CM-MISDK (ANSI C Version)

Reference Manual

[Rev.1.02]

CM-MISDK (ANSI C Version) Reference Manual

• Official application names used in this manual

Abbreviated in this manual as:	Official name
Windows 7	Microsoft® Windows® 7 Business Operating System
Windows 8.1	Microsoft® Windows® 8.1
Windows 10	Microsoft® Windows® 10
C#	Microsoft® Visual C#
C++	Microsoft® Visual C++
VB	Microsoft® Visual Basic .NET

• Trademarks

• Notes on this manual

CM-MISDK (ANSI C Version) Reference Manual

Contents

CM-MISDK (ANSI C Version) Reference Manual

Introduction

The SDK is a tool for developing PC applications for instruments that measure object color. This manual describes how to use the SDK.

1. System Environment

The following table lists the verified development environments.

Supported operating systems	Windows 7(x86), Windows 7(x64) Windows 8.1(x86), Windows 8.1(x64) Windows 10(x86), Windows 10(x64)
Development environment	Visual Studio 2012 Visual Studio 2013 Visual Studio 2015 Visual Studio 2017
Development languages	VC++, VC#, VB.NET
Controllable instruments	• CM-M6 • CM-25cG • CM-26dG • CM-26d • CM-25d • CM-23d

2. Installing/Uninstalling the SDK

2.1. Installation

Install this SDK according to the following procedure.

(1) Expand the contents of the provided "cm-misdk_verXXXrX.zip" file to the desired location on the PC.

(2) "cm-misdk_verXXXrX" will be displayed and the following folders will be present when the files are expanded.

No.	Folder name	Overview
1	SDK	CM-MISDK files
2	Manual	CM-MISDK Reference Manual
3	Driver	USB driver for instrument
4	SampleCode	Sample code files
5	License	License agreement

(3) To develop applications using the SDK, configure the appropriate settings so that the following files in the above SDK folder can be accessed from the development environment. For details, refer to "3. SDK Overview".

No.	Folder name	Overview
1	CMMISDK_x86.dll	SDK DLL file (32-bit version)
2	CMMISDK_x86.lib	VC++ import library file (32-bit version)
3	CMMISDK_x64.dll	SDK DLL file (64-bit version)
4	CMMISDK_x64.lib	VC++ import library file (64-bit version)
5	CMMISDK.NET.dll	C# version DLL file
6	CMMISDK.h	Definitions file
7	CMMISDK_Api.h	API definitions file
8	CMMISDK_Error.h	Error definitions file
9	CMMISDK_Parameters.h	Parameter and structure definitions file
10	CMMISDK_TypeDefine.h	Type definitions file
11	TypeDefine.h	Type definitions file

2.2. Uninstallation

Delete the cm-misdk_verXXXrX folder and manually copied folders.

3. SDK Overview

3.1 Function list

The following processing can be performed with the SDK.

Connect/disconnect		
	CMMISDK_Connect	Connects to the instrument.
	CMMISDK_Disconnect	Ends the connection with the instrument.
	CMMISDK_GetInstrumentInfo	Obtains instrument information.
	CMMISDK_GetSDKVersion	Obtains the SDK version.
	CMMISDK_GetWarning	Obtains the warning status.
Calibrate/measure		
	CMMISDK_GetCalibrationStatus	Obtains the calibration status.
	CMMISDK_PerformZeroCalibration	Executes zero calibration.
	CMMISDK_PerformWhiteCalibration	Executes white calibration.
	CMMISDK_PerformGlossCalibration	Executes gloss calibration.
	CMMISDK_PerformUserCalibration	Executes user calibration.
	CMMISDK_PerformMeasurement	Executes the measurement.
	CMMISDK_PollingMeasurement	Judges if the measurement is complete.
	CMMISDK_CancelMeasurement	Cancels measurement.
	CMMISDK_ReadLatestData	Reads the latest data.
	CMMISDK_LoadLaterstData	Loads the latest data onto the PC.
	CMMISDK_GetLatestData	Obtains the latest data on the PC.
	CMMISDK_SetWhiteCalibrationData	Sets the white calibration plate data.
	CMMISDK_GetWhiteCalibrationData	Obtains the white calibration plate data.
	CMMISDK_SetGlossCalibrationData	Sets the gloss calibration plate data.
	CMMISDK_GetGlossCalibrationData	Obtains the gloss calibration plate data.
	CMMISDK_SetUserCalibrationData	Sets the user calibration data.
	CMMISDK_GetUserCalibrationData	Obtains the user calibration data.
	CMMISDK_SetUserCalibrationEnable	Enables and disables user calibration.
	CMMISDK_GetUserCalibrationEnable	Obtains the user calibration enabled or disabled state.
	CMMISDK_SetTriggerMode	Sets the trigger mode.
	CMMISDK_GetTriggerMode	Obtains the trigger mode.
	CMMISDK_ClearTriggerData	Clears the trigger mode data.
	CMMISDK_IsTriggerData	Obtains the availability of trigger mode data.
	CMMISDK_GetZeroCalibrationDate	Obtains the zero calibration date and time.
	CMMISDK_GetWhiteCalibrationDate	Obtains the white calibration date and time.
	CMMISDK_GetGlossCalibrationDate	Obtains the gloss calibration date and time.
	CMMISDK_GetUserCalibrationDate	Obtains the user calibration date and time.
	CMMISDK_ClearUvAdjustInfo	Clears various data for fluorescence adjustment.
	CMMISDK_SetProfileForUvAdjust	Sets the fluorescence adjustment profile.
	CMMISDK_GetProfileForUvAdjust	Obtains the fluorescence adjustment profile.
	CMMISDK_SetWiForUvAdjust	Sets the WI for fluorescence adjustment.
	CMMISDK_GetWiForUvAdjust	Obtains the WI for fluorescence adjustment.
	CMMISDK_SetTintForUvAdjust	Sets the Tint for fluorescence adjustment.
	CMMISDK_GetTintForUvAdjust	Obtains the Tint for fluorescence adjustment.
	CMMISDK_SetIsoBrightnessForUvAdjust	Sets the ISO brightness for fluorescence adjustment.
	CMMISDK_GetIsoBrightnessForUvAdjust	Obtains the ISO brightness for fluorescence adjustment.
	CMMISDK_SetGanzForUvAdjust	Sets the Ganz & Griesser for fluorescence adjustment.

	CMMISDK_GetGanzForUvAdjust	Obtains the Ganz & Griesser for fluorescence adjustment.
	CMMISDK_SetDataForUvAdjust	Sets the data for fluorescence adjustment.
	CMMISDK_GetDataForUvAdjust	Obtains the data for fluorescence adjustment.
	CMMISDK_PerformUvAdjust	Executes fluorescence adjustment and sets the coefficient.
	CMMISDK_PerformUvAdjustUsingData	Executes fluorescence adjustment and sets the coefficient.
	CMMISDK_ClearCoefForUvAdjust	Clears the fluorescence adjustment coefficient in the instrument.
	CMMISDK_SetCoefForUvAdjust	Sets the fluorescence coefficient.
	CMMISDK_GetCoefForUvAdjust	Obtains the fluorescence coefficient.
Measurement Conditions		
•	CMMISDK_SetMeasurementArea	Sets the measurement area.
•	CMMISDK_GetMeasurementArea	Obtains the measurement area.
•	CMMISDK_SetMeasurementType	Sets the measurement type.
•	CMMISDK_GetMeasurementType	Obtains the measurement type.
•	CMMISDK_SetMeasurementAngle	Sets the measurement angle.
•	CMMISDK_GetMeasurementAngle	Obtains the measurement angle.
•	CMMISDK_SetTiltDetection	Sets tilt detection.
•	CMMISDK_GetTiltDetection	Obtains tilt detection.
•	CMMISDK_SetMeasurementMode	Sets the measurement mode.
•	CMMISDK_GetMeasurementMode	Obtains the measurement mode.
•	CMMISDK_SetSpecularComponent	Sets the specular component.
•	CMMISDK_GetSpecularComponent	Obtains the specular component.
•	CMMISDK_SetUv	Sets the UV condition.
•	CMMISDK_GetUv	Obtains the UV condition.
•	CMMISDK_SetAutoAverageTimes	Sets the number of times of automatic averaging.
•	CMMISDK_GetAutoAverageTimes	Obtains the number of times of automatic averaging.
	CMMISDK_SetManualAverageTimes	Sets the number of times of manual averaging.
	CMMISDK_GetManualAverageTimes	Obtains the number of times of manual averaging.
	CMMISDK_SetManualAverageSaveMode	Sets the manual averaging save method.
	CMMISDK_GetManualAverageSaveMode	Obtains the manual averaging save method.
	CMMISDK_SetCondSMC	Sets the SMC conditions.
	CMMISDK_GetCondSMC	Obtains the SMC conditions.
Display conditions		
	CMMISDK_SetDisplayType	Sets the display type.
	CMMISDK_GetDisplayType	Obtains the display type.
	CMMISDK_SetObserverAndIlluminant	Sets the observation field and illuminant.
	CMMISDK_GetObserverAndIlluminant	Obtains the observation field and illuminant.
	CMMISDK_SetUserIlluminant	Sets the user illuminant data.
	CMMISDK_GetUserIlluminant	Obtains the user illuminant data.
	CMMISDK_SetColorSpace	Sets the color space.
	CMMISDK_GetColorSpace	Obtains the color space.
	CMMISDK_SetEquation	Sets the color difference equation.
	CMMISDK_GetEquation	Obtains the color difference equation.
	CMMISDK_SetCustomIndex	Sets the custom items.
	CMMISDK_GetCustomIndex	Obtains the custom items.
	CMMISDK_SetDirection	Sets the irradiation direction to display.
	CMMISDK_GetDirection	Obtains the irradiation direction to display.

	CMMISDK_SetUserEquation	Sets the user index.
	CMMISDK_GetUserEquation	Obtains the user index.
Data		
	CMMISDK_SetActiveTarget	Sets the active target.
	CMMISDK_GetActiveTarget	Obtains the active target.
	CMMISDK_GetSavedTargetList	Obtains the list of saved target numbers.
	CMMISDK_GetTargetListInFilter	Obtains the list of target numbers when the display filter is applied.
	CMMISDK_DeleteTargetData	Deletes the target data.
	CMMISDK_DeleteAllTargetData	Deletes all target data.
	CMMISDK_ClearTargetInfo	Clears target information on the PC.
	CMMISDK_LoadTargetInfo	Loads target information on the PC.
	CMMISDK_SaveTargetInfo	Saves target information on the PC to the instrument.
	CMMISDK_SetTargetProperty	Sets target information properties.
	CMMISDK_GetTargetProperty	Obtains target information properties.
	CMMISDK_SetTargetData	Sets target information data.
	CMMISDK_GetTargetData	Obtains target information data.
	CMMISDK_SetToleranceForTarget	Sets the target tolerance data.
	CMMISDK_GetToleranceForTarget	Obtains the target tolerance data.
	CMMISDK_SetParametricForTarget	Sets the parametric coefficient for a target color.
	CMMISDK_GetParametricForTarget	Obtains the parametric coefficient for a target color.
	CMMISDK_SetTargetFilter	Sets the target filter conditions.
	CMMISDK_GetTargetFilter	Obtains the target filter conditions.
	CMMISDK_SetTargetProtect	Sets target protection.
	CMMISDK_GetTargetProtect	Obtains target protection.
	CMMISDK_GetSavedSampleCount	Obtains the number of saved measurement values.
	CMMISDK_DeleteSampleData	Deletes measurement value data.
	CMMISDK_DeleteAllSampleData	Deletes all measurement value data.
	CMMISDK_LoadSampleInfo	Loads measurement value information on the PC.
	CMMISDK_GetSampleProperty	Obtains measurement value information properties.
	CMMISDK_GetSampleData	Obtains measurement value information data.
Others		
	CMMISDK_SetActiveGroup	Sets the active group.
	CMMISDK_GetActiveGroup	Obtains the active group.
	CMMISDK_SetGroupName	Sets the group name.
	CMMISDK_GetGroupName	Obtains the group name.
	CMMISDK_SetMultipleGroupName	Sets group names in batch.
	CMMISDK_GetMultipleGroupName	Obtains group names in batch.
	CMMISDK_LoadDefaultInfo	Loads default information on the PC.
	CMMISDK_SaveDefaultInfo	Saves default information on the PC to the instrument.
	CMMISDK_SetTolerance	Sets default tolerance data.
	CMMISDK_GetTolerance	Obtains default tolerance data.
	CMMISDK_SetParametric	Sets the default parametric coefficient.
	CMMISDK_GetParametric	Obtains the default parametric coefficient.
	CMMISDK_SetWarningLevel	Sets the warning level.
	CMMISDK_GetWarningLevel	Obtains the warning level.
	CMMISDK_SetInstrumentMode	Sets the instrument mode.
	CMMISDK_GetInstrumentMode	Obtains the instrument mode.
	CMMISDK_SetUserType	Sets the user type.

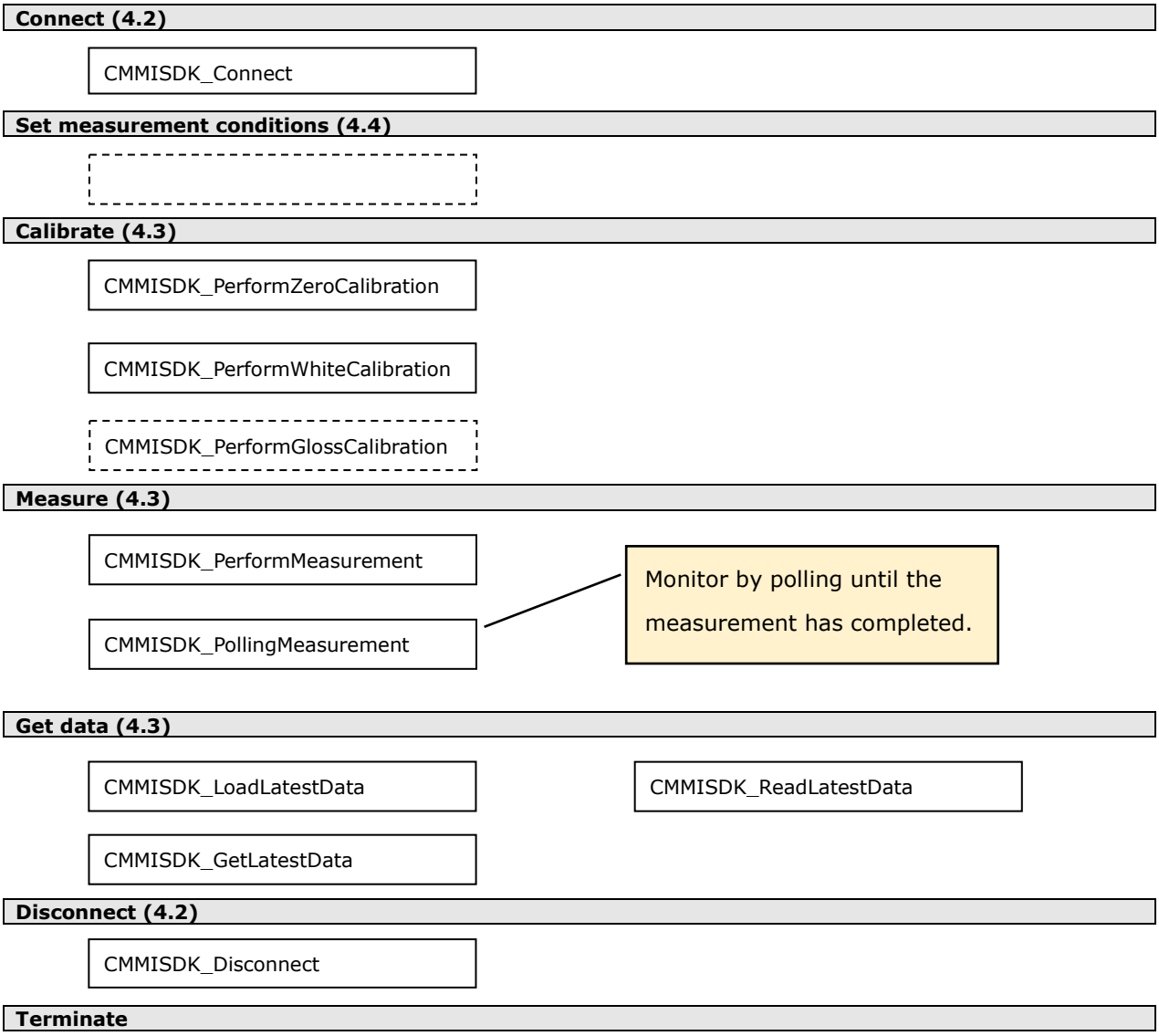
	CMMISDK_GetUserType	Obtains the user type.
	CMMISDK_SetAdminPassword	Sets the administrator password.
	CMMISDK_GetAdminPassword	Obtains the administrator password.
	CMMISDK_SetAutoPrint	Sets automatic printing.
	CMMISDK_GetAutoPrint	Obtains the automatic printing setting.
	CMMISDK_SetBrightness	Sets the brightness of the display.
	CMMISDK_GetBrightness	Obtains the brightness of the display.
	CMMISDK_SetScreenDirection	Sets the direction of the display.
	CMMISDK_GetScreenDirection	Obtains the direction of the display.
	CMMISDK_SetSound	Sets the sound.
	CMMISDK_GetSound	Obtains the sound.
	CMMISDK_SetCalibrationInterval	Sets the calibration interval.
	CMMISDK_GetCalibrationInterval	Obtains the calibration interval.
	CMMISDK_SetAnnualCalibration	Sets the periodical calibration notice.
	CMMISDK_GetAnnualCalibration	Obtains the periodical calibration notice.
	CMMISDK_SetZeroCalibrationSkip	Sets whether or not to skip zero calibration.
	CMMISDK_GetZeroCalibrationSkip	Obtains whether or not zero calibration can be skipped.
	CMMISDK_SetDateTime	Sets the date and time.
	CMMISDK_SetDateFormat	Sets the date format.
	CMMISDK_GetDateFormat	Obtains the date format.
	CMMISDK_SetLanguage	Sets the display language.
	CMMISDK_GetLanguage	Obtains the display language.
	CMMISDK_SetPowerSaving	Sets power saving mode.
	CMMISDK_GetPowerSaving	Obtains power saving mode.
	CMMISDK_ClearJobInfo	Clears job information.
	CMMISDK_SetJobIno	Sets job information.
	CMMISDK_GetJobInfo	Obtains job information.
	CMMISDK_GetJobStepType	Obtains the step type of the job.
	CMMISDK_SetJobStepForOperation	Sets an operation step of the job.
	CMMISDK_GetJobStepForOperation	Obtains an operation step of the job.
	CMMISDK_SetJobStepForResult	Sets a result step of the job.
	CMMISDK_GetJobStepForResult	Obtains a result step of the job.
	CMMISDK_SetJobImage	Sets job images.
	CMMISDK_GetJobImage	Obtains job images.
	CMMISDK_ResetSetting	Restores setting values to the initial state.
	CMMISDK_ResetSettingAndData	Restores setting values to the initial state and deletes all data.

* "●" indicates functions that are used also as conditions when calibrating and measuring via communications, not only when the instrument is standalone.

