Left margin: H/left, right margin: H/right Top margin: V/top, bottom margin: V/bottom</p> <p>For example, when the evaluation area width is 100 and left=10, right=20 are specified, the left margin is 10 and the right margin is 5.</p>		

CA2D-SDK Reference Manual**Spot result structure [tagSpotValue]**

```
typedef struct tagSpotValue{
    int16_km          color;
    float_km          result[3];
}
```

Parameter:

color	Color values to get		
result	Color values		
	Value	Definition	Description
	0	VALTYPE_X_Y_Z	$X \cdot Y \cdot Z$
	1	VALTYPE_LV_SX_SY	$L_v \cdot x \cdot y$
	2	VALTYPE_Y_UD_VD	$Y \cdot u' \cdot v'$
	3	VALTYPE_Y_TCP_DUV	$Y \cdot T_{cp} \cdot \Delta uv$
	4	VALTYPE_Y_TCP_DUV_JIS	$Y \cdot T_{cp}(JIS) \cdot \Delta uv(JIS)$
	5	VALTYPE_Y_DWL_PURITY	$Y \cdot \text{dominant wavelength} \cdot \text{excitation purity}$

Periodic calibration information structure [tagPeriodicalCallInfo]

```
typedef struct tagPeriodicalCallInfo{  
    int32_km      year;  
    int32_km      month;  
    int32_km      day;  
    int32_km      interval;  
}
```

Parameter:

year	Periodic calibration starting date - Year
month	Periodic calibration starting date - Month
day	Periodic calibration starting date - Day
interval	Periodic calibration interval (month)

CA2D-SDK Reference Manual**6. Error Codes**

Error ID	Value	Description	Action
CA2D_CANCEL	0	Description	Processing was canceled
		Action	
CA2D_OK	1	Description	Processing completed normally
		Action	
CA2D_OK_MEASURING	10	Description	Measuring
		Action	
CA2D_OK_RUNNING_DIAGNOSIS	11	Description	Running self-diagnostic
		Action	
CA2D_ER	-1	Description	A problem occurred in the SDK
		Action	Restart the instrument(s) and the SDK
CA2D_ER_ENABLE	-1000	Description	Initialization is not complete (Initialization failed)
		Action	When using the SDK, you must first initialize it.
CA2D_ER_CONNECT	-1001	Description	Connection is not complete (Connection failed)
		Action	Check the following items <ul style="list-style-type: none"> • Is the instrument's power turned on? • Are the PC and the USB cable connected?
CA2D_ER_INSTRUMENT	-1050	Description	The instrument information is incorrect
		Action	If this error is output many times, consult with the dealer where the instrument was purchased.
CA2D_ER_MEASURING	-1100	Description	Cannot be processed because the measurement is running
		Action	Process after waiting for the measurement to complete
CA2D_ER_RUNNING_FINDER	-1101	Description	Processing is not possible because the finder is running
		Action	Process after waiting for the finder to complete running
CA2D_ER_RUNNING_DIAGNOSIS	-1102	Description	Cannot be processed because the instrument diagnostic is running
		Action	Process after waiting for the instrument diagnostic to complete
CA2D_ER_DATA_NONE	-1150	Description	No measurement data (not measured or measurement error)
		Action	
CA2D_ER_EXPOSURE	-1200	Description	Not an appropriate exposure time
		Action	There are overexposed pixels, so we recommend changing the shutter speed
CA2D_ER_AUTOEXPO_OVER	-1201	Description	Failed to calculate the exposure time (overexposed)
		Action	Could not measure with automatic exposure, so measure with manual exposure
CA2D_ER_AUTOEXPO_UNDER	-1202	Description	Failed to calculate the exposure time (underexposed)
		Action	Could not measure with automatic exposure, so measure with manual exposure
CA2D_ER_TEMPERATURE	-1250	Description	Temperature error
		Action	Restart the instrument(s) and the SDK If this error is output many times, consult with the dealer where the instrument was purchased.
CA2D_ER_USERDATA_NONE	-1300	Description	User calibration data has not been set
		Action	Set the user calibration data