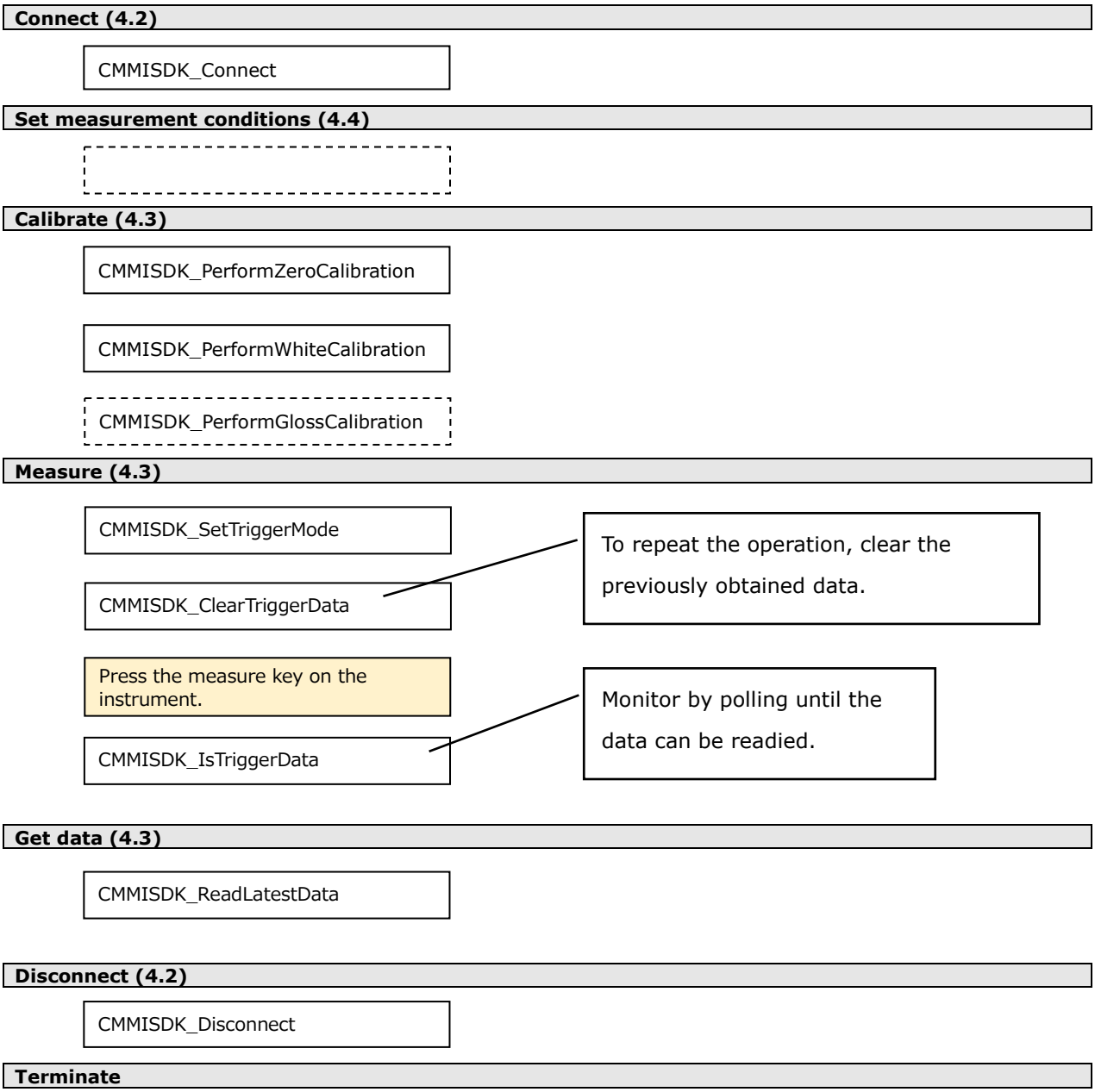
3.2 Basic processing flow

3.2.1 Measurements

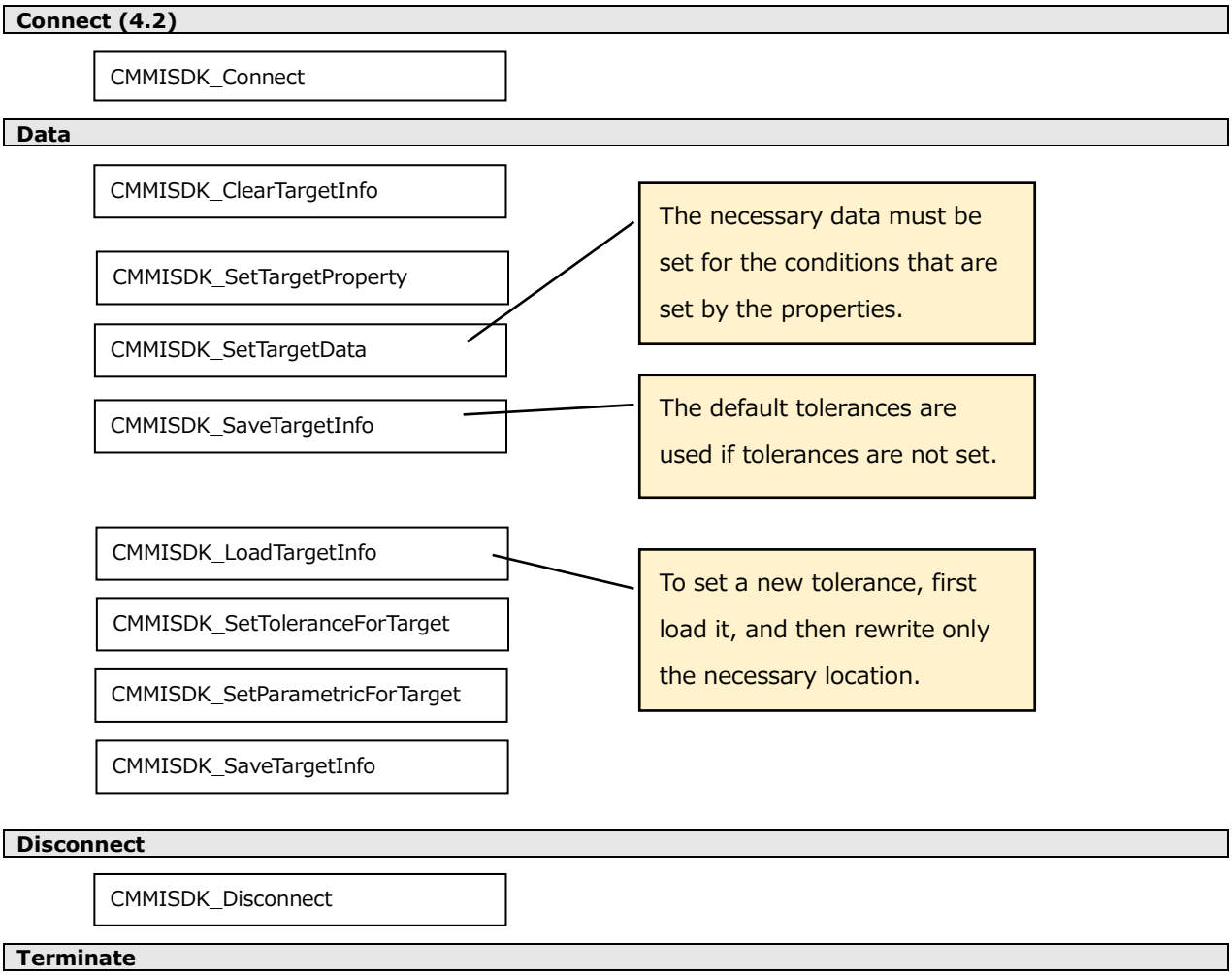
3.2.1.1 Measurements using the SDK



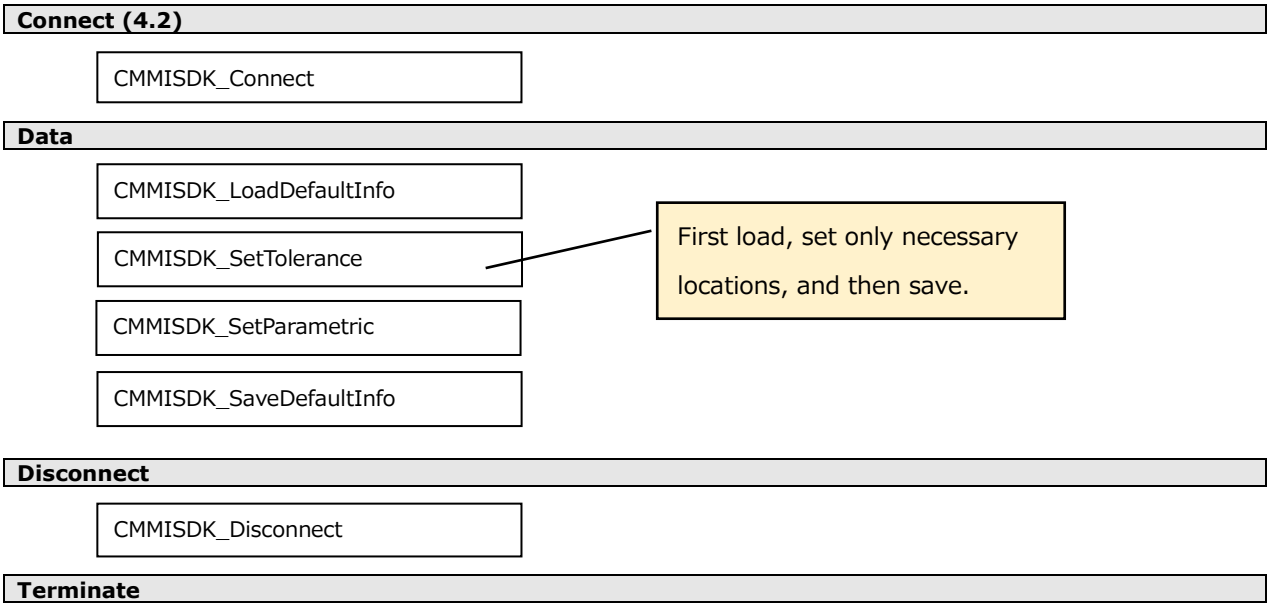
3.2.1.2 Measurements using an instrument key



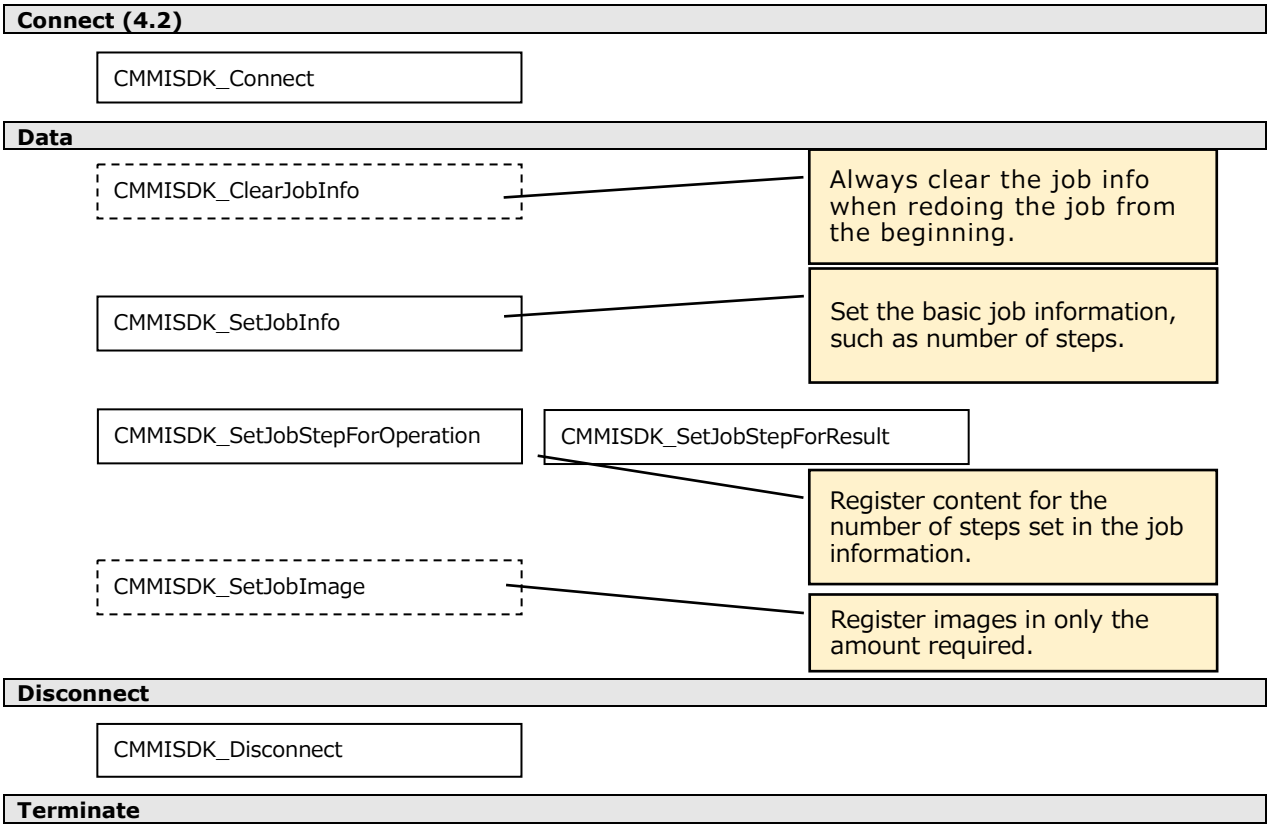
3.2.2 Target writing (when new)



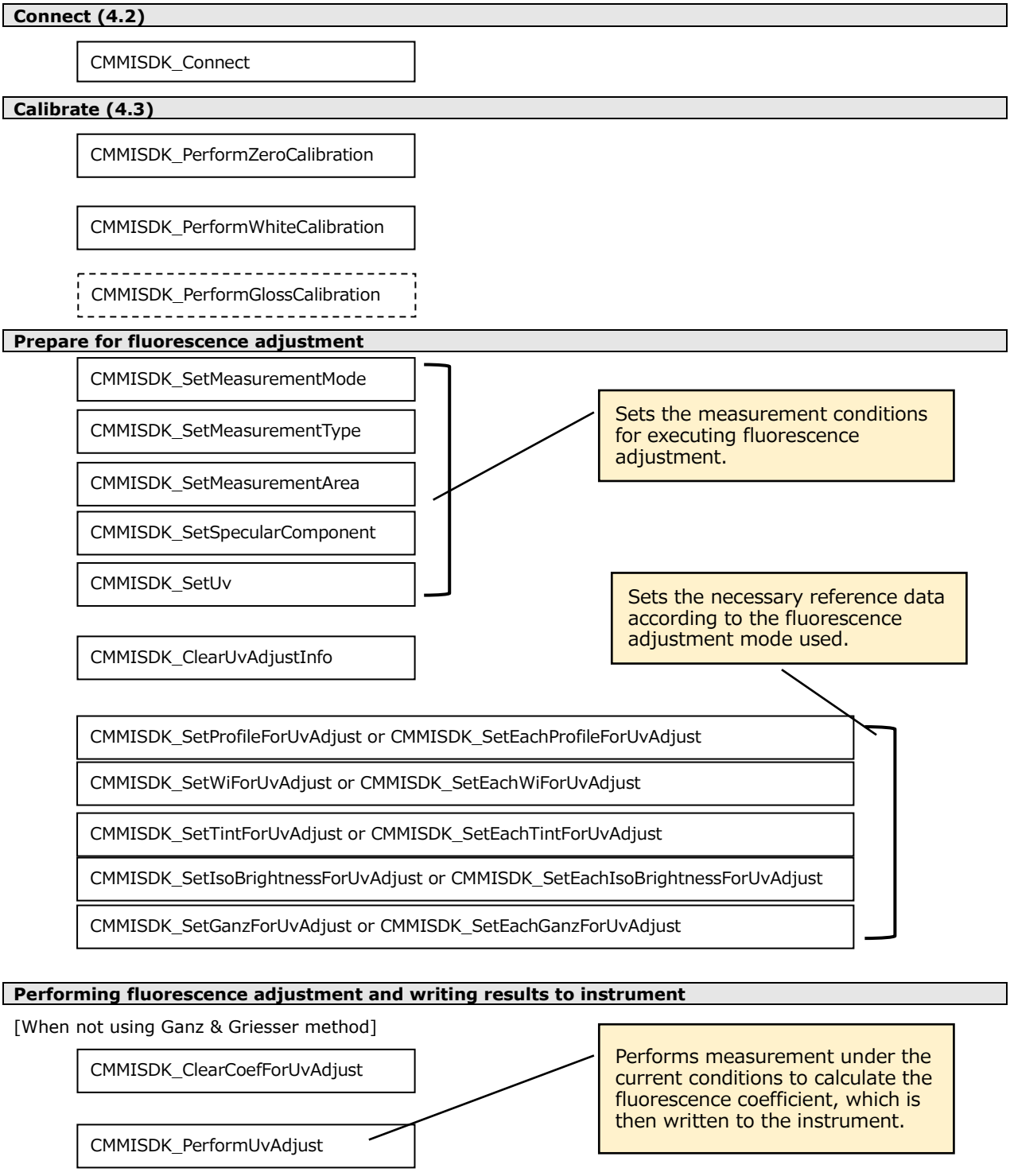
3.2.3 Default tolerance setting

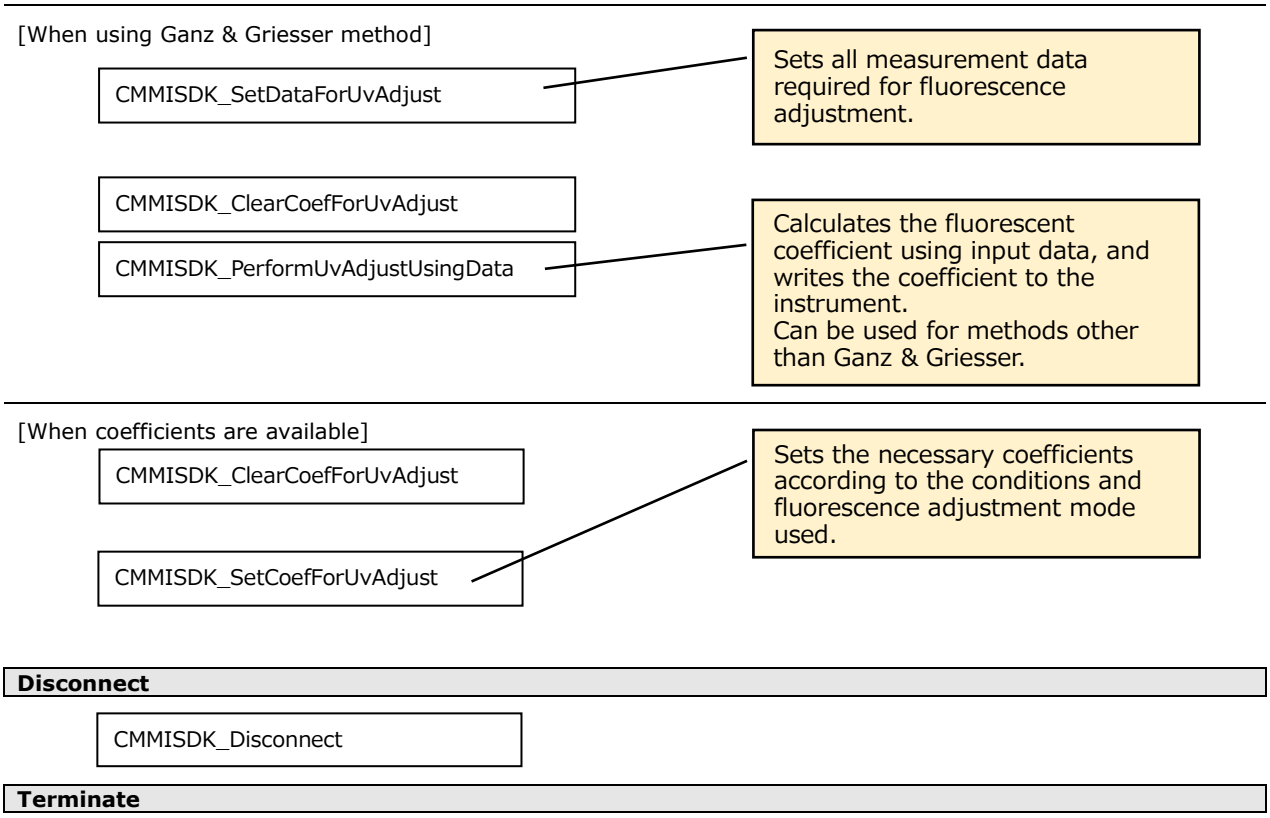


3.2.4 Job name


Connect (4.2)
CMMISDK_Connect
Data
CMMISDK_ClearJobInfo
Always clear the job info when redoing the job from the beginning.
CMMISDK_SetJobInfo
Set the basic job information, such as number of steps.
CMMISDK_SetJobStepForOperation
CMMISDK_SetJobStepForResult
Register content for the number of steps set in the job information.
CMMISDK_SetJobImage
Register images in only the amount required.
Disconnect
CMMISDK_Disconnect
Terminate

3.2.5 Fluorescence Adjustment




[When using Ganz & Griesser method]
CMMISDK_SetDataForUvAdjust
Sets all measurement data required for fluorescence adjustment.
CMMISDK_ClearCoefForUvAdjust
CMMISDK_PerformUvAdjustUsingData
Calculates the fluorescent coefficient using input data, and writes the coefficient to the instrument.
Can be used for methods other than Ganz & Griesser.
[When coefficients are available]
CMMISDK_ClearCoefForUvAdjust
CMMISDK_SetCoefForUvAdjust
Sets the necessary coefficients according to the conditions and fluorescence adjustment mode used.
Disconnect
CMMISDK_Disconnect
Terminate

3.3 How to create programs with the SDK

3.3.1 Using the SDK from a Development Environment

This section describes how to use the SDK with Visual Studio 2013 as an example.

(1) Create a C++ application project (referred to as "the project").

(2) Place "CMMISDK_***.dll" in either of the following locations.

The DLL is available as a 32-bit and 64-bit version. Use the appropriate version for the application that will be created.

- Execution folder of the application that will be created
- Folder set as an environment path

(3) To use the import library, place "CMMISDK_***.lib" in the same location as in step (2), open the project properties, select "Linker" - "Input" - "Additional Dependencies", and add "CMMISDK_***.lib".

(4) Add the header files to the project and include them in the code file.

(5) Create and build an application that uses the SDK.

3.3.2 Sample Code Overview

Three types of sample code have been prepared for this SDK.

(1) PerformMeasurement: Calibrate, measure, and get measurement data

(2) ReadSampleData: Get saved measurement data

(3) WriteTargetData: Write target data

Refer to the sample code for the specific implementation methods.

4. SDK Reference

4.1 Format of SDK functions

4.1.1 Format

The functions in the SDK are described using the following format.

Format:
Describes the format of the function.

Arguments:
Describes the arguments of the function.

Return Value:
Describes the return value that is returned when the function is used.

There are three types of return values.

Type	Value	
Success	0	Returned when the processing was successful.
Warning	1	Returned when the processing was successful, but with restrictions. Use CMMISDK_GetWarning to get detailed information.
Error	Value larger than 1	Returned when the processing failed. Refer to "6. List of errors" to make processing complete successfully.

Description:
Describes necessary information and precautions when using the function.

4.2 Connect/disconnect

CMMISDK_Connect: Connects to the instrument.

Format:

error_km CMMISDK_Connect(const CMMISDK_Port* inPortInfo, int32_km* outInstrumentNo)

Arguments:

Name	I/O	Explanation
inPortInfo	I	Communication port to which the instrument is connected * When connecting to COM1, for example, specify "COM1".
outInstrumentNo	O	Instrument number (0 to 7) * -1 is returned when failed.

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This method connects the computer to the instrument that is connected to the specified virtual COM port.
If the connection to the instrument succeeds, the instrument number is returned.
The instrument number is a number between 0 and 7, and this number is passed as a parameter to subsequent functions.

The SDK can simultaneously communicate with up to 8 instruments.
This function must be called 8 times to connect to 8 instruments.

CMMISDK_Disconnect: Ends the communication with an instrument.

Format:

error_km CMMISDK_Disconnect(int32_km inInstrumentNo)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function ends the communication with the instrument connected to the specified virtual COM port. When communications are ended, the measurement data of the specified instrument is cleared.

CMMISDK_GetInstrumentInfo: Obtains instrument information.

Format:

error_km CMMISDK_GetInstrumentInfo(int32_km inInstrumentNo, CMMISDK_InstrumentInfo* outInfo)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outInfo	O	Instrument information

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains instrument information.

CMMISDK_GetSDKVersion: Obtains the SDK version.

Format:

error_km CMMISDK_GetSDKVersion(CMMISDK_Version* version)

Arguments:

Name	I/O	Explanation
version	O	Version information

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the SDK version.

CMMISDK_GetWarning: Obtains the warning status.

Format:

error_km CMMISDK_GetWarning(int32_km inInstrumentNo, CMMISDK_Warning* warning)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
warning	O	Warning status

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the warning status.
If the return value of functions is KmWarning, use this function to obtain the warning.
Refer to the List of warnings for details on the warnings.

4.3 Calibration and measurement

CMMISDK_GetCalibrationStatus: Obtains the calibration status.

Format:

error_km CMMISDK_GetCalibrationStatus(int32_km inInstrumentNo, CMMISDK_CalStatus* outCalStatus)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outCalStatus	O	Calibration status

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function gets the calibration status.
Execute measurements after checking the calibration status and performing calibration if necessary.

The calibration status is managed for each condition type (e.g. measurement area, specular component).
This function obtains the calibration status based on the conditions set on the instrument.

CMMISDK_PerformZeroCalibration: Executes zero calibration.

Format:

error_km CMMISDK_PerformZeroCalibration(int32_km inInstrumentNo)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.
KmErCalibration	Calibration was not executed in the correct procedure.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This command executes zero calibration.
No response is returned until zero calibration has completed.
If the zero calibration fails, the normal state immediately before the zero calibration is attempted is maintained.

Zero calibration does not need to be performed each time, but it should be performed when the measurement environment changes greatly and when the instrument has not be used for a long period of time.

CMMISDK_PerformWhiteCalibration: Executes white calibration.

Format:

error_km CMMISDK_PerformWhiteCalibration(int32_km inInstrumentNo)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.
KmErCalibration	Calibration was not executed in the correct procedure.
KmErCalibrationRequired	Necessary calibration was not executed beforehand.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function executes white calibration. Check the IDs of the white calibration plates, and use the matching plate.
No response is returned until white calibration has completed.
If the white calibration fails, the normal state immediately before the white calibration is attempted is maintained.