⋮

CA2D-SDK Reference Manual

Error ID	Value		
CA2D_ER_USERDATA_CALC	-1301	Description	The user calibration data is invalid (calibration factor could not be calculated)
		Action	Check the user calibration data content
CA2D_ER_PERIODICAL_CAL	-1350	Description	Periodic calibration is now required
		Action	Contact the dealer where the instrument was purchased
CA2D_ER_DIAGNOSIS	-1400	Description	Stopped due to an error that occurred during the diagnostic
		Action	Restart the instrument(s) and the SDK and run the diagnostic again If this error is output many times, consult with the dealer where the instrument was purchased.
CA2D_ER_DIAGNOSIS_AREA	-1401	Description	There is a problem with the diagnostic results
		Action	Contact the dealer where the instrument was purchased
CA2D_ER_DIAGNOSIS_CAUTION	-1402	Description	The diagnostic results have reached the caution level
		Action	When the warning level is reached, please be aware that instrument performance can no longer be maintained
CA2D_ER_DIAGNOSIS_WARNING	-1403	Description	The diagnostic results have reached the warning level
		Action	The measurement accuracy may not satisfy the specifications, so contact the dealer where the instrument was purchased.
CA2D_ER_FILE_NOTFOUND_LENSTYPE	-1500	Description	The lens type file does not exist
		Action	Store the file in the specified location The file is included in the SDK
CA2D_ER_FILE_NOTFOUND_CALIBRATION	-1501	Description	Calibration data file does not exist
		Action	Store the file in the specified location It is located on the DVD included with the product
CA2D_ER_FILE_FORMAT	-1502	Description	Invalid file format
		Action	Use the correct file
CA2D_ER_FILE_ACCESS	-1503	Description	Could not access the file
		Action	Check the following items • The file is located in a location without access rights • The file is read-only
CA2D_ER_PARAM	-1600	Description	Invalid argument specification
		Action	Check the content of the argument
CA2D_ER_PARAM_INDEX	-1610	Description	Invalid instrument specification (the specified instrument does not exist)
		Action	Check the content of the argument
CA2D_ER_PARAM_LENSTYPE	-1620	Description	Invalid lens type specification (lens cannot be used)
		Action	Check the content of the argument
CA2D_ER_PARAM_LENSPOS	-1621	Description	Invalid focus ring distance indicator specification
		Action	Check the content of the argument
CA2D_ER_PARAM_EXPOSURE_MODE	-1622	Description	Invalid exposure mode specification
		Action	Check the content of the argument
CA2D_ER_PARAM_EXPOSURE_INDEX	-1623	Description	Invalid exposure table index number specification
		Action	Check the content of the argument

⋮

CA2D-SDK Reference Manual

Error ID	Value		
CA2D_ER_PARAM_EXPOSURE_AREA	-1624	Description	Invalid exposure setting area setting
		Action	Check the content of the argument
CA2D_ER_PARAM_SYNC_MODE	-1625	Description	Invalid synchronized measurement mode specification
		Action	Check the content of the argument
CA2D_ER_PARAM_SYNC_VALUE	-1626	Description	Invalid synchronization frequency specification
		Action	Check the content of the argument
CA2D_ER_PARAM_ADDITIONAL	-1627	Description	Invalid cumulative number specification
		Action	Check the content of the argument
CA2D_ER_PARAM_FILTER	-1628	Description	Invalid filter measurement specification
		Action	Check the content of the argument
CA2D_ER_PARAM_FILTER_INDEX	-1629	Description	Invalid filter type specification
		Action	Check the content of the argument
CA2D_ER_PARAM_SMEAR	-1630	Description	Invalid smear compensation specification
		Action	Check the content of the argument
CA2D_ER_PARAM_USERCAL	-1631	Description	Invalid user calibration specification
		Action	Check the content of the argument
CA2D_ER_PARAM_ROTATION	-1632	Description	Invalid image orientation specification
		Action	Check the content of the argument
CA2D_ER_PARAM_FILE	-1640	Description	Invalid file path (folder path)
		Action	Check the content of the argument
CA2D_ER_PARAM_CALTYPE	-1650	Description	Invalid calibration type specification
		Action	Check the content of the argument
CA2D_ER_PARAM_GETAREA	-1660	Description	Invalid data acquisition area specification
		Action	Check the content of the argument
CA2D_ER_PARAM_VALTYPE	-1670	Description	Invalid color specification value specification
		Action	Check the content of the argument
CA2D_ER_PARAM_RESOLUTION	-1671	Description	Invalid resolution specification
		Action	Check the content of the argument
CA2D_ER_PARAM_LOWER_LEVEL	-1672	Description	Invalid threshold value specification to process as an under error
		Action	Check the content of the argument
CA2D_ER_PARAM_EVAL_TYPE	-1680	Description	Invalid layout type specification
		Action	Check the content of the argument
CA2D_ER_PARAM_EVAL_AREA	-1681	Description	Invalid area specification
		Action	Check the content of the argument
CA2D_ER_PARAM_EVAL_COUNT	-1682	Description	Invalid detection count specification
		Action	Check the content of the argument
CA2D_ER_PARAM_EVAL_LEVEL	-1683	Description	Invalid detection threshold value specification
		Action	Check the content of the argument
CA2D_ER_PARAM_SPOT_COUNT	-1690	Description	Invalid layout count specification
		Action	Check the content of the argument
CA2D_ER_PARAM_SPOT_SHAPE	-1691	Description	Invalid shape specification
		Action	Check the content of the argument
CA2D_ER_PARAM_SPOT_SIZE	-1692	Description	Invalid size specification
		Action	Check the content of the argument
CA2D_ER_PARAM_SPOT_OFFSET_INPUT	-1693	Description	Invalid offset input method specification
		Action	Check the content of the argument
CA2D_ER_PARAM_SPOT_OFFSET_POS	-1694	Description	Invalid offset layout position specification
		Action	Check the content of the argument
CA2D_ER_PARAM_SPOT_OFFSET_AREA	-1695	Description	Invalid offset margin specification
		Action	Check the content of the argument
CA2D_ER_PARAM_NUM	-1700	Description	Invalid evaluation area number specification
		Action	Check the content of the argument