Because the calibration status is managed for each condition type (e.g. measurement area, specular component), re-calibration may be required if any condition is changed.
Use CMMISDK_GetCalibrationStatus to determine whether calibration is necessary.

If no white calibration data is set, use CMMISDK_SetWhiteCalibrationData to set the data.

CMMISDK_PerformGlossCalibration: Executes gloss calibration.

Format:

error_km CMMISDK_PerformGlossCalibration(int32_km inInstrumentNo)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.
KmErCalibration	Calibration was not executed in the correct procedure.
KmErCalibrationRequired	Necessary calibration was not executed beforehand.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	×	✓	×	×	×

Description:

This function executes gloss calibration. Check the IDs of the gloss calibration plates, and use the matching plate.
No response is returned until gloss calibration has completed.
If the gloss calibration fails, the normal state immediately before the gloss calibration is attempted is maintained.

Because the calibration status is managed for each condition type (e.g. measurement area, specular component), re-calibration may be required if any condition is changed.
Use CMMISDK_GetCalibrationStatus to determine whether calibration is necessary.

If no gloss calibration plate data is set, use CMMISDK_SetGlossCalibrationData to set the data.

CMMISDK_PerformUserCalibration: Executes user calibration.

Format:

error_km CMMISDK_PerformUserCalibration(int32_km inInstrumentNo)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.
KmErCalibration	Calibration was not executed in the correct procedure.
KmErCalibrationRequired	Necessary calibration was not executed beforehand.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	×	✓	✓	✓	×

Description:

This function executes user calibration. Check the IDs of the user calibration plates, and use the matching plate.
No response is returned until user calibration has completed.
If the user calibration fails, the normal state immediately before the user calibration is attempted is maintained.

Because the calibration status is managed for each condition type (e.g. measurement area, specular component), re-calibration may be required if any condition is changed.
Use CMMISDK_GetCalibrationStatus to determine whether calibration is necessary.

If no user calibration data is set, use CMMISDK_SetUserCalibrationData to set the data.

CMMISDK_PerformMeasurement: Executes the measurement.

Format:

error_km CMMISDK_PerformMeasurement(int32_km inInstrumentNo)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function executes the measurement.
This function returns a response when the measurement starts.
Use CMMISDK_PollingMeasurement to determine the completion of the measurement, and after the measurement is complete, use CMMISDK_ReadLatestData to get data.

- Errors related to the measurement can be obtained only with CMMISDK_PollingMeasurement.

When this function has completed successfully and the next measurement is started, the retained measurement data is cleared. For this reason, the previous data cannot be retrieved in case measurement fails.

CMMISDK_PollingMeasurement: Obtains the measurement status.

Format:

error_km CMMISDK_PollingMeasurement(int32_km inInstrumentNo, CMMISDK_MeasStatus* outStatus)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outStatus	O	Measurement status

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErOutOfRangeValue	The value is outside the range that can be measured by the instrument.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.
KmErCalibrationRequired	Necessary calibration was not executed beforehand.
KmErTileDetection	The instrument could not measure correctly because it is tilted.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function gets the measurement status.
After verifying that the status has changed from 'measuring' to 'idling,' use CMMISDK_ReadLatestData to retrieve data.

CMMISDK_CancelMeasurement: Stops the measurement.

Format:

error_km CMMISDK_CancelMeasurement(int32_km inInstrumentNo)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function stops the measurement.
It can be used when the number of times of automatic averaging is set to multiple times using CMMISDK_SetAutoAverageTimes.

If this function is executed when a measurement is not being executed, it will return KmSuccess or KmWarning.

CMMISDK_ReadLatestData: Obtains the latest measurement data.

Format 1:

error_km CMMISDK_ReadLatestDataSpec(int32_km inInstrumentNo, CMMISDK_DataType inDataType, CMMISDK_Data* outData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inDataType	I	Data type
outData	O	Reflectance data * Data in the amount of DataSize that was obtained with CMMISDK_GetInstrumentInfo is stored from the beginning of the array.

Format 2:

error_km CMMISDK_ReadLatestDataColor(int32_km inInstrumentNo, CMMISDK_DataType inDataType, const CMMISDK_ColorCond* inColorCond, CMMISDK_Data* outData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inDataType	I	Data type
inColorCond	I	Color value calculation conditions * The color value is calculated with the specified conditions
outData	O	Color value data * Data in the amount of the number of color values is stored from the beginning of the array.

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErNoData	No data
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function retrieves the latest measurement data.
For reflectance, the result is output according to the specified data type. For color value, the result is output according to the specified data type and color space.

Use this function to obtain data even when a key on the instrument was used for the measurement.

The size of data to allocate will depend on the instrument.
Use CMMISDK_GetInstrumentInfo to check the size of data.

Gloss data can be obtained with both format 1 and format 2.
When gloss data is obtained, it is stored from the beginning of the array.

* For the 26dG/26d/25d/23d
Opacity measurement is standalone only. If the measurement mode is opacity, the instrument operates in the following states.

26dG	MEASMODE_COLORANDGLOSS
26d	MEASMODE_COLORONLY
25d	MEASMODE_COLORONLY

23d	MEASMODE_COLORONLY

CMMISDK_LoadLatestData: Loads the latest data onto the PC.

Format:

error_km CMMISDK_LoadLatestData(int32_km inInstrumentNo)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErNoData	No data
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function loads the latest measurement data onto the PC.

Measurement data refers to all data that can be obtained according to the measurement conditions.

Use of this function instead of CMMISDK_ReadLatestData is recommended when collectively acquiring data because all measurement data can be acquired at once.

Use this function to load data even when a key on the instrument was used for the measurement.

After the data is loaded, use CMMISDK_GetLatestData to obtain data individually.

CMMISDK_GetLatestData: Obtains the latest data on the PC.

Format 1:

error_km CMMISDK_GetLatestDataSpec(int32_km inInstrumentNo, CMMISDK_DataType inDataType, CMMISDK_Data* outData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inDataType	I	Data type
outData	O	Reflectance data * Data in the amount of DataSize that was obtained with CMMISDK_GetInstrumentInfo is stored from the beginning of the array.

Format 2:

error_km CMMISDK_GetLatestDataColor(int32_km inInstrumentNo, CMMISDK_DataType inDataType, const CMMISDK_ColorCond* inColorCond, CMMISDK_Data* outData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inDataType	I	Data type
inColorCond	I	Color value calculation conditions * The color value is calculated with the specified conditions
outData	O	Color value data * Data in the amount of the number of color values is stored from the beginning of the array.

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErNoData	No data
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the latest measurement data on the PC.

The data for the specified number on the PC can be obtained by using CMMISDK_LoadLatestData.

For reflectance, the result is output according to the specified data type. For color value, the result is output according to the specified data type and color space.

Use this function to obtain data even when a key on the instrument was used for the measurement.

The size of data to allocate will depend on the instrument.
Use CMMISDK_GetInstrumentInfo to check the size of data.

Gloss data can be obtained with both format 1 and format 2.
When gloss data is obtained, it is stored from the beginning of the array.

* For the 26dG/26d/25d/23d
Opacity measurement is standalone only. If the measurement mode is opacity, the instrument operates in the following states.

26dG	MEASMODE_COLORANDGLOSS

26d	MEASMODE_COLORONLY
25d	MEASMODE_COLORONLY
23d	MEASMODE_COLORONLY

CMMISDK_SetWhiteCalibrationData: Sets the white calibration data.

Format:

error_km CMMISDK_SetWhiteCalibrationData(int32_km inInstrumentNo, CMMISDK_CalDataType inDataType, int32_km inCalId, const CMMISDK_Data* inCalData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inDataType	I	Data type
inCalId	I	White calibration plate ID * The calibration plate ID will be overwritten by the ID that was last set.
inCalData	I	White calibration data * Store the data from the beginning of the array in the amount of DataSize that was obtained with CMMISDK_GetInstrumentInfo.

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function sets the white calibration data.
White calibration is executed using the data set via this function.

CMMISDK_GetWhiteCalibrationData: Obtains the white calibration data.

Format:

error_km CMMISDK_GetWhiteCalibrationData(int32_km inInstrumentNo, CMMISDK_CalDataType inDataType, int32_km* outCalId, CMMISDK_Data* outCalData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inDataType	I	Data type
outCalId	O	White calibration plate ID
outCalData	O	White calibration data * Data in the amount of DataSize that was obtained with CMMISDK_GetInstrumentInfo is stored from the beginning of the array.

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the white calibration data.

CMMISDK_SetGlossCalibrationData: Sets the gloss calibration data.

Format:

error_km CMMISDK_SetGlossCalibrationData(int32_km inInstrumentNo, CMMISDK_MeasArea inArea, int32_km inCalId, float64_km inCalData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inArea	I	Measurement area
inCalId	I	Gloss calibration plate ID * The calibration plate ID will be overwritten by the ID that was last set.
inCalData	I	Gloss calibration data

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	×	✓	×	×	×

Description:

This function sets the gloss calibration data.
Gloss calibration is executed using the data set via this function.

CMMISDK_GetGlossCalibrationData: Obtains the gloss calibration data.

Format:

error_km CMMISDK_GetGlossCalibrationData(int32_km inInstrumentNo, CMMISDK_MeasArea inArea, int32_km* outCalId, float64_km* outCalData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inArea	I	Measurement area
outCalId	O	Gloss calibration plate ID
outCalData	O	Gloss calibration data

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	×	✓	×	×	×

Description:

This function obtains the gloss calibration data.

CMMISDK_SetUserCalibrationData: Sets the user calibration data.

Format:

error_km CMMISDK_SetUserCalibrationData(int32_km inInstrumentNo, CMMISDK_CalDataType inDataType, const CMMISDK_UserCalId* inCalId, const CMMISDK_Data* inCalData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inDataType	I	Data type
inCalId	I	User ID (8 characters) * The user ID will be overwritten by the ID that was last set.
inCalData	I	User calibration data (range: 50.0 to 150.0) * Store the data from the beginning of the array in the amount of DataSize that was obtained with CMMISDK_GetInstrumentInfo.

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	×	✓	✓	✓	×

Description:

This function sets the user calibration data.
User calibration is executed using the data set via this function.
If the user ID is blank, it will be treated as if there is no data.

CMMISDK_GetUserCalibrationData: Obtains the user calibration data.

Format:

error_km CMMISDK_GetUserCalibrationData(int32_km inInstrumentNo, CMMISDK_CalDataType inDataType, CMMISDK_UserCalId* outCalId, CMMISDK_Data* outCalData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inDataType	I	Data type
outCalId	O	User ID
outCalData	O	User calibration data * Data in the amount of DataSize that was obtained with CMMISDK_GetInstrumentInfo is stored from the beginning of the array.

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	×	✓	✓	✓	×

Description:

This function obtains the user calibration data.

CMMISDK_SetUserCalibrationEnable: Enables and disables user calibration.

Format:

error_km CMMISDK_SetUserCalibrationEnable(int32_km inInstrumentNo, CMMISDK_OnOff inCalEnable)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inCalEnable	I	User calibration on/off

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	×	✓	✓	✓	×

Description:

This function enables and disables user calibration.

If user calibration is turned on, perform user calibration instead of white calibration.

CMMISDK_GetUserCalibrationEnable: Obtains the user calibration enabled or disabled state.

Format:

error_km CMMISDK_GetUserCalibrationEnable(int32_km inInstrumentNo, CMMISDK_OnOff* outCalEnable)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outCalEnable	O	User calibration on/off

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	×	✓	✓	✓	×

Description:

This function obtains the user calibration enabled or disabled state.

CMMISDK_SetTriggerMode: Enables and disables trigger mode.

Format:

error_km CMMISDK_SetTriggerMode(int32_km inInstrumentNo, CMMISDK_OnOff inTrigger)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inTrigger	I	Trigger mode

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function enables and disables trigger mode.
If the trigger mode is turned on, the instrument key can be used as a measurement trigger.
To obtain the data, confirm the availability of the data with CMMISDK_IsTriggerData, and then use CMMISDK_ReadLatestData.

CMMISDK_GetTriggerMode: Obtains the trigger mode enabled or disabled state.

Format:

error_km CMMISDK_GetTriggerMode(int32_km inInstrumentNo, CMMISDK_OnOff* outTrigger)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outTrigger	O	Trigger mode

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the trigger mode enabled or disabled state.

CMMISDK_ClearTriggerData: Clears the trigger mode data.

Format:

error_km CMMISDK_ClearTriggerData(int32_km inInstrumentNo)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function clears the data measured with the instrument key.
Use this function to clear the data after you have taken a measurement using the instrument key and finished obtaining the data.

CMMISDK_IsTriggerData: Obtains the availability of trigger mode data.

Format:

error_km CMMISDK_IsTriggerData(int32_km inInstrumentNo, CMMISDK_OnOff* outData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outData	O	Availability of trigger mode data

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the availability of trigger mode data.
If the function returns a value of on, there is data. The data can be obtained by using CMMISDK_ReadLatestData.

CMMISDK_GetZeroCalibrationDate: Obtains the zero calibration date and time.

Format:

error_km CMMISDK_GetZeroCalibrationDate(int32_km inInstrumentNo, CMMISDK_DateType inType, CMMISDK_DateTime* outDate)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inDateType	I	Date/time type
outDate	O	Zero calibration date/time

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.
KmErCalibrationRequired	Necessary calibration was not executed beforehand.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the zero calibration date and time.

If calibration was not executed, this function returns KmErCalibrationRequired.

CMMISDK_GetWhiteCalibrationDate: Obtains the white calibration date and time.

Format:

error_km CMMISDK_GetWhiteCalibrationDate(int32_km inInstrumentNo, CMMISDK_DateTime* outDate)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outDate	O	White calibration date/time

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.
KmErCalibrationRequired	Necessary calibration was not executed beforehand.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the white calibration date and time.

If calibration was not executed, this function returns KmErCalibrationRequired.

CMMISDK_GetGlossCalibrationDate: Obtains the gloss calibration date and time.

Format:

error_km CMMISDK_GetGlossCalibrationDate(int32_km inInstrumentNo, CMMISDK_DateTime* outDate)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outDate	O	Gloss calibration date/time

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.
KmErCalibrationRequired	Necessary calibration was not executed beforehand.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	×	✓	×	×	×

Description:

This function obtains the gloss calibration date and time.
If calibration was not executed, this function returns KmErCalibrationRequired.

CMMISDK_GetUserCalibrationDate: Obtains the user calibration date and time.

Format:

error_km CMMISDK_GetUserCalibrationDate(int32_km inInstrumentNo, CMMISDK_DateTime* outDate)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outDate	O	User calibration date/time

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.
KmErCalibrationRequired	Necessary calibration was not executed beforehand.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	×	✓	✓	✓	×

Description:

This function obtains the user calibration date and time.

If calibration was not executed, this function returns KmErCalibrationRequired.

CMMISDK_ClearUvAdjustInfo: Clears various data for fluorescence adjustment.

Format:

error_km CMMISDK_ClearUvAdjustInfo(int32_km inInstrumentNo)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	×	×

* Dependent on instrument and version.

Description:

This function clears various data for fluorescence adjustment.
Such data includes profile, WI, Tint, ISO brightness, Ganz & Griesser, and fluorescence coefficient/correction value information.

To clear fluorescence coefficient/correction value information in the instrument, use CMMISDK_ClearCoefForUvAdjust.

CMMISDK_SetProfileForUvAdjust: Sets the fluorescence adjustment profile data.

Format 1:

error_km CMMISDK_SetProfileForUvAdjust(int32_km inInstrumentNo, const CMMISDK_Data* inData)

Format 2:

error_km CMMISDK_SetEachProfileForUvAdjust(int32_km inInstrumentNo, CMMISDK_UvAdjustDataType inType, const CMMISDK_Data* inData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inType	I	Type of data to be set
inData	I	Fluorescence adjustment profile data Value range: 0.01 to 200.00

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	×	×

* Dependent on instrument and version.

Description:

This function sets the fluorescence adjustment profile data.

For Format 1, this function sets the same value for all data types that can be set.

CMMISDK_GetProfileForUvAdjust: Obtains the fluorescence adjustment profile data.

Format 1:

error_km CMMISDK_GetProfileForUvAdjust(int32_km inInstrumentNo, CMMISDK_Data* outData)

Format 2:

error_km CMMISDK_GetEachProfileForUvAdjust(int32_km inInstrumentNo, CMMISDK_UvAdjustDataType inType, CMMISDK_Data* outData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inType	I	Type of data to be get
outData	O	Fluorescence adjustment profile data

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErNoData	No data
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	×	×

* Dependent on instrument and version.

Description:

This function obtains the fluorescence adjustment profile data.

For Format 1, this function returns the first data for the data type.

CMMISDK_SetWiForUvAdjust: Sets the WI for fluorescence adjustment.

Format 1:

error_km CMMISDK_SetWiForUvAdjust(int32_km inInstrumentNo, const CMMISDK_UvAdjustIndex* inData)

Format 2:

error_km CMMISDK_SetEachWiForUvAdjust(int32_km inInstrumentNo, CMMISDK_UvAdjustDataType inType, const CMMISDK_UvAdjustIndex* inData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inType	I	Type of data to be set
inData	I	WI for fluorescence adjustment Value range: 40.00 to 250.00 Tolerance range: 0.20 to 3.00

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	×	×

* Dependent on instrument and version.

Description:

This function sets the WI for fluorescence adjustment.

For Format 1, this function sets the same value for all data types that can be set.

CMMISDK_GetWiForUvAdjust: Obtains the WI for fluorescence adjustment.

Format 1:

error_km CMMISDK_GetWiForUvAdjust(int32_km inInstrumentNo, CMMISDK_UvAdjustIndex* outData)

Format 2:

error_km CMMISDK_GetEachWiForUvAdjust(int32_km inInstrumentNo, CMMISDK_UvAdjustDataType inType, CMMISDK_UvAdjustIndex* outData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inType	I	Type of data to be get
outData	O	WI for fluorescence adjustment

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErNoData	No data
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	×	×

* Dependent on instrument and version.

Description:

This function obtains the WI for fluorescence adjustment.

For Format 1, this function returns the first data for the data type.

CMMISDK_SetTintForUvAdjust: Sets the Tint for fluorescence adjustment.

Format 1:

error_km CMMISDK_SetTintForUvAdjust(int32_km inInstrumentNo, const CMMISDK_UvAdjustIndex* inData)

Format 2:

error_km CMMISDK_SetEachTintForUvAdjust(int32_km inInstrumentNo, CMMISDK_UvAdjustDataType inType, const CMMISDK_UvAdjustIndex* inData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inType	I	Type of data to be set
inData	I	Tint for fluorescence adjustment Value range: -6.00 to 6.00 Tolerance range: 0.05 to 0.30

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	×	×

* Dependent on instrument and version.

Description:

This function sets the Tint for fluorescence adjustment.

For Format 1, this function sets the same value for all data types that can be set.

CMMISDK_GetTintForUvAdjust: Obtains the Tint for fluorescence adjustment.

Format 1:

error_km CMMISDK_GetTintForUvAdjust(int32_km inInstrumentNo, CMMISDK_UvAdjustIndex* outData)

Format 2:

error_km CMMISDK_GetEachTintForUvAdjust(int32_km inInstrumentNo, CMMISDK_UvAdjustDataType inType, CMMISDK_UvAdjustIndex* outData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inType	I	Type of data to be get
outData	O	Tint for fluorescence adjustment

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErNoData	No data
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	×	×

* Dependent on instrument and version.

Description:

This function obtains the Tint for fluorescence adjustment.

For Format 1, this function returns the first data for the data type.

CMMISDK_SetIsoBrightnessForUvAdjust: Sets the ISO brightness for fluorescence adjustment.

Format 1:

error_km CMMISDK_SetIsoBrightnessForUvAdjust(int32_km inInstrumentNo, const CMMISDK_UvAdjustIndex* inData)

Format 2:

error_km CMMISDK_SetEachIsoBrightnessForUvAdjust(int32_km inInstrumentNo, CMMISDK_UvAdjustDataType inType, const CMMISDK_UvAdjustIndex* inData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inType	I	Type of data to be set
inData	I	ISO brightness for fluorescence adjustment Value range: 40.00 to 250.00 Tolerance range: 0.50 to 3.00

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	×	×

* Dependent on instrument and version.

Description:

This function sets the ISO brightness for fluorescence adjustment.

For Format 1, this function sets the same value for all data types that can be set.

CMMISDK_GetIsoBrightnessForUvAdjust: Obtains the ISO brightness for fluorescence adjustment.

Format 1:

error_km CMMISDK_GetIsoBrightnessForUvAdjust(int32_km inInstrumentNo, CMMISDK_UvAdjustIndex* outData)

Format 2:

error_km CMMISDK_GetEachIsoBrightnessForUvAdjust(int32_km inInstrumentNo, CMMISDK_UvAdjustDataType inType, CMMISDK_UvAdjustIndex* outData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inType	I	Type of data to be get
outData	O	ISO brightness for fluorescence adjustment

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErNoData	No data
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	×	×

* Dependent on instrument and version.

Description:

This function obtains the ISO brightness for fluorescence adjustment.

For Format 1, this function returns the first data for the data type.

CMMISDK_SetGanzForUvAdjust: Sets the Ganz & Griesser for fluorescence adjustment.

Format 1:

error_km CMMISDK_SetGanzForUvAdjust(int32_km inInstrumentNo, const CMMISDK_UvAdjustGG* inData)

Format 2:

error_km CMMISDK_SetGanzForUvAdjust(int32_km inInstrumentNo, CMMISDK_UvAdjustDataType inType, const CMMISDK_UvAdjustGG* inData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inType	I	Type of data to be set
inData	I	Ganz & Griesser for fluorescence adjustment WI range: 40.00 to 250.00 Tint range: -6.00 to 6.00

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	×	×

* Dependent on instrument and version.

Description:

This function sets the Ganz & Griesser targets for fluorescence adjustment.

For Format 1, this function sets the same value for all data types that can be set.

CMMISDK_GetGanzForUvAdjust: Obtains the Ganz & Griesser for fluorescence adjustment.

Format 1:

error_km CMMISDK_GetGanzForUvAdjust(int32_km inInstrumentNo, CMMISDK_UvAdjustGG* outData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inType	I	Type of data to be get
outData	O	Ganz & Griesser for fluorescence adjustment

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErNoData	No data
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	×	×

* Dependent on instrument and version.

Description:

This function obtains the Ganz & Griesser targets for fluorescence measurement.

For Format 1, this function returns the first data for the data type.

CMMISDK_SetDataForUvAdjust: Sets the data for fluorescence adjustment.

Format:

error_km CMMISDK_SetDataForUvAdjust(int32_km inInstrumentNo, CMMISDK_UvAdjustDataType inType, int32_km inNum, const CMMISDK_Data* inFull, const CMMISDK_Data* inCut)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inType	I	Data types for fluorescence adjustment
inNum	I	No. * When not using Ganz & Griesser: "0" only * When using Ganz & Griesser: "0" to "3" or "0" to "4"
inFull	I	UV full measurement data Value range: 0.00 to 300.00
inCut	I	UV cut measurement data Value range: 0.00 to 300.00

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	×	×

* Dependent on instrument and version.

Description:

This function sets the data for fluorescence adjustment.
When using the Ganz & Griesser method, the data must be set by using this API.

CMMISDK_GetDataForUvAdjust: Obtains the data for fluorescence adjustment.

Format:

error_km CMMISDK_GetDataForUvAdjust(int32_km inInstrumentNo, CMMISDK_UvAdjustDataType inType, int32_km inNum, CMMISDK_Data* outFull, CMMISDK_Data* outCut)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inType	I	Data types for fluorescence adjustment
inNum	I	No. * When not using Ganz & Griesser: "0" only * When using Ganz & Griesser: "0" to "3" or "0" to "4"
outFull	O	UV full measurement data
outCut	O	UV cut measurement data

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErNoData	No data
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	×	×

* Dependent on instrument and version.

Description:

This function obtains the data for fluorescence adjustment.

CMMISDK_PerformUvAdjust: Executes fluorescence adjustment and sets the coefficient to the instrument.

Format:

error_km CMMISDK_PerformUvAdjust(int32_km inInstrumentNo, CMMISDK_CondUvAdjust inCond)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inCond	I	Conditions for fluorescence adjustment

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.
KmErNoData	The data does not exist (the required data is not available).
KmErUvAdjust	The measurement sample does not contain fluorescence.
KmErCalculateCoef	The fluorescence coefficient cannot be calculated.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	×	×

* Dependent on instrument and version.

Description:

This function executes fluorescence adjustment.
This API executes measurement, calculates the fluorescence coefficient under the specified conditions, and then writes the data to the instrument.

If UVADJ_GG is specified in the conditions, KmErInvalidParameter will be returned.
If fluorescence adjustment is performed using the Ganz & Griesser method, CMMISDK_PerformUvAdjustUsingData should be used.

CMMISDK_PerformUvAdjustUsingData: Executes fluorescence adjustment and sets the coefficient to the instrument.

Format:

error_km CMMISDK_PerformUvAdjustUsingData(int32_km inInstrumentNo, CMMISDK_CondUvAdjust inCond)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inCond	I	Conditions for fluorescence adjustment

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.
KmErNoData	The data does not exist (the required data is not available).
KmErUvAdjust	The measurement sample does not contain fluorescence.
KmErCalculateCoef	The fluorescence coefficient cannot be calculated.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	×	×

* Dependent on instrument and version.

Description:

This function executes fluorescence adjustment.

This API uses the data from CMMISDK_SetDataForUvAdjust, calculates the fluorescence coefficient under the specified conditions, and then writes the data to the instrument.

CMMISDK_ClearCoefForUvAdjust: Clears the fluorescence adjustment coefficient in the instrument.

Format:

error_km CMMISDK_ClearCoefForUvAdjust(int32_km inInstrumentNo)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	×	×

* Dependent on instrument and version.

Description:

This function clears the fluorescence adjustment coefficient saved in the instrument.

CMMISDK_SetCoefForUvAdjust: Sets the fluorescence coefficient.

Format:

error_km CMMISDK_SetCoefForUvAdjust(int32_km inInstrumentNo, CMMISDK_MeasArea inArea, CMMISDK_UvAdjustDataType inType, CMMISDK_CondUvAdjust inCond, const CMMISDK_UvAdjustCoef* inCoef)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inArea	I	Measurement area
inType	I	Type of data to be set
inCond	I	Conditions for fluorescence adjustment
inCoef	I	Fluorescence adjustment coefficient

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	×	×

* Dependent on instrument and version.

Description:

This function sets the fluorescence coefficient in the instrument.

CMMISDK_GetCoefForUvAdjust: Obtains the fluorescence coefficient.

Format:

error_km CMMISDK_GetCoefForUvAdjust(int32_km inInstrumentNo, CMMISDK_MeasArea inArea, CMMISDK_UvAdjustDataType inType, CMMISDK_CondUvAdjust* outCond, CMMISDK_UvAdjustCoef* outCoef)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inArea	I	Measurement area
inType	I	Type of data to be get
outCond	O	Conditions for fluorescence adjustment
outCoef	O	Fluorescence adjustment coefficient

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErNoData	No data
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	×	×

* Dependent on instrument and version.

Description:

This method obtains the fluorescence coefficient in the instrument.

4.4 Measurement conditions (instruments settings)

CMMISDK_SetMeasurementArea: Sets the measurement area.

Format:

error_km CMMISDK_SetMeasurementArea(int32_km inInstrumentNo, CMMISDK_MeasArea inArea)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inArea	I	Measurement area

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	×	×	×	×

Description:

This function sets the measurement area.

Calibration may be required again when conditions are changed.
Use CMMISDK_GetCalibrationStatus to determine whether calibration is necessary.

CMMISDK_GetMeasurementArea: Obtains the measurement area.

Format:

error_km CMMISDK_GetMeasurementArea(int32_km inInstrumentNo, CMMISDK_MeasArea* outArea)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outArea	O	Measurement area

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	×	✓	✓	×	×

Description:

This function obtains the measurement area.

CMMISDK_SetMeasurementType: Sets the measurement type.

Format:

error_km CMMISDK_SetMeasurementType(int32_km inInstrumentNo, CMMISDK_MeasType inType)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inType	I	Measurement type

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	×	×	×	×

Description:

This function sets the measurement type.

Calibration may be required again when conditions are changed.
Use CMMISDK_GetCalibrationStatus to determine whether calibration is necessary.

CMMISDK_GetMeasurementType: Obtains the measurement type.

Format:

error_km CMMISDK_GetMeasurementType(int32_km inInstrumentNo, CMMISDK_MeasType* outType)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outType	O	Measurement type

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	×	×	×	×

Description:

This function obtains the measurement type.

CMMISDK_SetMeasurementAngle: Sets the measurement angle.

Format:

error_km CMMISDK_SetMeasurementAngle(int32_km inInstrumentNo, CMMISDK_MeasAngle inAngle)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inAngle	I	Measurement angle

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	✓	×	×	×	×

Description:

This function sets the measurement angle.
If at least one angle is specified, the angle(s) can be set with any combination.

CMMISDK_GetMeasurementAngle: Obtains the measurement angle.

Format:

error_km CMMISDK_GetMeasurementAngle(int32_km inInstrumentNo, CMMISDK_MeasAngle* outAngle)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outAngle	O	Measurement angle

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	✓	×	×	×	×

Description:

This function obtains the measurement angle.

CMMISDK_SetTiltDetection: Sets tilt detection.

Format:

error_km CMMISDK_SetTiltDetection(int32_km inInstrumentNo, CMMISDK_OnOff inDetection)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inDetection	I	Tilt detection

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	✓	×	×	×	×

Description:

This function sets tilt detection.
If tilt detection is on, an error will be output when tilting is detected that exceeds a certain degree during measurements.

CMMISDK_GetTiltDetection: Obtains whether tilt detection is enabled or disabled.

Format:

error_km CMMISDK_GetTiltDetection(int32_km inInstrumentNo, CMMISDK_OnOff* outDetection)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outDetection	O	Tilt detection

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	✓	×	×	×	×

Description:

This function obtains whether tilt detection is enabled or disabled.

CMMISDK_SetMeasurementMode: Sets the measurement mode.

Format:

error_km CMMISDK_SetMeasurementMode(int32_km inInstrumentNo, CMMISDK_MeasMode inMode)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inMode	I	Measurement mode

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	×	✓	✓	✓	✓

Description:

This function sets the measurement mode.

Calibration may be required again when conditions are changed.
Use CMMISDK_GetCalibrationStatus to determine whether calibration is necessary.

CMMISDK_GetMeasurementMode: Obtains the measurement mode.

Format:

error_km CMMISDK_GetMeasurementMode(int32_km inInstrumentNo, CMMISDK_MeasMode* outMode)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outMode	O	Measurement mode

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	×	✓	✓	✓	✓

Description:

This function obtains the measurement mode.

CMMISDK_SetSpecularComponent: Sets the specular component.

Format:

error_km CMMISDK_SetSpecularComponent(int32_km inInstrumentNo, CMMISDK_SpecularComponent inSpecularComponent)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inSpecularComponent	I	Specular component

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	✓	✓

Description:

This function sets the specular component.

Calibration may be required again when conditions are changed.
Use CMMISDK_GetCalibrationStatus to determine whether calibration is necessary.

CMMISDK_GetSpecularComponent: Obtains the specular component.

Format:

error_km CMMISDK_GetSpecularComponent(int32_km inInstrumentNo, CMMISDK_SpecularComponent* outSpecularComponent)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outSpecularComponent	O	Specular component

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	✓	✓

Description:

This function obtains the specular component.

CMMISDK_SetUv: Sets the UV condition.

Format:

error_km CMMISDK_SetUv(int32_km inInstrumentNo, CMMISDK_Uv inUv)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inUv	I	UV condition

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	×	×

* Dependent on instrument and version.

Description:

This function sets the UV condition.

CMMISDK_GetUv: Obtains the UV condition.

Format:

error_km CMMISDK_GetUv(int32_km inInstrumentNo, CMMISDK_Uv* outUv)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outUv	O	UV condition

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	×	×

* Dependent on instrument and version.

Description:

This function obtains the UV condition.

CMMISDK_SetAutoAverageTimes: Sets the number of times of automatic averaging.

Format:

error_km CMMISDK_SetAutoAverageTimes(int32_km inInstrumentNo, int32_km inTimes)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inTimes	I	Number of times of automatic averaging (1-10)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function sets the number of times of automatic averaging.

CMMISDK_PerformMeasurement function executes measurements for the number of times specified in this condition.
CMMISDK_CancelMeasurement can be used when the number of times of automatic averaging is set to multiple times.

CMMISDK_GetAutoAverageTimes: Obtains the number of times of automatic averaging.

Format:

error_km CMMISDK_GetAutoAverageTimes(int32_km inInstrumentNo, int32_km* outTimes)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outTimes	O	Number of times of automatic averaging (1-10)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the number of times of automatic averaging.

CMMISDK_SetManualAverageTimes: Sets the number of times of manual averaging.

Format:

error_km CMMISDK_SetManualAverageTimes(int32_km inInstrumentNo, int32_km inTimes)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inTimes	I	Manual average count * The range of the setting is dependent on the instrument and version. Refer to Appendix C.

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function sets the number of times of manual averaging.

This function is used when performing standalone measurements. It is not applied when communication is used.

CMMISDK_GetManualAverageTimes: Obtains the number of times of manual averaging.

Format:

error_km CMMISDK_GetManualAverageTimes(int32_km inInstrumentNo, int32_km* outTimes)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outTimes	O	Manual average count

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the number of times of manual averaging.

CMMISDK_SetManualAverageSaveMode: Sets the manual averaging save method.

Format:

error_km CMMISDK_SetManualAverageSaveMode(int32_km inInstrumentNo, CMMISDK_SaveMode inMode)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inMode	I	Save method

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function sets the manual averaging save method.

CMMISDK_GetManualAverageSaveMode: Obtains the manual averaging save method.

Format:

error_km CMMISDK_GetManualAverageSaveMode(int32_km inInstrumentNo, CMMISDK_SaveMode* outMode)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outMode	O	Manual averaging save method

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the manual averaging save method.

CMMISDK_SetCondSMC: Sets the SMC conditions.

Format:

error_km CMMISDK_SetCondSMC(int32_km inInstrumentNo, const CMMISDK_CondSMC* inCond)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inCond	I	SMC conditions

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	✓	×

Description:

This function sets the SMC conditions.

This condition is only applied to standalone measurements, and not applied when communication is used.

CMMISDK_GetCondSMC: Obtains the SMC conditions.

Format:

error_km CMMISDK_GetCondSMC(int32_km inInstrumentNo, CMMISDK_CondSMC* outCond)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outCond	O	SMC conditions

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	✓	×

Description:

This function obtains the SMC conditions.

4.5 Display conditions (instruments settings)

CMMISDK_SetDisplayType: Sets the display type.

Format:

error_km CMMISDK_SetDisplayType(int32_km inInstrumentNo, CMMISDK_DisplayType inDisplayType)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inDisplayType	I	Display Type

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	×

Description:

This function sets the display type.
The display type can be set with any combination.

CMMISDK_GetDisplayType: Obtains the display type.

Format:

error_km CMMISDK_GetDisplayType(int32_km inInstrumentNo, CMMISDK_DisplayType* outDisplayType)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outDisplayType	O	Display Type

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	×

Description:

This function obtains the display type.

CMMISDK_SetObserverAndIlluminant: Sets the observation field and illuminant.

Format:

error_km CMMISDK_SetObserverAndIlluminant(int32_km inInstrumentNo, int32_km inNum, CMMISDK_Observer inObs, CMMISDK_Illuminant inIll)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inNum	I	Observation field / illuminant number (0-1)
inObs	I	Observer
inIll	I	Illuminant

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function sets the observation field and illuminant.
Setting the illuminant to 'None' when the observation field / illuminant number is 1 enables the use of one observation field and illuminant set.

When using a user illuminant, use CMMISDK_SetUserIlluminant to register illuminant data in advance.

CMMISDK_GetObserverAndIlluminant: Obtains the observation field and illuminant.

Format:

error_km CMMISDK_GetObserverAndIlluminant(int32_km inInstrumentNo, int32_km inNum, CMMISDK_Observer* outObs, CMMISDK_Illuminant* outIll)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inNum	I	Observation field / illuminant number (0-1)
outObs	O	Observer
outIll	O	Illuminant

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the observation field and illuminant.

CMMISDK_SetUserIlluminant: Sets the user illuminant.

Format:

error_km CMMISDK_SetUserIlluminant(int32_km inInstrumentNo, const CMMISDK_UserIlluminant* inIllData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inIllData	I	Illuminant data

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	×

Description:

This function sets the user illuminant.

Set the illuminant data as 85 items of data between 360 to 780 nm (5-nm pitch).

The name setting is ignored for models that do not allow the name to be set.

CMMISDK_GetUserIlluminant: Obtains the user illuminant.

Format:

error_km CMMISDK_GetUserIlluminant(int32_km inInstrumentNo, CMMISDK_UserIlluminant* outIllData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outIllData	O	Illuminant data

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	×

Description:

This function obtains the user illuminant.

CMMISDK_SetColorSpace: Sets the color space.

Format:

error_km CMMISDK_SetColorSpace(int32_km inInstrumentNo, CMMISDK_ColorSpace inColorSpace)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inColorSpace	I	Color space

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function sets the color space.

CMMISDK_GetColorSpace: Obtains the color space.

Format:

error_km CMMISDK_GetColorSpace(int32_km inInstrumentNo, CMMISDK_ColorSpace* outColorSpace)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outColorSpace	O	Color space

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the color space.

CMMISDK_SetEquation: Sets the color difference equation.

Format:

error_km CMMISDK_SetEquation(int32_km inInstrumentNo, CMMISDK_Equation inEquation)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inEquation	I	Color difference equation

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function sets the color difference equation.

CMMISDK_GetEquation: Obtains the color difference equation.

Format:

error_km CMMISDK_GetEquation(int32_km inInstrumentNo, CMMISDK_Equation* outEquation)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outEquation	O	Color difference equation

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the color difference equation.

CMMISDK_SetCustomIndex: Sets the custom display items.

Format:

error_km CMMISDK_SetCustomIndex(int32_km inInstrumentNo, int32_km inCustomNum, CMMISDK_CustomIndex inCustomIndex)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inCustomNum	I	Custom display number (14 numbers, 0-13)
inCustomIndex	I	Custom display item

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	×	✓	✓	✓	✓

Description:

This function sets the custom display items.
To display custom items, use CMMISDK_SetDisplayType to enable 'custom.'

CMMISDK_GetCustomIndex: Obtains the custom display items.

Format:

error_km CMMISDK_GetCustomIndex(int32_km inInstrumentNo, int32_km inCustomNum, CMMISDK_CustomIndex* outCustomIndex)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inCustomNum	I	Custom display number (14 numbers, 0-13)
outCustomIndex	O	Custom display item

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	×	✓	✓	✓	✓

Description:

This function obtains the custom display items.

CMMISDK_SetDirection: Sets the irradiation direction to display.

Format:

error_km CMMISDK_SetDirection(int32_km inInstrumentNo, CMMISDK_Direction inDirection)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
indirection	I	Irradiation direction to display

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	✓	×	×	×	×

Description:

This function sets the irradiation direction to display.
The irradiation direction to display may not be settable depending on the version of the instrument, even if the instrument itself is supported.

CMMISDK_GetDirection: Obtains the irradiation direction to display.

Format:

error_km CMMISDK_GetDirection(int32_km inInstrumentNo, CMMISDK_Direction* outDirection)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outDirection	O	Irradiation direction to display

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	✓	×	×	×	×

Description:

This function obtains the irradiation direction to display.

CMMISDK_SetUserEquation: Sets the user index.

Format:

error_km CMMISDK_SetUserEquation(int32_km inInstrumentNo, int32_km inNum, const CMMISDK_UserEquation* inEquation)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inNum	I	User index number (0-2)
inEquation	I	User index information

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	×	✓	✓	✓	✓

Description:

This function sets the user index.

The color space values that can be used for a user index are only the items that are selected in CMMISDK_SetColorSpace, CMMISDK_SetEquation, and CMMISDK_SetCustomIndex.

Sample user index: Equivalent to ΔE*ab

SQRT(POW([DL])+POW([DA])+POW([DB]))

The following variables can be used for a user index.

[L]	L*	[DL]	ΔL*	[MI]	MI
[A]	a*	[DA]	Δa*	[WIE]	WI(E313-73)
[B]	b*	[DB]	Δb*	[DWIE]	ΔWI(E313-73)
[C]	C*	[DC]	ΔC*	[WIC]	WI(CIE)
[H]	H	[DH]	ΔH*	[DWIC]	ΔWI(CIE)
[HL]	L(Hunter)	[DHL]	ΔL	[TINT]	Tint
[HA]	a(Hunter)	[DHA]	Δa	[DTINT]	ΔTint
[HB]	b(Hunter)	[DHB]	Δb	[YIE]	YI(E313-73)
[X]	X	[DX]	ΔX	[DYIE]	ΔYI(E313-73)
[Y]	Y	[DY]	ΔY	[YID]	YI(D1925)
[Z]	Z	[DZ]	ΔZ	[DYID]	ΔYI(D1925)
[SX]	x	[DSX]	Δx	[BISO]	B
[SY]	y	[DSY]	Δy	[DBISO]	ΔB
[GU]	GU	[DGU]	ΔGU		
[DE]	ΔE*ab	[DE94]	ΔE*94	[DEH]	ΔE(Hunter)
[CMC]	CMC	[DE00]	ΔE00		

* For the CM-26dG/26d/25d/23d, when items are limited by SCI and SCE, add "I" and "E" to the variables.

E.g.) To calculate L*(SCI)+L*(SCE), set [LI]+[LE].

The operators and functions that can be used for a user index are as follows:

+	A+B	A+B

-	A-B	A-B
*	A*B	A×B
/	A/B	A÷B
POW	POW(A)	Square of A
SQRT	SQRT(A)	Square root of A
ABS	ABS(A)	Absolute value of A
SIN	SIN(A)	Sine of A (degree)
COS	COS(A)	Cosine of A (degree)
TAN	TAN(A)	Tangent of A (degree)
ASIN	ASIN(A)	Arcsine (degree)
ACOS	ACOS(A)	Arccosine (degree)
ATAN	ATAN(A)	Arctangent (degree)
LOG	LOG(A)	Common logarithm
LN	LN(A)	Natural logarithm
EXP	EXP(A)	Exponential function
POW2	POW2(A,B)	Power function (A to the power of B)

Input format of the user classes

CLASS (n, "str1", d1, "str2", d2, "str3", d3, "str4", d4, "str5", …)

n: Indicates the number of threshold values that separate the classes. (Number of classes - 1). Specify this as a number within 50.

"str1",d1: When the judgment result of the user classes is d1 or higher, "str1" is displayed in the results field of the instrument.
d1 can be numeric value setting of 20 or fewer digits, but the effective digits in the calculation are 5 digits. Enter the threshold values from the left in order of the largest values, and always set a class to be displayed if a value is not classified into a threshold value greater than or equal to all of the threshold values.
The total length of characters that are entered (including CLASS()) must be within 200 single-byte characters.
Leave blank if the user classes will not be used.
Use "." for the decimal point and "," as the separator between parameters.

Input format of the user classes

CLASS (4, "A", 4, "B", 3, "C", 2, "D", 1, "E")

The result of the user index is split into 5 classes.

User index result	Class
4 or higher	A
3 or higher	B
2 or higher	C
1 or higher	D
Less than 1	E

CMMISDK_GetUserEquation: Obtains the user index.

Format:

error_km CMMISDK_GetUserEquation(int32_km inInstrumentNo, int32_km inNum, CMMISDK_UserEquation* outEquation)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inNum	I	User index number (0-2)
outEquation	O	User index information

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	×	✓	✓	✓	✓

Description:

This function obtains the user index.

4.6 Data

CMMISDK_SetActiveTarget: Sets the active target number.

Format:

error_km CMMISDK_SetActiveTarget(int32_km inInstrumentNo, int32_km inNum)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inNum	I	Target number

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function sets the active target number.
The active target is used as a target number associated for measurements after setting.

CMMISDK_GetActiveTarget: Obtains the active target number.

Format:

error_km CMMISDK_GetActiveTarget(int32_km inInstrumentNo, int32_km* outNum)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outNum	O	Target number

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the active target number.

CMMISDK_GetSavedTargetList: Obtains the list of saved target numbers.

Format:

error_km CMMISDK_GetSavedTargetList(int32_km inInstrumentNo, CMMISDK_SavedTargetList* outList)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outList	O	List of saved target numbers

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the list of saved target numbers.

CMMISDK_GetTargetListInFilter: Obtains the list of target numbers when the display filter is applied.

Format:

error_km CMMISDK_GetTargetListInFilter(int32_km inInstrumentNo, CMMISDK_SavedTargetList* outList)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outList	O	List of target numbers when the display filter is applied

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	×	✓	✓	✓	✓

Description:

This function obtains the list of target numbers when the display filter is applied.
Set the display filter conditions with CMMISDK_SetTargetFilter.

CMMISDK_DeleteTargetData: Deletes the target.

Format:

error_km CMMISDK_DeleteTargetData(int32_km inInstrumentNo, int32_km inNum)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inNum	I	Target number

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function deletes the target for the specified number.

CMMISDK_DeleteAllTargetData: Deletes all targets.

Format:

error_km CMMISDK_DeleteAllTargetData(int32_km inInstrumentNo)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function deletes all targets.

CMMISDK_ClearTargetInfo: Clears target information on the PC.

Format:

error_km CMMISDK_ClearTargetInfo(int32_km inInstrumentNo)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function clears target information on the PC.
Target information means properties, data, tolerances, and parametric coefficients.
When cleared, tolerances and parametric coefficients are set to the default settings. When setting information, handle only the item to change.

Writing target information to the instrument can be handled by setting the information using the CMMISDK_SetTargetProperty, CMMISDK_SetTargetData, CMMISDK_SetToleranceForTarget, and CMMISDK_SetParametricForTarget functions, and then using CMMISDK_SaveTargetInfo.