⋮

CA2D-SDK Reference Manual

Error ID	Value		
CA2D_ER_PARAM_POINT	-1701	Description	Invalid spot area specification
		Action	Check the content of the argument
CA2D_ER_EVAL_CALC	-1710	Description	Could not correctly lay out the evaluation areas
		Action	Check the content of the argument
CA2D_ER_SPOT_CALC	-1711	Description	Could not correctly lay out the spots
		Action	Check the content of the argument
CA2D_ER_MEASURE_CALC	-1800	Description	Measurement calculation failed
		Action	
CA2D_ER_COMMUNICATION	-5000	Description	Could not communicate with the instrument
		Action	Restart the instrument(s) and the SDK and run the diagnostic again
CA2D_ER_COM_FILTER	-5010	Description	Filter control is not operating correctly
		Action	Restart the instrument(s) and the SDK and run the diagnostic again
CA2D_ER_COM_STATUS	-5020	Description	Get status is not operating correctly
		Action	Restart the instrument(s) and the SDK and run the diagnostic again
CA2D_ER_COM_MEASURE	-5030	Description	Measurement control is not operating correctly
		Action	Restart the instrument(s) and the SDK and run the diagnostic again
CA2D_ER_COM_FINDER	-5040	Description	Finder measurement is not operating correctly
		Action	Restart the instrument(s) and the SDK and run the diagnostic again
CA2D_ER_COM_SHUTTERSPEED	-5050	Description	Shutter speed control is not operating correctly
		Action	Restart the instrument(s) and the SDK and run the diagnostic again
CA2D_ER_COM_GAIN	-5060	Description	Gain control is not operating correctly
		Action	Restart the instrument(s) and the SDK and run the diagnostic again
CA2D_ER_COM_ADDITIONAL	-5070	Description	Cumulative number control is not operating correctly
		Action	Restart the instrument(s) and the SDK and run the diagnostic again
CA2D_ER_COM_GETDATA	-5110	Description	Data acquisition control is not operating correctly
		Action	Restart the instrument(s) and the SDK and run the diagnostic again
CA2D_ER_COM_TEMPERATURE	-5120	Description	Temperature control is not operating correctly
		Action	Restart the instrument(s) and the SDK and run the diagnostic again
CA2D_ER_COM_FAN	-5130	Description	Fan control is not operating correctly
		Action	Restart the instrument(s) and the SDK and run the diagnostic again
CA2D_ER_COM_MOTOR	-5140	Description	Motor control is not operating correctly
		Action	Restart the instrument(s) and the SDK and run the diagnostic again
CA2D_ER_COM_MEMORY	-5150	Description	Insufficient memory
		Action	Restart the instrument(s) and the SDK and run the diagnostic again
CA2D_ER_COM_DETECT	-5200	Description	The instrument was disconnected
		Action	Restart the instrument(s) and the SDK and run the diagnostic again
CA2D_ER_USBDLL	-5300	Description	USB driver control is not operating correctly
		Action	Restart the instrument(s) and the SDK and run the diagnostic again



KONICA MINOLTA