CMMISDK_LoadTargetInfo: Loads target information on the PC.

Format:

error_km CMMISDK_LoadTargetInfo(int32_km inInstrumentNo, int32_km inNum)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inNum	I	Target number

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErNoData	No data
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function loads target information on the PC.
Target information means properties, data, tolerances, and parametric coefficients.

To obtain information using the CMMISDK_GetTargetProperty, CMMISDK_GetTargetData, CMMISDK_GetToleranceForTarget, and CMMISDK_GetParametricForTarget functions, always load the information on the PC using this function.

CMMISDK_SaveTargetInfo: Saves target information on the PC to the instrument.

Format:

error_km CMMISDK_SaveTargetInfo(int32_km inInstrumentNo, int32_km inNum)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inNum	I	Target number

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function saves target information on the PC to the instrument.
Target information means properties, data, tolerances, and parametric coefficients.
Refer to "3.2 Basic processing flow" for the procedure.

Before saving the information, use the CMMISDK_SetTargetProperty and CMMISDK_SetTargetData functions to set the necessary information.
Setting information with CMMISDK_SetToleranceForTarget and CMMISDK_SetParametricForTarget is optional.

CMMISDK_SetTargetProperty: Sets target information properties.

Format:

error_km CMMISDK_SetTargetProperty(int32_km inInstrumentNo, const CMMISDK_TargetProperty* inProperty)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inProperty	I	Target properties

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function sets target information properties on the PC.
To apply target information to the instrument, set the target information (properties, data, tolerances, and parametric coefficients) and execute CMMISDK_SaveTargetInfo.

The following table gives the properties that must be set for each instrument. Entered information is ignored for cells indicated by "-".

	25cG	M6	26dG	26d	25d	23d
date	✓	✓	✓	✓	✓	✓
group_list	✓	-	✓	✓	✓	✓
meas_type	-	-	-	-	-	-
meas_mode	✓	-	✓	✓	✓	✓
meas_area	✓	-	✓	✓	✓	✓
meas_angle	-	✓	-	-	-	-
meas_ldirection	-	✓	-	-	-	-
meas_scie	-	-	✓	✓	✓	✓
meas_uv	-	-	✓	✓	✓	✓
warning_level	✓	✓	✓	✓	✓	✓
warning	✓	✓	✓	✓	✓	✓
diagnosis	✓	✓	✓	✓	✓	✓
data_attr	✓	Spectral only	✓	✓	✓	✓
name	30 characters max.	16 characters max.		30 characters max.		

* If the character length is exceeded, the name is stored only up to the corresponding character length.
* With opacity, the color value cannot be registered.
* For 25d and 23d data, set AREA_MAV for mea_area, and UV_CUT400 for meas_uv.

If spectral is specified in data_attr, use CMMISDK_SetTargetData to set the data.
If a value other than spectral is specified in data_attr, use CMMISDK_SetTargetDataColor to set the

data.

CMMISDK_GetTargetProperty: Obtains target information properties.

Format:

error_km CMMISDK_GetTargetProperty(int32_km inInstrumentNo, CMMISDK_TargetProperty* outProperty)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outProperty	O	Target properties

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErNoData	No data
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains target information properties on the PC.
The data for the specified number on the PC can be obtained by using CMMISDK_LoadTargetInfo.

If data_attr is spectral, use CMMISDK_GetTargetData to obtain the data.
If data_attr is a value other than spectral, use CMMISDK_GetTargetDataColor to obtain the data.

CMMISDK_SetTargetData: Sets target information data.

Format 1:

error_km CMMISDK_SetTargetData(int32_km inInstrumentNo, CMMISDK_DataType inDataType, const CMMISDK_Data* inData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inDataType	I	Data type
inData	I	Reflectance data * Store the data from the beginning of the array in the amount of DataSize that was obtained with CMMISDK_GetInstrumentInfo.

Format 2:

error_km CMMISDK_SetTargetDataColor(int32_km inInstrumentNo, CMMISDK_DataType inDataType, int32_km inNum, const CMMISDK_ColorCond* inCond, const CMMISDK_Data* inData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inDataType	I	Data type
inNum	I	Observation field / illuminant number (0-1)
inCond	I	Color value calculation conditions
inData	I	Color value data * Store data from the beginning of the array in the amount of the number of color values.

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function sets target information data on the PC.
To apply target information to the instrument, set the target information (properties, data, tolerances, and parametric coefficients) and execute CMMISDK_SaveTargetInfo.

The data that must be set for each instrument is determined by the following conditions. Set the necessary data according to the conditions.

	25cG	M6	26dG	26d	25d	23d
meas_mode	✓	-	✓	✓	✓	✓
meas_angle	-	✓	-	-	-	-
meas_ldirection	-	✓	-	-	-	-
meas_scie	-	-	✓	✓	✓	✓

* For the CM-M6
Six angles of data must always be set for each meas_ldirection.
If the data is insufficient, KmErInvalidParameter will be returned.

CMMISDK_GetTargetData: Obtains target information data.

Format:

error_km CMMISDK_GetTargetData(int32_km inInstrumentNo, CMMISDK_DataType inDataType, CMMISDK_Data* outData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inDataType	I	Data type
outData	O	Reflectance data

Format:

error_km CMMISDK_GetTargetDataColor(int32_km inInstrumentNo, CMMISDK_DataType inDataType, int32_km inNum, CMMISDK_ColorCond* outCond, CMMISDK_Data* outData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inDataType	I	Data type
inNum	I	Observation field / illuminant number (0-1)
outCond	O	Observation field / illuminant information
outData	O	Color value data * Data in the amount of the number of color values is stored from the beginning of the array.

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErNoData	No data
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains target information data on the PC.
The data for the specified number on the PC can be obtained by using CMMISDK_LoadTargetInfo.

The data that can be obtained with each instrument is determined by the following conditions obtained with CMMISDK_GetTargetProperty.

	25cG	M6	26dG	26d	25d	23d
meas_mode	✓	-	✓	✓	✓	✓
meas_angle	-	✓	-	-	-	-
meas_Idirection	-	✓	-	-	-	-
meas_scie	-	-	✓	✓	✓	✓

For example, the CM-25cG has the following three output patterns depending on the measurement mode: reflectance and gloss value, reflectance only, and gloss only.

CMMISDK_SetToleranceForTarget: Sets the tolerance for target information.

Format:

error_km CMMISDK_SetToleranceForTarget(int32_km inInstrumentNo, CMMISDK_ToleranceType inType, int32_km inObsIll, CMMISDK_ToleranceId inId, const CMMISDK_ToleranceData* inTolerance)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inType	I	Tolerance type
inObsIll	I	Observation field / illuminant number (0-1)
inId	I	Tolerance ID
inTolerance	I	Tolerance data

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function sets the tolerance for the target on the PC.

To apply target information to the instrument, set the target information (properties, data, tolerances, and parametric coefficients) and execute CMMISDK_SaveTargetInfo.

CMMISDK_GetToleranceForTarget: Obtains the tolerance for the target information.

Format:

error_km CMMISDK_GetToleranceForTarget(int32_km inInstrumentNo, CMMISDK_ToleranceType inType, int32_km inObsIll, CMMISDK_ToleranceId inId, CMMISDK_ToleranceData* outTolerance)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inType	I	Tolerance type
inObsIll	I	Observation field / illuminant number (0-1)
inId	I	Tolerance ID
outTolerance	O	Tolerance data

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErNoData	No data
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the tolerance for the target information on the PC.
The data for the specified number on the PC can be obtained by using CMMISDK_LoadTargetInfo.

CMMISDK_SetParametricForTarget: Sets the parametric coefficient for the target information.

Format:

error_km CMMISDK_SetParametricForTarget(int32_km inInstrumentNo, CMMISDK_ToleranceType inType, CMMISDK_ParametricId inId, const CMMISDK_ParametricCoef* inParametric)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inType	I	Tolerance type
inId	I	Parametric coefficient ID
inParametric	I	Parametric coefficient

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function sets the tolerance for the target on the PC.

To apply target information to the instrument, set the target information (properties, data, tolerances, and parametric coefficients) and execute CMMISDK_SaveTargetInfo.

CMMISDK_GetParametricForTarget: Obtains the parametric coefficient for the target information.

Format:

error_km CMMISDK_GetParametricForTarget(int32_km inInstrumentNo, CMMISDK_ToleranceType inType, CMMISDK_ParametricId inId, CMMISDK_ParametricCoef* outParametric)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inType	I	Tolerance type
inId	I	Parametric coefficient ID
outParametric	O	Parametric coefficient

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErNoData	No data
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the parametric coefficient for the target information on the PC.
The data for the specified number on the PC can be obtained by using CMMISDK_LoadTargetInfo.

CMMISDK_SetTargetFilter: Sets the target filter conditions.

Format:

error_km CMMISDK_SetTargetFilter(int32_km inInstrumentNo, CMMISDK_FilterIndex inIndex, const CMMISDK_GroupList* inGroup)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inIndex	I	Filter index
inGroup	I	Group list * This item is only used when the filter index is 'group.' Otherwise, set this to 0.

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	×	✓	✓	✓	✓

Description:

This function sets the target filter conditions.

If the filter is not set to OFF, only the target data with the number that matches the condition is displayed.

CMMISDK_GetTargetFilter: Obtains the target filter conditions.

Format:

error_km CMMISDK_GetTargetFilter(int32_km inInstrumentNo, CMMISDK_FilterIndex* outIndex, CMMISDK_GroupList* outGroup)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outIndex	O	Filter index
outGroup	O	Group list * This item is only used when the filter index is 'group.' Otherwise, set this to 0.

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	×	✓	✓	✓	✓

Description:

This function obtains the target filter conditions.

CMMISDK_SetTargetProtect: Sets target protection.

Format:

error_km CMMISDK_SetTargetProtect(int32_km inInstrumentNo, CMMISDK_OnOff inProtect)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inProtect	I	Protection ON/OFF

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function sets target protection.
When target protection is turned on, only new measurement (save) operations are allowed.

CMMISDK_GetTargetProtect: Obtains target protection.

Format:

error_km CMMISDK_GetTargetProtect(int32_km inInstrumentNo, CMMISDK_OnOff* outProtect)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outProtect	O	Protection ON/OFF

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the target protection setting.

CMMISDK_GetSavedSampleCount: Obtains the number of saved measurement values.

Format:

error_km CMMISDK_GetSavedSampleCount(int32_km inInstrumentNo, int32_km* outCount)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outCount	O	Number of saved measurement values

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the number of saved measurement values.

CMMISDK_DeleteSampleData: Deletes a measurement value.

Format:

error_km CMMISDK_DeleteSampleData(int32_km inInstrumentNo, int32_km inNum)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inNum	I	Measurement value number

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function deletes the measurement value for the specified number.

CMMISDK_DeleteAllSampleData: Deletes all measurement values.

Format:

error_km CMMISDK_DeleteAllSampleData(int32_km inInstrumentNo)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function deletes all measurement values.

CMMISDK_LoadSampleInfo: Loads measurement value information on the PC.

Format:

error_km CMMISDK_LoadSampleInfo(int32_km inInstrumentNo, int32_km inNum)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inNum	I	Measurement value number

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErNoData	No data
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function loads measurement value information on the PC.
Measurement value information means properties and data.

To obtain information using the CMMISDK_GetSampleProperty and CMMISDK_GetSampleData functions, always load the information on the PC using this function.

CMMISDK_GetSampleProperty: Obtains measurement value information properties.

Format:

error_km CMMISDK_GetSampleProperty(int32_km inInstrumentNo, CMMISDK_SampleProperty* outProperty)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outProperty	O	Measurement value properties

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErNoData	No data
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains measurement value information properties on the PC.
The data for the specified number on the PC can be obtained by using CMMISDK_LoadSampleInfo.

CMMISDK_GetSampleData: Obtains measurement value information data.

Format:

error_km CMMISDK_GetSampleData(int32_km inInstrumentNo, CMMISDK_DataType inDataType, CMMISDK_Data* outData)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inDataType	I	Data type
outData	O	Reflectance data

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErNoData	No data
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains measurement value information data on the PC.
The data for the specified number on the PC can be obtained by using CMMISDK_LoadSampleInfo.

The data that can be obtained with each instrument is determined by the following conditions obtained with CMMISDK_GetSampleProperty.

	25cG	M6	26dG	26d	25d	23d
meas_mode	✓	-	✓	✓	✓	✓
meas_angle	-	✓	-	-	-	-
meas_Idirection	-	✓	-	-	-	-
meas_scie	-	-	✓	✓	✓	✓

For example, the CM-25cG has the following three output patterns depending on the measurement mode: reflectance and gloss value, reflectance only, and gloss only.

4.7 Other functions (instruments settings)

CMMISDK_SetActiveGroup: Sets the active group number.

Format:

error_km CMMISDK_SetActiveGroup(int32_km inInstrumentNo, const CMMISDK_GroupList* inGroup)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inGroup	I	Group list

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	×	✓	✓	✓	✓

Description:

This function sets the active group number.
The active group number is used as a group number associated for measurements after setting.

CMMISDK_GetActiveGroup: Obtains the active group number.

Format:

error_km CMMISDK_GetActiveGroup(int32_km inInstrumentNo, CMMISDK_GroupList* outGroup)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outGroup	O	Group list

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	×	✓	✓	✓	✓

Description:

This function obtains the active group number.

CMMISDK_SetGroupName: Sets the group name.

Format:

error_km CMMISDK_SetGroupName(int32_km inInstrumentNo, int32_km inGroup, const CMMISDK_Group* inName)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inGroup	I	Group number (1-50)
inName	I	Group name

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	×	✓	✓	✓	✓

Description:

This function sets the group name for the specified number.

CMMISDK_GetGroupName: Obtains the group name.

Format:

error_km CMMISDK_GetGroupName(int32_km inInstrumentNo, int32_km inGroup, CMMISDK_Group* outName)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inGroup	I	Group number (1-50)
outName	O	Group name

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	×	✓	✓	✓	✓

Description:

This function obtains the group name corresponding to the specified number.

CMMISDK_SetMultipleGroupName: Sets group names in batch.

Format:

error_km CMMISDK_SetMultipleGroupName(int32_km inInstrumentNo, const CMMISDK_GroupAll* inName)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inName	I	All group names

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	×	✓	✓	✓	✓

Description:

This function sets all group names.

CMMISDK_GetMultipleGroupName: Obtains group names in batch.

Format:

error_km CMMISDK_GetMultipleGroupName(int32_km inInstrumentNo, CMMISDK_GroupAll* outName)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outName	O	All group names

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	×	✓	✓	✓	✓

Description:

This function obtains all group names.

CMMISDK_LoadDefaultInfo: Loads default information on the PC.

Format:

error_km CMMISDK_LoadDefaultInfo(int32_km inInstrumentNo)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function loads default information on the PC.
Default information means the defaults for tolerances and parametric coefficients.

To obtain information using the CMMISDK_GetTolerance and CMMISDK_GetParametric functions, always load the information on the PC using this function.

CMMISDK_SaveDefaultInfo: Saves default information on the PC to the instrument.

Format:

error_km CMMISDK_SaveDefaultInfo(int32_km inInstrumentNo)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function loads default information on the PC to the instrument.
Default information means the defaults for tolerances and parametric coefficients.
Refer to "3.2 Basic processing flow" for the procedure.

Before saving the information, use the CMMISDK_SetTolerance and CMMISDK_SetParametric functions to set the necessary information.

CMMISDK_SetTolerance: Sets tolerances in the default information.

Format:

error_km CMMISDK_SetTolerance(int32_km inInstrumentNo, CMMISDK_ToleranceType inType, int32_km inObsIll, CMMISDK_ToleranceId inId, const CMMISDK_ToleranceData* inTolerance)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inType	I	Tolerance type
inObsIll	I	Observation field / illuminant number (0-1)
inId	I	Tolerance ID
inTolerance	I	Tolerance data

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function sets the default tolerance on the PC.
To apply default information to the instrument, load the information on the PC with CMMISDK_LoadDefaultInfo, set the default information (tolerances and parametric coefficients) to change, and execute CMMISDK_SaveDefaultInfo.

CMMISDK_GetTolerance: Obtains tolerances in the default information.

Format:

error_km CMMISDK_GetTolerance(int32_km inInstrumentNo, CMMISDK_ToleranceType inType, int32_km inObsIll, CMMISDK_ToleranceId inId, CMMISDK_ToleranceData* outTolerance)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inType	I	Tolerance type
inObsIll	I	Observation field / illuminant number (0-1)
inId	I	Tolerance ID
outTolerance	O	Tolerance data

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErNoData	No data
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains tolerances in the default information on the PC.
The data can be obtained by using CMMISDK_LoadDefaultInfo.

CMMISDK_SetParametric: Sets parametric coefficients in the default information.

Format:

error_km CMMISDK_SetParametric(int32_km inInstrumentNo, CMMISDK_ToleranceType inType, CMMISDK_ParametricId inId, const CMMISDK_ParametricCoef* inParametric)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inType	I	Tolerance type
inId	I	Parametric coefficient ID
inParametric	I	Parametric coefficient

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function sets the default tolerance on the PC.
To apply default information to the instrument, load the information on the PC with CMMISDK_LoadDefaultInfo, set the default information (tolerances and parametric coefficients) to change, and execute CMMISDK_SaveDefaultInfo.

CMMISDK_GetParametric: Obtains parametric coefficients in the default information.

Format:

error_km CMMISDK_GetParametric(int32_km inInstrumentNo, CMMISDK_ToleranceType inType, CMMISDK_ParametricId inId, CMMISDK_ParametricCoef* outParametric)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inType	I	Tolerance type
inId	I	Parametric coefficient ID
outParametric	O	Parametric coefficient

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErNoData	No data
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains parametric coefficients in the default information on the PC.
The data can be obtained by using CMMISDK_LoadDefaultInfo.

CMMISDK_SetWarningLevel: Sets the warning level.

Format:

error_km CMMISDK_SetWarningLevel(int32_km inInstrumentNo, int32_km inLevel)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inLevel	I	Warning level (0 to 100%)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function sets the warning level.

CMMISDK_GetWarningLevel: Obtains the warning level.

Format:

error_km CMMISDK_GetWarningLevel(int32_km inInstrumentNo, int32_km* outLevel)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outLevel	O	Warning level (0 to 100%)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the warning level.

CMMISDK_SetInstrumentMode: Sets the instrument mode.

Format:

error_km CMMISDK_SetInstrumentMode(int32_km inInstrumentNo, CMMISDK_InstrumentMode inMode)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inMode	I	Instrument mode

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	✓	✓

Description:

This function sets the instrument mode.

When the power supply to the instrument is turned off, the setting reverts normal mode.

CMMISDK_GetInstrumentMode: Obtains the instrument mode.

Format:

error_km CMMISDK_GetInstrumentMode(int32_km inInstrumentNo, CMMISDK_InstrumentMode* outMode)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outMode	O	Instrument mode

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	✓	✓

Description:

This function obtains the instrument mode.

CMMISDK_SetUserType: Sets the user type.

Format:

error_km CMMISDK_SetUserType(int32_km inInstrumentNo, CMMISDK_UserType inType)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inType	I	User type

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function sets the user type.
It is applied during standalone use.

CMMISDK_GetUserType: Obtains the user type.

Format:

error_km CMMISDK_GetUserType(int32_km inInstrumentNo, CMMISDK_UserType* outType)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outType	O	User type

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the user type.

CMMISDK_SetAdminPassword: Sets the administrator password.

Format:

error_km CMMISDK_SetAdminPassword(int32_km inInstrumentNo, const CMMISDK_AdminPass* inPass)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inPass	I	Administrator password

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	✓	✓

Description:

This function sets the administrator password.

The password is used to change the settings from operator to administrator when the instrument is used standalone.

CMMISDK_GetAdminPassword: Obtains the administrator password.

Format:

error_km CMMISDK_GetAdminPassword(int32_km inInstrumentNo, CMMISDK_AdminPass* outPass)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outPass	O	Administrator password

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	✓	✓

Description:

This function obtains the administrator password.

CMMISDK_SetAutoPrint: Sets automatic printing.

Format:

error_km CMMISDK_SetAutoPrint(int32_km inInstrumentNo, CMMISDK_OnOff inPrint)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inPrint	I	Automatic printing ON/OFF

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function sets automatic printing.

CMMISDK_GetAutoPrint: Obtains the automatic printing setting.

Overview:

This function obtains the automatic printing setting.

Format:

error_km CMMISDK_GetAutoPrint(int32_km inInstrumentNo, CMMISDK_OnOff* outPrint)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outPrint	O	Automatic printing ON/OFF

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the automatic printing setting.

CMMISDK_SetBrightness: Sets the brightness of the display.

Format:

error_km CMMISDK_SetBrightness(int32_km inInstrumentNo, int32_km inBrightness)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inBrightness	I	Display brightness (0 to 4)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function sets the brightness of the display.

CMMISDK_GetBrightness: Obtains the brightness of the display.

Format:

error_km CMMISDK_GetBrightness(int32_km inInstrumentNo, int32_km* outBrightness)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outBrightness	O	Display brightness (0 to 4)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the brightness of the display.

CMMISDK_SetScreenDirection: Sets the display direction of the screen.

Format:

error_km CMMISDK_SetScreenDirection(int32_km inInstrumentNo, CMMISDK_ScreeDirection inScreenDirection)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inScreenDirection	I	Display direction

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	✓	✓	✓	✓	✓

Description:

This function sets the display direction of the screen.

CMMISDK_GetScreenDirection: Obtains the display direction of the screen.

Format:

error_km CMMISDK_GetScreenDirection(int32_km inInstrumentNo, CMMISDK_ScreeDirection* outScreenDirection)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outScreenDirection	O	Screen display direction

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	✓	✓	✓	✓	✓

Description:

This function obtains the display direction of the screen.

CMMISDK_SetSound: Sets the beep.

Format:

error_km CMMISDK_SetSound(int32_km inInstrumentNo, CMMISDK_OnOff inSound)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inSound	I	Volume setting

Return Value:

Definition value	説明
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function sets the beep.
When the sound is turned on, the instrument will generate a sound when calibration or measurement is completed or ends in an error.

CMMISDK_GetSound: Obtains the beep.

Format:

error_km CMMISDK_GetSound(int32_km inInstrumentNo, CMMISDK_OnOff* outSound)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outSound	O	Volume setting

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the beep.

CMMISDK_SetCalibrationInterval: Sets the calibration interval.

Format:

error_km CMMISDK_SetCalibrationInterval(int32_km inInstrumentNo, int32_km inInterval)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inInterval	I	Calibration interval (1-24) (1-hr pitch)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function sets the calibration interval.
Note that, although it is possible to run a measurement after the calibration interval ends, the instrument is in an alert state where calibration is recommended.

CMMISDK_GetCalibrationInterval: Obtains the calibration interval.

Format:

error_km CMMISDK_GetCalibrationInterval(int32_km inInstrumentNo, int32_km* outInterval)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outInterval	O	Calibration interval (1-24) (1-hr pitch)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the calibration interval.

CMMISDK_SetAnnualCalibration: Sets the periodical calibration notice.

Format:

error_km CMMISDK_SetAnnualCalibration(int32_km inInstrumentNo, CMMISDK_OnOff inCal)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inCal	I	Periodical calibration notice ON/OFF

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function sets the periodical calibration notice.

CMMISDK_GetAnnualCalibration: Obtains the periodical calibration notice.

Format:

error_km CMMISDK_GetAnnualCalibration(int32_km inInstrumentNo, CMMISDK_OnOff* outCal)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outCal	O	Periodical calibration notice ON/OFF

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the periodical calibration notice setting.

CMMISDK_SetZeroCalibrationSkip: Sets whether or not to skip zero calibration.

Format:

error_km CMMISDK_SetZeroCalibrationSkip(int32_km inInstrumentNo, CMMISDK_OnOff inSkip)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inSkip	I	Skip zero calibration on/off

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	✓	✓

Description:

This function sets whether or not to skip zero calibration.
If skip zero calibration is turned on, zero calibration can be skipped when the instrument starts. If turned off, zero calibration must always be performed.

CMMISDK_GetZeroCalibrationSkip: Obtains the skip zero calibration setting.

Format:

error_km CMMISDK_GetZeroCalibrationSkip(int32_km inInstrumentNo, CMMISDK_OnOff* outSkip)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outSkip	O	Skip zero calibration on/off

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	✓	✓

Description:

This function obtains whether skip zero calibration is on or off.

CMMISDK_SetDateTime: Sets the date and time.

Format:

error_km CMMISDK_SetDateTime(int32_km inInstrumentNo, const CMMISDK_DateTime* inDate)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inDate	I	Date/time

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function sets the date and time.

CMMISDK_SetDateFormat: Sets the date format.

Format:

error_km CMMISDK_SetDateFormat(int32_km inInstrumentNo, CMMISDK_DateFormat inFormat)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inFormat	I	Date format

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function sets the date format.

CMMISDK_GetDateFormat: Obtains the date format.

Format:

error_km CMMISDK_GetDateFormat(int32_km inInstrumentNo, CMMISDK_DateFormat* outFormat)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outFormat	O	Date format

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the date format.

CMMISDK_SetLanguage: Sets the display language.

Format:

error_km CMMISDK_SetLanguage(int32_km inInstrumentNo, CMMISDK_Language inLanguage)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inLanguage	I	Display language

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function sets the display language.

CMMISDK_GetLanguage: Obtains the display language.

Format:

error_km CMMISDK_GetLanguage(int32_km inInstrumentNo, CMMISDK_Language* outLanguage)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outLanguage	O	Display language

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function obtains the display language.

CMMISDK_SetPowerSaving: Sets the time to switch to power saving mode.

Format:

error_km CMMISDK_SetPowerSaving(int32_km inInstrumentNo, int32_km inPowerSaving)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inPowerSaving	I	Time to switch to power saving mode (0 to 60 minutes) * 0 minutes is treated as off

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
△	△	✓	✓	✓	✓

* Dependent on instrument and version.

Description:

This function sets the time to switch to power saving mode.

CMMISDK_GetPowerSaving: Obtains the time to switch to power saving mode.

Format:

error_km CMMISDK_GetPowerSaving(int32_km inInstrumentNo, int32_km* outPowerSaving)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
outPowerSaving	O	Time to switch to power saving mode (0 to 60 minutes) * 0 minutes is treated as off

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
△	△	✓	✓	✓	✓

* Dependent on instrument and version.

Description:

This function obtains the time to switch to power saving mode.

CMMISDK_ClearJobInfo: Clears job information.

Format:

error_km CMMISDK_ClearJobInfo(int32_km inInstrumentNo, int32_km inJobNum)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inJobNum	I	Job number (0 to 4)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	✓	×

Description:

This function clears job information.

It clears the step information and image information registered to the specified jog number.

CMMISDK_SetJobInfo: Sets job information.

Format:

error_km CMMISDK_SetJobInfo(int32_km inInstrumentNo, int32_km inJobNum, const CMMISDK_JobInfo* inInfo)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inJobNum	I	Job number (0 to 4)
inInfo	I	Job information

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	✓	×

Description:

This function sets job information.
Use CMMISDK_SetJobStepForOperation and CMMISDK_SetJobStepForResult to set the number of steps specified here.
Refer to "3.2 Basic processing flow" for the procedure.
After the job information is registered to the instrument, perform trial operation to determine if the job has been configured appropriately before putting the job into operation.

CMMISDK_GetJobInfo: Obtains job information.

Format:

error_km CMMISDK_GetJobInfo(int32_km inInstrumentNo, int32_km inJobNum, CMMISDK_JobInfo* outInfo)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inJobNum	I	Job number (0 to 4)
outInfo	O	Job information

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	✓	×

Description:

This function obtains job information.

CMMISDK_GetJobStepType: Obtains the step type of the job.

Format:

error_km CMMISDK_GetJobStepType(int32_km inInstrumentNo, int32_km inJobNum, int32_km inStepNum, CMMISDK_JobStepType* outType)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inJobNum	I	Job number (0 to 4)
inSetpNum	I	Step number (0 to 19)
outType	O	Step type

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	✓	×

Description:

This function obtains the step type of the job.

If the job type is operation step, use CMMISDK_GetJobStepForOperation. If the job type is result step, use CMMISDK_GetJobStepForResult.

CMMISDK_SetJobStepForOperation: Sets an operation step of the job.

Format:

error_km CMMISDK_SetJobStepForOperation(int32_km inInstrumentNo, int32_km inJobNum, int32_km inStepNum, const CMMISDK_JobStepOperation* inOperation)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inJobNum	I	Job number (0 to 4)
inSetpNum	I	Step number (0 to 19)
inOperation	I	Step content (operation step)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	✓	×

Description:

This function sets an operation step of the job.
The steps (operation steps or result steps) must be set in the amount of the number of steps specified by CMMISDK_SetJobInfo.
Refer to "3.2 Basic processing flow" for the procedure.

The following table gives the items that must be set for each instrument. Entered information is ignored for cells indicated by "-".

	26dG	26d	25d
meas_type	-	-	-
meas_mode	✓	-	-
meas_area	✓	✓	-
meas_angle	-	-	-
meas_ldirection	-	-	-
meas_scie	✓	✓	✓
meas_uv	✓	✓	-

CMMISDK_GetJobStepForOperation: Obtains an operation step of the job.

Format:

error_km CMMISDK_GetJobStepForOperation(int32_km inInstrumentNo, int32_km inJobNum, int32_km inStepNum, CMMISDK_JobStepOperation* outOperation)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inJobNum	I	Job number (0 to 4)
inSetpNum	I	Step number (0 to 19)
outOperation	O	Step content (operation step)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	✓	×

Description:

This function obtains an operation step of the job.

CMMISDK_SetJobStepForResult: Sets a result step of the job.

Format:

error_km CMMISDK_SetJobStepForResult(int32_km inInstrumentNo, int32_km inJobNum, int32_km inStepNum, const CMMISDK_JobStepResult* inResult)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inJobNum	I	Job number (0 to 4)
inSetpNum	I	Step number (0 to 19)
inResult	I	Step content (result step)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	✓	×

Description:

This function sets a result step of the job.

The steps (operation steps or result steps) must be set in the amount of the number of steps specified by CMMISDK_SetJobInfo.

Refer to "3.2 Basic processing flow" for the procedure.

CMMISDK_GetJobStepForResult: Obtains a result step of the job.

Format:

error_km CMMISDK_GetJobStepForResult(int32_km inInstrumentNo, int32_km inJobNum, int32_km inStepNum, CMMISDK_JobStepResult* outResult)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inJobNum	I	Job number (0 to 4)
inSetpNum	I	Step number (0 to 19)
outResult	O	Step content (result step)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	✓	×

Description:

This function obtains a result step of the job.

CMMISDK_SetJobImage: Sets job images.

Format:

error_km CMMISDK_SetJobImage(int32_km inInstrumentNo, int32_km inJobNum, int32_km inImageNum, const CMMISDK_JobImage* inImage)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inJobNum	I	Job number (0 to 4)
inImageNum	I	Image number (0 to 9)
inImage	I	Image

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	✓	×

Description:

This function sets job images.
To set an image, job information must be set with CMMISDK_SetJobInfo.
Refer to "3.2 Basic processing flow" for the procedure.
This function directly writes to the flash memory on the instrument.

Up to 10 images can be set for one job.
The resolutions of images that can be registered on each instrument are as follows. From the top left, register data left-aligned in the amount of the size.

	26dG	26d	25d
Resolution	W: 240 H: 128	W: 240 H: 128	W: 240 H: 128

CMMISDK_GetJobImage: Obtains job images.

Format:

error_km CMMISDK_GetJobImage(int32_km inInstrumentNo, int32_km inJobNum, int32_km inImageNum, CMMISDK_JobImage* outImage)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)
inJobNum	I	Job number (0 to 4)
inImageNum	I	Image number (0 to 9)
outImage	O	Image

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErInvalidParameter	The specified parameter is incorrect.
KmErCannotCommand	The current instrument does not support the specified function.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
×	×	✓	✓	✓	×

Description:

This function obtains job images.

CMMISDK_ResetSetting: Restores settings to the initial state.

Format:

error_km CMMISDK_ResetSetting(int32_km inInstrumentNo)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	×	✓	✓	✓	✓

Description:

This function restores settings to the initial state.
Measurement values and target data will not be deleted.

CMMISDK_ResetSettingAndData: Restores settings to the initial state and deletes all data.

Format:

error_km CMMISDK_ResetSettingAndData(int32_km inInstrumentNo)

Arguments:

Name	I/O	Explanation
inInstrumentNo	I	Instrument number (0 to 7)

Return Value:

Definition value	Explanation
KmSuccess	The processing was completed normally.
KmWarning	The processing was completed normally (there was a warning).
KmErNoConnect	No instrument is connected to the specified virtual COM port.
KmErConnectFailed	Failed to connect to the instrument. Or, connection is not established using Connect.

* Refer to the List of errors for errors not listed above.

Supported Instruments:

25cG	M6	26dG	26d	25d	23d
✓	✓	✓	✓	✓	✓

Description:

This function restores settings to the initial state and deletes all data, measurement values, and targets.

5. Definitions/Structures

5.1 Type definitions

Definition	Byte	Description	C/C++	C# and 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	
float32_km	4	Single precision floating point type	float	Single	Single
float64_km	8	Double precision floating point type	double	Double	Double

5.2 Structure definitions

CMMISDK_Port (COM port information)

Format:

```
struct CMMISDK_Port
{
    char    port_name[SIZE_PORTNAME];
}
```

Variable:

Variable	Explanation
port_name	COM port name

CMMISDK_InstrumentInfo (Instrument information)

Format:

```
struct CMMISDK_InstrumentInfo
{
    int32_km        DataSize;
    int32_km        WaveLengthStart;
    int32_km        WaveLengthEnd;
    int32_km        WaveLengthPitch;
    int32_km        SerialNo;
    int32_km        VersionMajor;
    int32_km        VersionMinor;
    int32_km        VersionFree;
    char            InstrumentName[SIZE_INSTRUMENT_NAME];
 }
```

Variable:

Variable	Explanation
DataSize	Reflectance Size of data (determined by minimum wavelength, maximum wavelength, and wavelength pitch)
WaveLengthStart	Reflectance Wavelength range: Minimum wavelength
WaveLengthEnd	Reflectance Wavelength range: Maximum wavelength
WaveLengthPitch	Reflectance Wavelength range: Wavelength pitch
SerialNo	Serial number
VersionMajor	Product version (Major)
VersionMinor	Product version (Minor)
VersionFree	Product version (Free)
InstrumentName	Product name

CMMISDK_Version (Version information)

Format:

```
struct CMMISDK_Version
{
    int32_km        major;
    int32_km        minor;
    int32_km        free;
}
```

Variable:

Variable	Explanation
major	Major
minor	Minor
free	Free

CMMISDK_Data (Measurement data)

Format:

```
struct CMMISDK_Data
{
   float64_km    data[SIZE_DATA];
}
```

Variable:

Variable	Explanation
data	Data *the DataSize number obtained by CMMISDK_GetInstrumentInfo is used.

CMMISDK_ColorCond (Color value calculation conditions)

Format:

```
struct CMMISDK_ColorCond
{
   CMMISDK_Observer     obs;
   CMMISDK_Illuminant   ill;
   CMMISDK_ColorSpace   colorSpace;
}
```

Variable:

Variable	Explanation
obs	Observer
ill	Illuminant * ILL_USER1 cannot be specified with CMMISDK_ReadLatestDataColor. * ILL_USER1 cannot be specified with CMMISDK_GetLatestDataColor.
colorSpace	Color space * A value other than COLOR_MUNSELL_C can be specified with CMMISDK_ReadLatestDataColor. * A value other than COLOR_MUNSELL_C can be specified with CMMISDK_GetLatestDataColor. * Only L*a*b*, Hunter Lab, and XYZ can be specified with CMMISDK_SetTargetDataColor and CMMISDK_GetTargetDataColor.

CMMISDK_UserCalId (User calibration ID)

Format:

```
struct CMMISDK_UserCalId
{
   char    id[SIZE_USERCAL_ID];
}
```

Variable:

Variable	Explanation
id	ID

CMMISDK_UvAdjustIndex (Index data for fluorescence adjustment)

Format:

```
struct CMMISDK_UvAdjustIndex
{
   float64_km    value;
   float64_km    tolerance;
}
```

Variable:

Variable	Explanation
value	Value
tolerance	Allowable width

CMMISDK_UvAdjustCoef (Fluorescence adjustment coefficient)

Format:

```
struct CMMISDK_UvAdjustCoef
{
    float64_km    coefficient[SIZE_DATA];
    float64_km    correction[SIZE_DATA];
    float64_km    param[SIZE_GG_PARAM];
    int32_km      date[SIZE_DATE];
}
```

Variable:

Variable	Explanation
coefficient	Fluorescence coefficient
correction	Fluorescence correction value
param	Parameter * Stored in the order of P, Q, C, m, n, k, and dWdS only when using the Ganz & Griesser method.
date	Adjustment date/time Values are stored in the array in year/month/day/hours/minutes/seconds order.

CMMISDK_UvAdjustGG (Ganz & Griesser fluorescence adjustment data)

Format:

```
struct CMMISDK_UvAdjustGG
{
    int32_km      count;
    float64_km    WI[SIZE_GG];
    float64_km    Tint[SIZE_GG];
}
```

Variable:

Variable	Explanation
count	Number of samples to be used
WI	WI value for the number of samples used
Tint	Tint value for the number of samples used

CMMISDK_GGData (Measurement data for Ganz & Griesser)

Format:

```
struct CMMISDK_GGData
{
    CMMISDK_UvAdjustData    UvFull[SIZE_GG];
    CMMISDK_UvAdjustData    UvCut[SIZE_GG];
}
```

Variable:

Variable	Explanation
UvFull	UV full reflectance
UvCut	UV cut reflectance

CMMISDK_CondSMC (SMC conditions)

Format:

```
struct CMMISDK_CondSMC
{
    CMMISDK_OnOff    enable;
    int32_km         times;
    float64_km       threshold;
```

```
}
```

Variable:

Variable	Explanation
enable	Enable or disable function
times	Number of times to average (3 to 10 times)
threshold	Threshold value level (0.01 to 9.99)

CMMISDK_UserIlluminant (User illuminant data)

Format:

```
struct CMMISDK_UserIlluminant
{
    float64_km    data[SIZE_USER_ILLUMINANT];
    char          name[SIZE_USER_ILL_NAME];
}
```

Variable:

Variable	Explanation
data	User illuminant data 85 items of data between 360 to 780 nm at 5-nm pitch
name	User illuminant name * The size is 16 bytes, but no more than 10 characters can be set. * Name displayed on the menu. “User” will be applied if the character length is zero.

CMMISDK_SavedTargetList (Saved target list)

Format:

```
struct CMMISDK_SavedTargetList
{
    int32_km    size;
    int32_km    list[SIZE_TARGET];
}
```

Variable:

Variable	Explanation
size	Number of saved targets
list	List of saved numbers * Stores the target numbers in the amount of the size variable.

CMMISDK_TargetProperty (Target properties)

Format:

```
struct CMMISDK_TargetProperty
{
    int32_km                    date[SIZE_DATE];
    int32_km                    group_list[SIZE_GROUP];
    CMMISDK_MeasType            meas_type;
    CMMISDK_MeasMode            meas_mode;
    CMMISDK_MeasArea            meas_area;
    CMMISDK_MeasAngle           meas_angle;
    CMMISDK_LightDirection      meas_ldirection;
    CMMISDK_SpecularComponent   meas_scie;
    CMMISDK_Uv                  meas_uv;
    int32_km                    warning_level;
    CMMISDK_Warning             warning;
    int32_km                    diagnosis;
    CMMISDK_DataAttr            data_attr;
    char                        name[SIZE_DATANAME];
```

}

Variable:

Variable	Explanation
date	Measurement (registration) date/time Values are stored in the array in year/month/day/hours/minutes/seconds order.
group_list	Group number list
meas_type	Measurement method
meas_mode	Measurement mode
meas_area	Measurement area
meas_angle	Measurement angle
meas_ldirection	Irradiation direction
meas_scie	Specular component
meas_uv	UV condition
warning_level	Warning level
warning	Warning information 0x01 Voltage drop 0x02 Calibration recommended 0x04 Xe lamp deterioration 0x08 LED lamp deterioration 0x10 Reflectance outside of measurable range 0x20 Gloss outside of measurable range
diagnosis	Diagnosis information 0x01 Repeatability failure 0x02 Repeatability warning 0x04 Reproducibility failure 0x08 Reproducibility warning 0x10 Intensity failure 0x20 Intensity warning
data_attr	Data attribute
name	Data name * The size is 64 bytes, but the character length depends on the instrument capabilities.

CMMISDK_ToleranceData (Tolerance data)

Format:

```
struct CMMISDK_ToleranceData
{
    int32_km   upper_enable;
    int32_km   upper_value;
    int32_km   lower_enable;
    int32_km   lower_value;
}
```

Variable:

Variable	Explanation
upper_enable	Upper limit enable/disable (0: disable, 1: enable)
upper_value	Upper limit (Value multiplied by 100. However, only x・y is a value multiplied by 10000.)
lower_enable	Lower limit enable/disable (0: disable, 1: enable)
lower_value	Lower limit (Value multiplied by 100. However, only x・y is a value multiplied by 10000.)

CMMISDK_ParametricCoef (Parametric coefficient data)

Format:

```
struct CMMISDK_ParametricCoef
{
    float64_km   coef[SIZE_PARAMETRIC_COEF];
}
```

Variable:

Variable	Explanation
coef	Parametric coefficient * The coefficients have the following order: l, c, h.

CMMISDK_SampleProperty (Measurement value properties)

Format:

```
struct CMMISDK_SampleProperty
{
    int32_km                        date[SIZE_DATE];
    CMMISDK_MeasType                meas_type;
    CMMISDK_MeasMode                meas_mode;
    CMMISDK_MeasArea                meas_area;
    CMMISDK_MeasAngle               meas_angle;
    CMMISDK_LightDirection          meas_ldirection;
    CMMISDK_SpecularComponent       meas_scie;
    CMMISDK_Uv                      meas_uv;
    CMMISDK_Warning                 warning;
    int32_km                        diagnosis;
    CMMISDK_DataAttr                data_attr;
    int32_km                        relation_target;
    char                            name[SIZE_DATANAME];
}
```

Variable:

Variable	Explanation
date	Measurement (registration) date/time Values are stored in the array in year/month/day/hours/minutes/seconds order.
meas_type	Measurement method
meas_mode	Measurement mode
meas_area	Measurement area
meas_angle	Measurement angle
meas_ldirection	Irradiation direction
meas_scie	Specular component
meas_uv	UV condition
warning	Warning information 0x01 Voltage drop 0x02 Calibration recommended 0x04 Xe lamp deterioration 0x08 LED lamp deterioration 0x10 Reflectance outside of measurable range 0x20 Gloss outside of measurable range
diagnosis	Diagnosis information 0x01 Repeatability failure 0x02 Repeatability warning 0x04 Reproducibility failure 0x08 Reproducibility warning 0x10 Intensity failure 0x20 Intensity warning
data_attr	Data attribute

relation_target	Related target number
name	Data name * The size is 64 bytes, but the character length depends on the instrument capabilities.

CMMISDK_DateTime (Date and time data)

Format:

```
struct CMMISDK_DateTime
{
    int32_km      year;
    int32_km      month;
    int32_km      day;
    int32_km      hour;
    int32_km      minute;
    int32_km      second;
}
```

Variable:

Variable	Explanation
year	Year
month	Month
day	Day
hour	Hour
minute	Minute
second	Second

CMMISDK_UserEquation (User index information)

Format:

```
struct CMMISDK_UserEquation
{
    char    formula[SIZE_USER_EQUATION];
    char    user_class[SIZE_USER_EQUATION];
 }
```

Variable:

Variable	Explanation
frmula	User index
user_cass	User classes

CMMISDK_GroupList (Group list)

Format:

```
struct CMMISDK_GroupList
{
    int32_km    group[SIZE_GROUP];
 }
```

Variable:

Variable	Explanation
group	Group list

CMMISDK_Group (Group information)

Format:

```
struct CMMISDK_Group
{
   char    name[SIZE_GROUP_NAME];
}
```

Variable:

Variable	Explanation
name	Name * The size is 32 bytes, but the character length depends on the instrument. Refer to Appendix C.

CMMISDK_GroupAll (All group information)

Format:

```
struct CMMISDK_GroupAll
{
   char    name[SIZE_GROUP_ALL][SIZE_GROUP_NAME];
}
```

Variable:

Variable	Explanation
name	Name * The size is 32 bytes, but the character length depends on the instrument. Refer to Appendix C.

CMMISDK_AdminPass (Administrator password)

Format:

```
struct CMMISDK_AdminPass
{
   char    password[SIZE_ADMIN_PASS];
}
```

Variable:

Variable	Explanation
password	Administrator password

CMMISDK_JobInfo (Job information)

Format:

```
struct CMMISDK_JobInfo
{
   int32_km          step_count;
   CMMISDK_OnOff     step_loop;
   char              name[SIZE_JOB_NAME];
}
```

Variable:

Variable	Explanation
step_count	Number of steps to register for job (1 to 20)
step_loop	Repeat job on/off
name	Name * Dependent on instrument and version.

CMMISDK_JobStepOperation (Job operation step)

Format:

```
struct CMMISDK_JobStepOperation
{
    int32_km                    image_num;
    CMMISDK_MeasType            meas_type;
    CMMISDK_MeasMode            meas_mode;
    CMMISDK_MeasArea            meas_area;
    CMMISDK_MeasAngle           meas_angle;
    CMMISDK_LightDirection      meas_ldirection;
    CMMISDK_SpecularComponent   meas_scie;
    CMMISDK_Uv                  meas_uv;
    int32_km                    auto_ave_times;
    int32_km                    manu_ave_times;
    int32_km                    relation_target;
    CMMISDK_OnOff               enable_meas;
    CMMISDK_OnOff               enable_prev;
    CMMISDK_OnOff               enable_next;
    CMMISDK_OnOff               enable_end;
    char                        name[SIZE_DATANAME];
    char                        comment[SIZE_JOB_COMMENT];
}
```

Variable:

Variable	Explanation
image_num	Image number (0 to 10) * 10 is handled as “No image”.
meas_type	Measurement type
meas_mode	Measurement mode
meas_area	Measurement area
meas_angle	Measurement angle
meas_ldirection	Irradiation direction
meas_scie	Specular component
meas_uv	UV condition
auto_ave_times	Auto average count (1 to 10)
manu_ave_times	Manual average count (1 to 30)
relation_target	Related target number * Opacity attribute data cannot be set.
enable_meas	Measurement button enable/disable
eable_prev	Display previous button on/off
enable_next	Display next button on/off
enable_end	End button enable/disable
name	Data name * Dependent on instrument and version.
comment	Comment * Dependent on instrument and version.

CMMISDK_JobStepResult (Job result step)

Format:

```
struct CMMISDK_JobStepResult
{
    CMMISDK_SpecularComponent   meas_scie;
    CMMISDK_Observer            obs1;
    CMMISDK_Observer            obs2;
    CMMISDK_Illuminant          ill1;
    CMMISDK_Illuminant          ill2;
    CMMISDK_CustomIndex         index[SIZE_JOB_INDEX];
    CMMISDK_OnOff               enable_meas;
```

```
    CMMISDK_OnOff       enable_prev;
    CMMISDK_OnOff       enable_next;
    CMMISDK_OnOff       enable_end;
  }
```

Variable:

Variable	Explanation
meas_scie	Specular component
obs1	Observer 1
obs2	Observer 2
ill1	Illuminant 1
ill2	Illuminant 2
index	Custom items
enable_meas	Measurement button enable/disable
eable_prev	Display previous button on/off
enable_next	Display next button on/off
enable_end	End button enable/disable

CMMISDK_JobImage (Job image)

Format:

```
struct CMMISDK_JobImage
{
    int32_km    width;
    int32_km    height;
    int32_km    data[SIZE_IMAGEDATA];
    char        name[SIZE_JOBIMAGE_NAME];
}
```

Variable:

Variable	Explanation
width	Width (240 fixed)
height	Height (128 fixed)
data	Image data (arranged in Z order from top left) The data is stored as 1 pixel per element. R, G, and B are each 8 bits. The data is stored right-aligned in BGR order.
name	Image name

5.3 Value definition

CMMISDK_Warning (Warning status)

	Value	Explanation
KmWrBattery	0x01	Low battery voltage.
KmWrCalibration	0x02	Recalibration required. It has been a long time since calibration.
KmWrPreAnnualCalibraton	0x04	Periodical calibration required soon.
KmWrAnnualCalibraton	0x08	Periodical calibration required.
KmWrLampForColor	0x10	Low illuminant intensity for color measurement.
KmWrOutOfColorRange	0x20	Reflectance outside range of guaranteed performance.
KmWrOutOfGlossRange	0x40	Gloss outside range of guaranteed performance.
KmWrLampForGloss	0x80	Low illuminant intensity for gloss measurement.

* Dependent on instrument and version.

CMMISDK_CalStatus (Calibration status)

	Value	Explanation
StatusZero	0	Zero calibration is required
StatusWhite	1	White calibration is required
StatusGloss	2	Gloss calibration is required
StatusMeasure	3	Measurement is possible
StatusMeasureWrn	4	Measurement is possible (calibration recommended)
StatusUser	5	User calibration is required

* Dependent on instrument and version.

CMMISDK_CalDataType (Calibration data type)

■ For the CM-25cG

	Value	Explanation
CALTYPE_MAV	0	MAV
CALTYPE_SAV	1	SAV

■ For the CM-26dG/CM-26d

	Value	Explanation
CALTYPE_MAV_SCI	0	MAV/SCI
CALTYPE_MAV_SCE	1	MAV/SCE
CALTYPE_SAV_SCI	2	SAV/SCI
CALTYPE_SAV_SCE	3	SAV/SCE

■ For CM-25d/CM-23d

	Value	Explanation
CALTYPE_MAV_SCI	0	MAV/SCI
CALTYPE_MAV_SCE	1	MAV/SCE

■ For the CM-M6

	Value	Explanation
CALTYPE_L_ANGLE_M15	0	Left -15°
CALTYPE_L_ANGLE_15	1	Left 15°
CALTYPE_L_ANGLE_25	2	Left 25°
CALTYPE_L_ANGLE_45	3	Left 45°
CALTYPE_L_ANGLE_75	4	Left 75°
CALTYPE_L_ANGLE_110	5	Left 110°
CALTYPE_R_ANGLE_M15	6	Right -15°

CALTYPE_R_ANGLE_15	7	Right 15°
CALTYPE_R_ANGLE_25	8	Right 25°
CALTYPE_R_ANGLE_45	9	Right 45°
CALTYPE_R_ANGLE_75	10	Right 75°
CALTYPE_R_ANGLE_110	11	Right 110°

CMMISDK_MeasStatus (Measurement status)

	Value	Explanation
Idling	0	Idling state
Measuring	1	Measuring

CMMISDK_DataType (Data type)

■ For the CM-25cG

	Value	Explanation
DATATYPE_GLOSS	0	GU (1 item)
DATATYPE_SPEC	1	Spectral data

■ For the CM-26dG

	Value	Explanation
DATATYPE_GLOSS	0	GU (1 item)
DATATYPE_SCI	1	SCI (UV condition: Only when number of outputs is 1)
DATATYPE_SCE	2	SCE (UV condition: Only when number of outputs is 1)
DATATYPE_BACKWHITE	3	Opacity white back
DATATYPE_BACKBLACK	4	Opacity black back
DATATYPE_SCI_UVFULL	10	SCI (UV100%)
DATATYPE_SCE_UVFULL	11	SCE (UV100%)
DATATYPE_SCI_UVCUT	12	SCI (UV cut)
DATATYPE_SCE_UVCUT	13	SCE (UV cut)
DATATYPE_SCI_UVADJ	14	SCI (UV adjustment)
DATATYPE_SCE_UVADJ	15	SCE (UV adjustment)

■ For the CM-26d

	Value	Explanation
DATATYPE_SCI	1	SCI (UV condition: Only when number of outputs is 1)
DATATYPE_SCE	2	SCE (UV condition: Only when number of outputs is 1)
DATATYPE_BACKWHITE	3	Opacity white back
DATATYPE_BACKBLACK	4	Opacity black back
DATATYPE_SCI_UVFULL	10	SCI (UV100%)
DATATYPE_SCE_UVFULL	11	SCE (UV100%)
DATATYPE_SCI_UVCUT	12	SCI (UV cut)
DATATYPE_SCE_UVCUT	13	SCE (UV cut)
DATATYPE_SCI_UVADJ	14	SCI (UV adjustment)
DATATYPE_SCE_UVADJ	15	SCE (UV adjustment)

■ For the CM-25d/CM-23d

	Value	Explanation
DATATYPE_SCI	1	SCI
DATATYPE_SCE	2	SCE
DATATYPE_BACKWHITE	3	Opacity white back
DATATYPE_BACKBLACK	4	Opacity black back

■ For the CM-M6

	Value	Explanation
DATATYPE_L_ANGLE_M15	0	Left -15°
DATATYPE_L_ANGLE_15	1	Left 15°
DATATYPE_L_ANGLE_25	2	Left 25°

DATATYPE_L_ANGLE_45	3	Left 45°
DATATYPE_L_ANGLE_75	4	Left 75°
DATATYPE_L_ANGLE_110	5	Left 110°
DATATYPE_R_ANGLE_M15	6	Right -15°
DATATYPE_R_ANGLE_15	7	Right 15°
DATATYPE_R_ANGLE_25	8	Right 25°
DATATYPE_R_ANGLE_45	9	Right 45°
DATATYPE_R_ANGLE_75	10	Right 75°
DATATYPE_R_ANGLE_110	11	Right 110°
DATATYPE_DP_ANGLE_M15	12	Double path -15°
DATATYPE_DP_ANGLE_15	13	Double path 15°
DATATYPE_DP_ANGLE_25	14	Double path 25°
DATATYPE_DP_ANGLE_45	15	Double path 45°
DATATYPE_DP_ANGLE_75	16	Double path 75°
DATATYPE_DP_ANGLE_110	17	Double path 110°

* If no number is indicated, the DataSize number obtained by CMMISDK_GetInstrumentInfo is used.

CMMISDK_CondUvAdjust (Fluorescence adjustment conditions)

	Value	Explanation
UVADJ_PROFILE	0	Profile
UVADJ_WI	1	WI
UVADJ_TINT	2	Tint
UVADJ_WITINT	3	WI & Tint
UVADJ_BRIGHTNESS	4	ISO brightness
UVADJ_GG	5	Ganz & Griesser
UVADJ_NONE	-1	No condition

* Dependent on instrument and version.

CMMISDK_UvAdjustDataType (Fluorescence coefficient data type)

	Value	Explanation
UVADJ_DATATYPE_SCI	0	SCI
UVADJ_DATATYPE_SCE	1	SCE
UVADJ_DATATYPE_NONE	-1	No condition

* Dependent on instrument and version.

CMMISDK_MeasType (Measurement type)

	Value	Explanation
MEASTYPE_REF	0	Reflected
MEASTYPE_TRA	1	Transmitted
MEASTYPE_NONE	-1	No condition

* Dependent on instrument and version.

CMMISDK_MeasArea (Measurement area)

	Value	Explanation
AREA_MAV	0	MAV
AREA_SAV	1	SAV
AREA_LAV	2	LAV
AREA_LMAV	3	LMAV
AREA_NONE	-1	Area fixed

* Dependent on instrument and version.

CMMISDK_MeasAngle (Measurement angle)

	Value	Explanation
MEAS_ANGLE_M15	0x01	-15°
MEAS_ANGLE_15	0x02	15°
MEAS_ANGLE_25	0x04	25°
MEAS_ANGLE_45	0x08	45°
MEAS_ANGLE_75	0x10	75°
MEAS_ANGLE_110	0x20	110°
MEAS_ANGLE_NONE	-1	No condition

* Dependent on instrument and version.

CMMISDK_MeasMode (Measurement mode)

	Value	Explanation
MEASMODE_COLORANDGLOSS	0	Obtain measured color and gloss
MEASMODE_COLORONLY	1	Obtain measured color only
MEASMODE_GLOSSONLY	2	Obtain gloss only
MEASMODE_OPACITY	3	Opacity
MEASMODE_NONE	-1	No condition

* Dependent on instrument and version.

CMMISDK_SpecularComponent (Specular component)

	Value	Explanation
SC_SCI	0	SCI
SC_SCE	1	SCE
SC_SCIE	2	SCI+SCE
SC_NONE	-1	No condition

* Dependent on instrument and version.

CMMISDK_Uv (UV condition)

	Value	Explanation
UV_100	0	UV100%
UV_CUT400	1	UV Cut 400 nm
UV_CUT420	2	UV Cut 420 nm
UV_CUT400N	3	UV adjustment Cut 400 nm, normal light emission
UV_CUT400L	4	UV adjustment Cut 400 nm, low light emission
UV_CUT420N	5	UV adjustment Cut 420 nm, normal light emission
UV_CUT420L	6	UV adjustment Cut 420 nm, low light emission
UV_100_CUT400	7	UV 100% + UV cut 400 nm
UV_100_CUT420	8	UV 100% + UV cut 420 nm
UV_100_CUT400N	9	UV 100% + UV cut 400 nm + UV adjustment cut 400 nm, normal light emission
UV_100_CUT400L	10	UV 100% + UV cut 400 nm + UV adjustment cut 400 nm, low light emission
UV_100_CUT420N	11	UV 100% + UV cut 420 nm + UV adjustment cut 420 nm, normal light emission
UV_100_CUT420L	12	UV 100% + UV cut 420 nm + UV adjustment cut 420 nm, low light emission
UV_NONE	-1	No condition

* Dependent on instrument and version.

CMMISDK_SaveMode (Save method)

	Value	Explanation
SAVEMODE_AUTO	0	Auto save
SAVEMODE_MANUAL	1	Manual save

CMMISDK_DisplayType (Display type)

	Value	Explanation
DISPTYPE_ABS	0x001	Absolute values
DISPTYPE_DIF	0x002	Color difference
DISPTYPE_ABSDIF	0x004	Absolute value & color difference
DISPTYPE_CUSTOM	0x008	Custom
DISPTYPE_GRAPH_ABS	0x010	Absolute value graph
DISPTYPE_GRAPH_DIF	0x020	Color difference graph
DISPTYPE_GRAPH_REF	0x040	Spectral graph
DISPTYPE_PASS_FAIL	0x080	Judgment
DISPTYPE_MI	0x100	MI
DISPTYPE_GRAPH_LINE	0x200	Line graph
DISPTYPE_AUDI2000_EC	0x400	ΔEc(Audi2000)
DISPTYPE_AUDI2000_EP	0x800	Δep(Audi2000)

* Dependent on instrument and version.

CMMISDK_Observer (Observer)

	Value	Explanation
OBS_02	0	2°
OBS_10	1	10°

CMMISDK_Illuminant (Illuminant)

	Value	Explanation
ILL_NONE	0	None
ILL_A	1	A
ILL_C	2	C
ILL_D50	3	D50
ILL_D65	4	D65
ILL_ID50	5	ID50
ILL_ID65	6	ID65
ILL_F2	7	F2
ILL_F6	8	F6
ILL_F7	9	F7
ILL_F8	10	F8
ILL_F10	11	F10
ILL_F11	12	F11
ILL_F12	13	F12
ILL_USER1	14	User 1

* Dependent on instrument and version.

CMMISDK_ColorSpace (Color space)

	Value	Explanation
COLOR_LAB	0	L*a*b*
COLOR_LCH	1	L*C*h

COLOR_HLAB	2	Hunter Laboratory
COLOR_YXY	3	Yxy
COLOR_XYZ	4	XYZ
COLOR_MUNSELL_C	5	Munsell (C)

* Dependent on instrument and version.

CMMISDK_Equation (Color equation)

	Value	Explanation
EQUATION_DE1976	0	ΔE*ab
EQUATION_CMC	1	CMC
EQUATION_DE1994	2	ΔE*94
EQUATION_DE2000	3	ΔE00
EQUATION_DEH	4	ΔE(Hunter)
EQUATION_DEP	5	Δep(DIN6175)
EQUATION_DEC	6	Δec(DIN6175)
EQUATION_DE99o	7	ΔE99(DN99o)

* Dependent on instrument and version.

CMMISDK_CustomIndex (Custom item)

	Value	Explanation
CUSTOM_NONE	0	None
CUSTOM_L	1	L*
CUSTOM_A	2	a*
CUSTOM_B	3	b*
CUSTOM_C	4	C*
CUSTOM_H	5	h
CUSTOM_HL	6	L(Hunter)
CUSTOM_HA	7	a(Hunter)
CUSTOM_HB	8	b(Hunter)
CUSTOM_X	9	X
CUSTOM_Y	10	Y
CUSTOM_Z	11	Z
CUSTOM_SX	12	x
CUSTOM_SY	13	y
CUSTOM_MH	14	H
CUSTOM_MV	15	V
CUSTOM_MC	16	C
CUSTOM_WI_E	17	WI(E313-73)
CUSTOM_WI_C	18	WI(CIE)
CUSTOM_TINT_C	19	Tint(CIE)
CUSTOM_YI_E	20	YI(E313)
CUSTOM_YI_D	21	YI(D1925)
CUSTOM_B_ISO	22	B(ISO)
CUSTOM_GU	23	GU
CUSTOM_USER_E1	24	UserE1
CUSTOM_USER_C1	25	UserC1
CUSTOM_USER_E2	26	UserE2
CUSTOM_USER_C2	27	UserC2
CUSTOM_USER_E3	28	UserE3
CUSTOM_USER_C3	29	UserC3
CUSTOM_GLOSS8	30	8° gloss
CUSTOM_WI_G	31	WI(Ganz)
CUSTOM_TINT_G	32	Tint(Ganz)

CUSTOM_DL	-1	ΔL*
CUSTOM_DA	-2	Δa*
CUSTOM_DB	-3	Δb*
CUSTOM_DC	-4	ΔC*
CUSTOM_DH	-5	ΔH*
CUSTOM_DHL	-6	ΔL(Hunter)
CUSTOM_DHA	-7	Δa(Hunter)
CUSTOM_DHB	-8	Δb(Hunter)
CUSTOM_DX	-9	ΔX
CUSTOM_DY	-10	ΔY
CUSTOM_DZ	-11	ΔZ
CUSTOM_DSX	-12	Δx
CUSTOM_DSY	-13	Δy
CUSTOM_DWI_E	-14	ΔWI(E313-73)
CUSTOM_DWI_C	-15	ΔWI(CIE)
CUSTOM_DTINT_C	-16	Δtint(CIE)
CUSTOM_DYI_E	-17	ΔYI(E313)
CUSTOM_DYI_D	-18	ΔYI(D1925)
CUSTOM_DB_ISO	-19	ΔB(ISO)
CUSTOM_DGU	-20	ΔGU
CUSTOM_MI	-21	MI
CUSTOM_DE	-22	ΔE*ab
CUSTOM_CMC	-23	CMC
CUSTOM_DE94	-24	ΔE*94
CUSTOM_DE00	-25	ΔE00
CUSTOM_DEH	-26	ΔE(Hunter)
CUSTOM_DE99O	-27	ΔE99o
CUSTOM_STRENGTH_XYZ	-28	StrengthXYZ
CUSTOM_STRENGTH_X	-29	StrengthX
CUSTOM_STREMGTH_Y	-30	StrengthY
CUSTOM_STRENGTH_Z	-31	StrengthZ
CUSTOM_GREYSCALE	-32	GreyScale
CUSTOM_DWI_G	-33	ΔWI(Ganz)
CUSTOM_DTINT_G	-34	ΔTint(Ganz)

* Dependent on instrument and version.

CMMISDK_Direction (Irradiation direction to display)

	Value	Explanation
DIRECTION_DP	0	Double path
DIRECTION_L	1	L direction

* Dependent on instrument and version.

CMMISDK_LightDirection (Irradiation direction)

	Value	Explanation
LDIRECTION_NONE	0	None
LDIRECTION_L	0x01	L direction
LDIRECTION_R	0x02	R direction
LDIRECTION_DP	0x04	Double path

* Dependent on instrument and version.

CMMISDK_DataAttr (Data attribute)

	Value	Explanation

DATAATTR_SPEC	0	Spectral data
DATAATTR_LAB	1	L*a*b*
DATAATTR_HLAB	2	Hunter Laboratory
DATAATTR_XYZ	3	XYZ

CMMISDK_FilterIndex (Filter attribute)

	Value	Explanation
FILTER_OFF	0	OFF
FILTER_SAVE	1	Displays only the saved data
FILTER_GROUP	2	Displays only the data corresponding to the specified group number

CMMISDK_InstrumentMode (Instrument mode)

	Value	Explanation
INSTRUMENTMODE_NORMAL	0	Normal mode
INSTRUMENTMODE_SIMPLE	1	Simple mode

* Dependent on instrument and version.

CMMISDK_UserType (User type)

	Value	Explanation
USERTYPE_ADMIN	0	Administrator
USERTYPE_WORKER	1	Operator

* Dependent on instrument and version.

CMMISDK_ScreenDirection (Display direction of screen)

	Value	Explanation
SCREENDIR_0	0	Not rotated
SCREENDIR_180	1	Rotated 180°

* Dependent on instrument and version.

CMMISDK_DateFormat (Date format)

	Value	Explanation
DF_YYYYMMDD	0	YYYY/MM/DD
DF_MMDDYYYY	1	MM/DD/YYYY
DF_DDMMYYYY	2	DD/MM/YYYY

CMMISDK_Language (Language)

	Value	Explanation
LANGUAGE_ENGLISH	0	English
LANGUAGE_JAPANESE	1	Japanese
LANGUAGE_GERMAN	2	German
LANGUAGE_FRENCH	3	French
LANGUAGE_SPANISH	4	Spanish
LANGUAGE_ITALIAN	5	Italian
LANGUAGE_CHINESE_S	6	Chinese (simplified)
LANGUAGE_PORTUGUESE	7	Portuguese
LANGUAGE_RUSSIAN	8	Russian
LANGUAGE_POLISH	9	Polish

LANGUAGE_TURKISH	10	Turkish

CMMISDK_JobStepType (Job step type)

	Value	Explanation
JOB_STEPTYPE_OPERATION	0	Operation step
JOB_STEPTYPE_RESULT	1	Result step

CMMISDK_OnOff (ON/OFF)

	Value	Explanation
OFF	0	OFF (disabled)
ON	1	ON (enabled)

CMMISDK_ToleranceType (Tolerance type)

■ For the CM-25cG

	Value	Explanation
TOLETYPE_SPEC	0	

■ For the CM-26dG/CM-26d/CM-25d/CM-23d

	Value	Explanation
TOLETYPE_SCI	0	SCI
TOLETYPE_SCE	1	SCE

■ For the CM-M6

	Value	Explanation
TOLETYPE_L_ANGLE_M15	0	Left -15°
TOLETYPE_L_ANGLE_15	1	Left 15°
TOLETYPE_L_ANGLE_25	2	Left 25°
TOLETYPE_L_ANGLE_45	3	Left 45°
TOLETYPE_L_ANGLE_75	4	Left 75°
TOLETYPE_L_ANGLE_110	5	Left 110°
TOLETYPE_R_ANGLE_M15	6	Right -15°
TOLETYPE_R_ANGLE_15	7	Right 15°
TOLETYPE_R_ANGLE_25	8	Right 25°
TOLETYPE_R_ANGLE_45	9	Right 45°
TOLETYPE_R_ANGLE_75	10	Right 75°
TOLETYPE_R_ANGLE_110	11	Right 110°
TOLETYPE_DP_ANGLE_M15	12	Double path -15°
TOLETYPE_DP_ANGLE_15	13	Double path 15°
TOLETYPE_DP_ANGLE_25	14	Double path 25°
TOLETYPE_DP_ANGLE_45	15	Double path 45°
TOLETYPE_DP_ANGLE_75	16	Double path 75°
TOLETYPE_DP_ANGLE_110	17	Double path 110°

CMMISDK_ToleranceId (Tolerance ID)

	Value	Explanation
TOLERANCE_ID_L	-1	ΔL*
TOLERANCE_ID_A	-2	Δa*
TOLERANCE_ID_B	-3	Δb*
TOLERANCE_ID_C	-4	ΔC*
TOLERANCE_ID_H	-5	ΔH*
TOLERANCE_ID_HL	-6	ΔL(Hunter)
TOLERANCE_ID_HA	-7	Δa(Hunter)

TOLERANCE_ID_HB	-8	Δb(Hunter)
TOLERANCE_ID_X	-9	ΔX
TOLERANCE_ID_Y	-10	ΔY
TOLERANCE_ID_Z	-11	ΔZ
TOLERANCE_ID_SX	-12	Δx
TOLERANCE_ID_SY	-13	Δy
TOLERANCE_ID_WI_E	-14	ΔWI(E313-73)
TOLERANCE_ID_WI_C	-15	ΔWI(CIE)
TOLERANCE_ID_TINT_C	-16	Δtint(CIE)
TOLERANCE_ID_YI_E	-17	ΔYI(E313)
TOLERANCE_ID_YI_D	-18	ΔYI(D1925)
TOLERANCE_ID_B_ISO	-19	ΔB(ISO)
TOLERANCE_ID_GU	-20	ΔGU
TOLERANCE_ID_MI	-21	MI
TOLERANCE_ID_DE	-22	ΔE*ab
TOLERANCE_ID_CMC	-23	CMC
TOLERANCE_ID_DE94	-24	ΔE*94
TOLERANCE_ID_DE00	-25	ΔE00
TOLERANCE_ID_DEH	-26	ΔE(Hunter)
TOLERANCE_ID_DEP_DIN6175	-27	Δep(DIN6175)
TOLERANCE_ID_DEC_DIN6175	-28	Δec(DIN6175)
TOLERANCE_ID_FF	-29	ΔFF
TOLERANCE_ID_DE99O	-30	ΔE99o
TOLERANCE_ID_DEC_AUDI2000	-31	Δec(Audi2000)
TOLERANCE_ID_MDEC_AUDI2000	-32	Δec average (Audi2000)
TOLERANCE_ID_DECM_AUDI2000	-33	Δec maximum (Audi2000)
TOLERANCE_ID_DEP_AUDI2000	-34	Δep(Audi2000)
TOLERANCE_ID_MDEP_AUDI2000	-35	Δep average (Audi2000)
TOLERANCE_ID_DEPM_AUDI2000	-36	Δep maximum (Audi2000)
TOLERANCE_ID_DSTRENGTH_XYZ	-37	ΔStrengthXYZ
TOLERANCE_ID_DSTRENGTH_X	-38	ΔstrengthX
TOLERANCE_ID_DSTRENGTH_Y	-39	ΔstrengthY
TOLERANCE_ID_DSTRENGTH_Z	-40	ΔstrengthZ
TOLERANCE_ID_DOPACITY	-41	Opacity difference
TOLERANCE_ID_DGRAYSCALE	-42	Grayscale difference
TOLERANCE_ID_WI_G	-43	ΔWI(Ganz)
TOLERANCE_ID_TINT_G	-44	ΔTint(Ganz)

* Dependent on instrument and version.

CMMISDK_ParametricId (Parametric coefficient ID)

	Value	Explanation
PARAMETRIC_ID_CMC	0	CMC
PARAMETRIC_ID_DE94	1	ΔE*94
PARAMETRIC_ID_DE00	2	ΔE00

CMMISDK_DateType (Date/time type)

	Value	Explanation
DATETYPE_COLOR	0	Color
DATETYPE_GLOSS	1	Gloss

* Dependent on instrument and version.

Size definitions

SIZE_PORTNAME	32	Size of port number name
SIZE_INSTRUMENT_NAME	32	Size of instrument name
SIZE_DATA	39	Amount of data
SIZE_USERCAL_ID	16	Size of user ID
SIZE_GG	5	Ganz & Griesser size
SIZE_GG_PARAM	7	Ganz & Griesser parameter size
SIZE_USER_ILLUMINANT	85	User illuminant data
SIZE_USER_ILL_NAME	16	User illuminant name
SIZE_TARGET	2500	Amount of target data
SIZE_DATE	6	Size of date/time
SIZE_GROUP	5	Size of group list
SIZE_GROUP_ALL	50	Total size of groups
SIZE_DATANAME	64	Size of data name
SIZE_PARAMETRIC_COEF	3	Size of parametric coefficient
SIZE_USER_EQUATION	200	Size of user index syntax
SIZE_GROUP_NAME	32	Size of group name
SIZE_ADMIN_PASS	8	Size of administrator password
SIZE_JOB_NAME	32	Size of job name
SIZE_JOB_COMMENT	128	Size of job comment
SIZE_JOB_INDEX	7	Size of job custom item
SIZE_JOBIMAGE_NAME	32	Size of job image name
SIZE_IMAGEDATA	153600	Image size (assuming 320 × 480 max.)

6. Errors/Warnings

6.1 List of errors

Error ID	Value		
KmSuccess	0	Description	The processing was completed normally.
		Action	
KmWarning	1	Description	The processing was completed normally (there was a warning).
		Action	Use CMMISDK_GetWarning to check the error status and then take action.
KmErNoConnect	10	Description	No instrument is connected to the specified virtual COM port.
		Action	Check the following: • Is the instrument powered on? • Is the cable correctly connected? • Is the COM port number correct? • No other software is controlling the instrument?
KmErInvalidParameter	25	Description	The specified parameter is incorrect.
		Action	• Check the input range and enter an appropriate value.
KmErCannotCommand	30	Description	The current instrument does not support the specified function.
		Action	
KmErNoData	45	Description	No data
		Action	• The necessary data must be registered beforehand.
KmErDataProtected	46	Description	The data is protected.
		Action	• Cancel data protection before performing the operation.
KmErOutOfRangeValue	50	Description	The value is outside the range that can be measured by the instrument.
		Action	The sample cannot be measured.
KmErConnectFailed	100	Description	Failed to connect to the instrument. Or, connection is not established using Connect.
		Action	• Perform operation after establishing the connection using Connect.
KmErDevice	110	Description	A device in the instrument has malfunctioned.
		Action	* If this problem persists even after the instrument is restarted, contact Customer Service.
KmErAd	111	Description	The A/D converter in the instrument has malfunctioned.
		Action	* If this problem persists even after the instrument is restarted, contact Customer Service.
KmErCharge	112	Description	Charging of the light emission circuit in the instrument has malfunctioned.
		Action	* If this problem persists even after the instrument is restarted, contact Customer Service.
KmErFlash	113	Description	The light emission circuit in the instrument has malfunctioned.
		Action	* If this problem persists even after the instrument is restarted, contact Customer Service.

KmErFinder	114	Description	Operation is not possible because the finder is open.
		Action	• Close the finder before performing the operation. * If this error occurs even when the finder is closed, contact Customer Service.
KmErCalculation	115	Description	The calculation cannot be performed because the required information is lacking.
		Action	* If this problem persists even after the instrument is restarted, contact Customer Service.
KmErCalibration	120	Description	Calibration was not executed in the correct procedure.
		Action	• Is the zero calibration box used for zero calibration? • Is the white calibration plate used for white calibration? • Is the gloss calibration plate used for gloss calibration?
KmErCalibrationRequired	130	Description	Necessary calibration was not executed beforehand.
		Action	• Zero calibration must be completed before performing white calibration. • White calibration must be completed before performing gloss calibration. • Gloss calibration must be completed before performing measurements.
KmErNoCalibrationData	131	Description	Calibration data is not set.
		Action	• Set calibration data.
KmErTiltDetection	140	Description	The instrument is tilted.
		Action	• Install the instrument correctly for the sample.
KmErNotUse	170	Description	This setting cannot be used due to its combination with another setting.
		Action	• Change the other setting to solve this problem.
KmErDisagreeCond	171	Description	This cannot be set because the conditions do not agree.
		Action	• Data of the same mode must be associated. For example, opacity data cannot be associated with color+gloss data.
KmErUvAdjust	172	Description	The measurement sample does not contain fluorescence.
		Action	Be sure to measure an appropriate sample.
KmErBattery	180	Description	Low battery voltage.
		Action	• Charge the battery or connect the instrument to a power supply. * If this problem persists even after charging, contact Customer Service.
KmErMemory	181	Description	Reading or writing the memory in the instrument has failed.
		Action	* If this problem persists even after the instrument is restarted, contact Customer Service.
KmErMotor	182	Description	The motor in the instrument has malfunctioned.
		Action	* If this problem persists even after the instrument is restarted, contact Customer

			Service.
KmErNotSupported	190	Description	The instrument supports the function, but the function cannot be used.
		Action	• The format being used is not supported. Use another format of the function. • The SDK version is old and the data cannot be used. Upgrade the SDK to a more recent version.
KmErCalculateColor	195	Description	A color value cannot be calculated from reflectance.
		Action	Color cannot be calculated from the reflectance. Make sure the material being measured is appropriate for measurement.
KmErCalculateCoef	196	Description	The fluorescence coefficient cannot be calculated.
		Action	Check the tolerance.
KmErUuid	198	Description	The data cannot be registered because the uuid is already in use.
		Action	• Change the uuid and register the data again.
KmEr	200	Description	An unexpected error has occurred.
		Action	Check again after restarting the instrument.

6.2 List of warnings

Warning ID	Value	
KmWrBattery	0x01	Low battery voltage.
KmWrCalibration	0x02	Recalibration recommended. It has been a long time since calibration.
KmWrPreAnnualCalibraton	0x04	Periodical calibration required soon.
KmWrAnnualCalibraton	0x08	Periodical calibration required. Perform periodical calibration.
KmWrLampForColor	0x10	Low light intensity in illuminant for color measurements.
KmWrOutOfColorRange	0x20	Reflectance outside range of guaranteed performance.
KmWrOutOfGlossRange	0x40	Gloss outside range of guaranteed performance.
KmWrLampForGloss	0x80	Low light intensity in illuminant for gloss measurements.

Appendix A. Available character codes

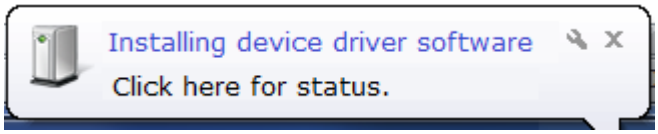
The character codes that can be used for names and comments are listed below.

	00	10	20	30	40	50	60	70
0			(sp)	0	@	P	`	p
1			!	1	A	Q	a	q
2			"	2	B	R	b	r
3			#	3	C	S	c	s
4			$	4	D	T	d	t
5			%	5	E	U	e	u
6			&	6	F	V	f	v
7			’	7	G	W	g	w
8			(	8	H	X	h	x
9			)	9	I	Y	i	y
A			*	:	J	Z	j	z
B			+	;	K	[	k	{
C				<	L	¥	l	|
D			-	=	M	]	m	}
E			.	>	N	^	n	~
F			/	?	O	_	o	

Appendix B. Installing the device driver

The device driver for the instrument must be installed in advance to connect the instrument to the PC via USB.

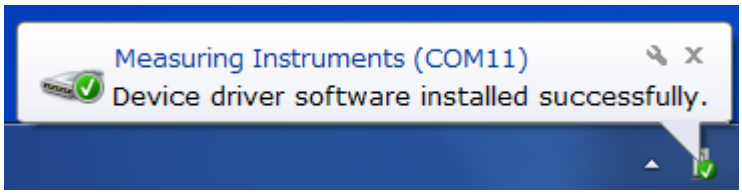
First connect the instrument to the PC, and then turn on the power supply to the instrument.
The driver installation will begin automatically. For Windows 7, the "Installing device driver software" popup will be displayed on the taskbar at the bottom right of the screen.



The device driver that is installed may not function correctly due to Windows 10. For this reason, manually install the "KMMIUSB.INF" device driver according to the following installation procedure.

Automatic installation

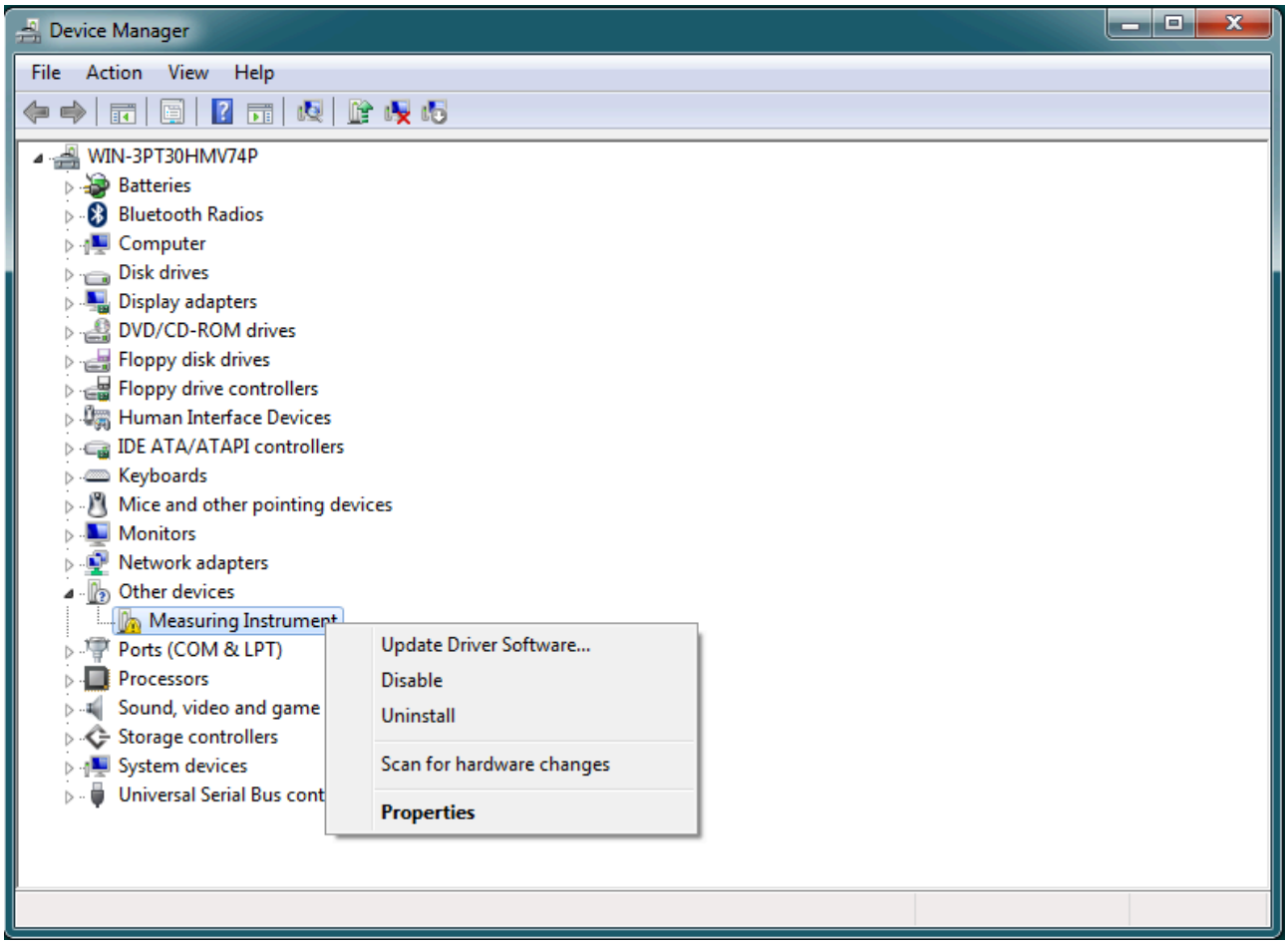
The installation is finished if the automatic installation was successful.



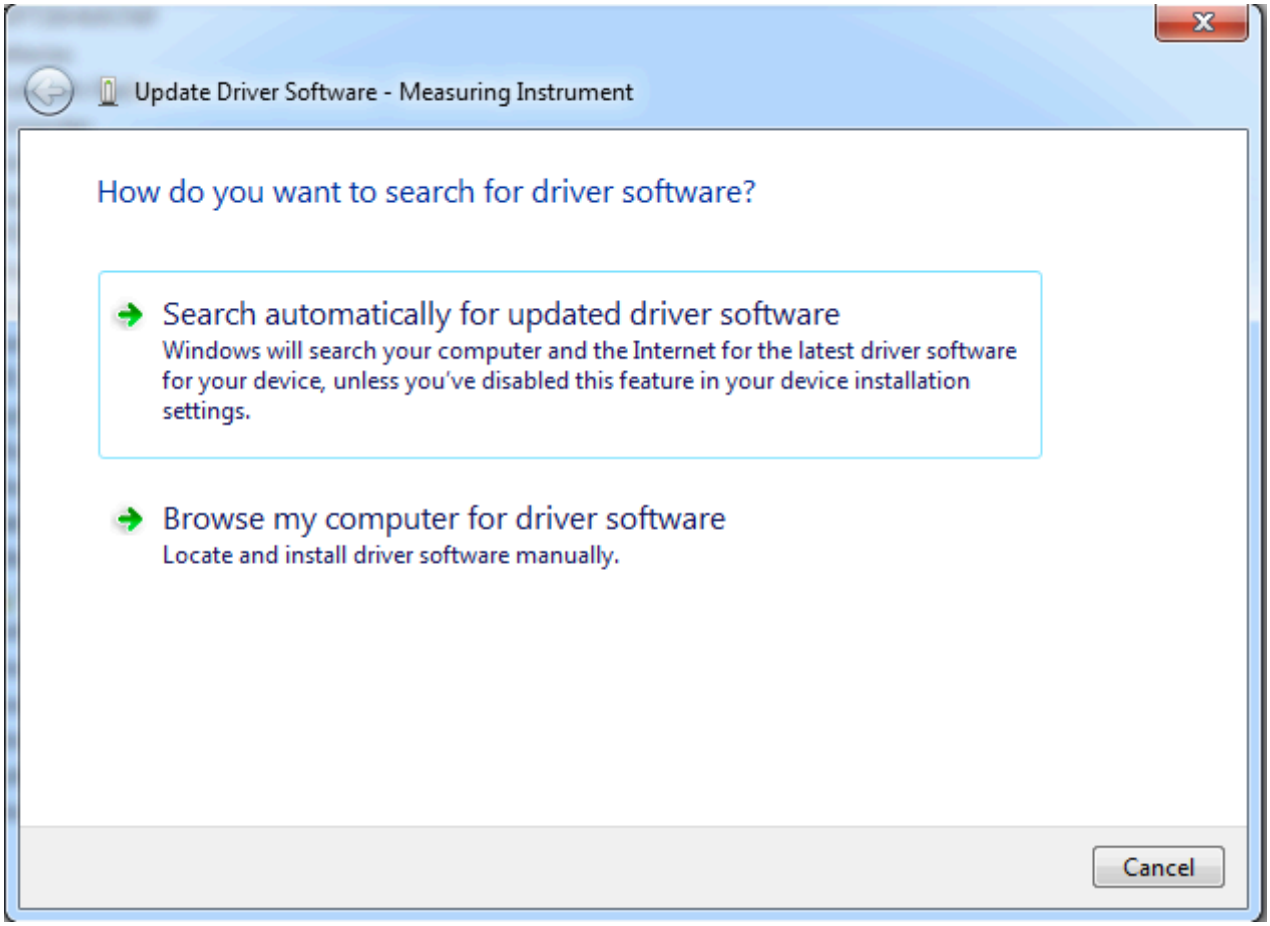
Manual installation

If the automatic installation has failed, use the following procedure to perform manual installation.

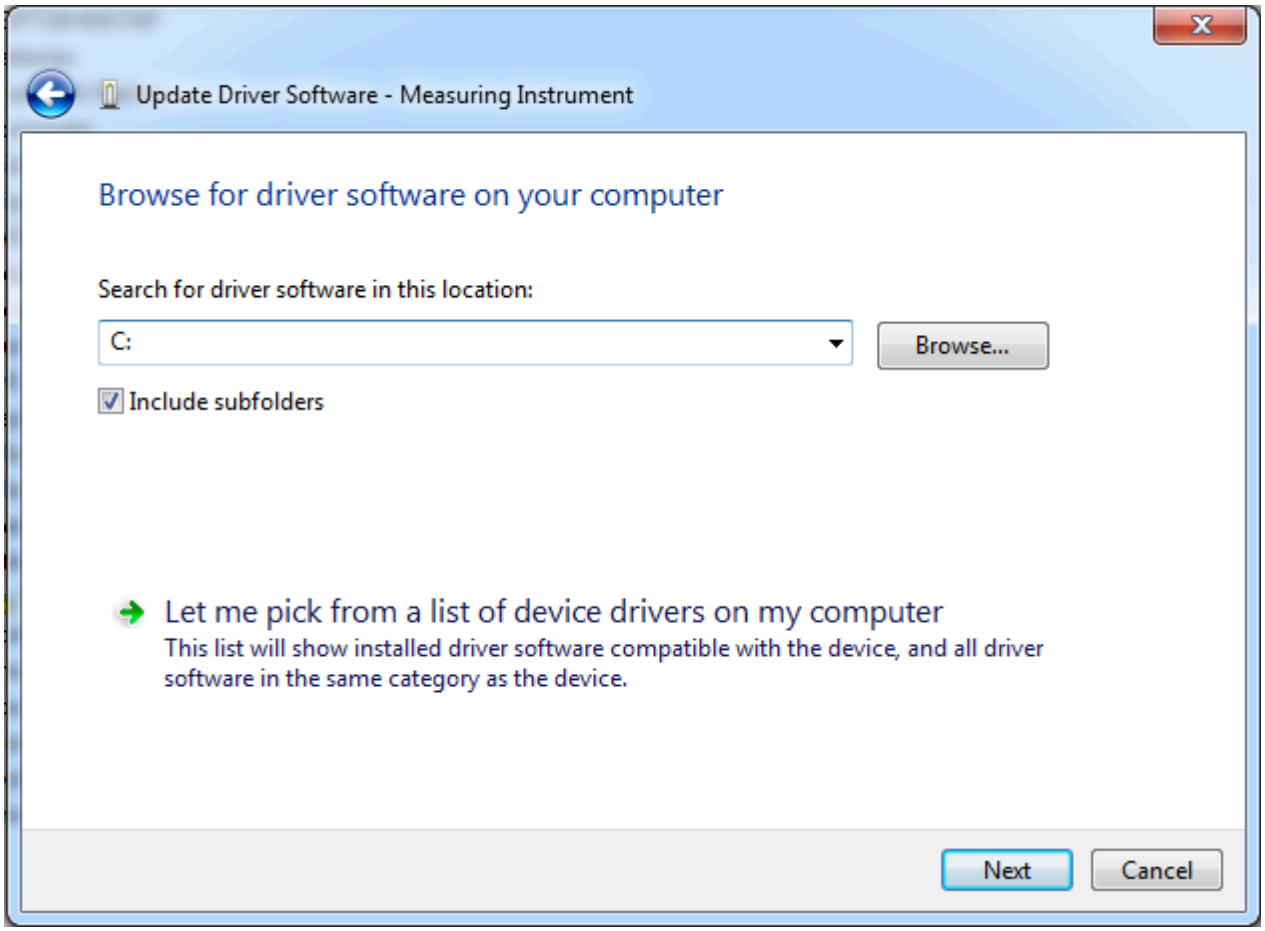
Open Device Manager, right-click [Other devices]-[Measuring Instruments], and then click [Update Driver Software]. As shown in the following screenshot, the warning symbol will be added to [Measuring Instruments] if the driver installation has failed.



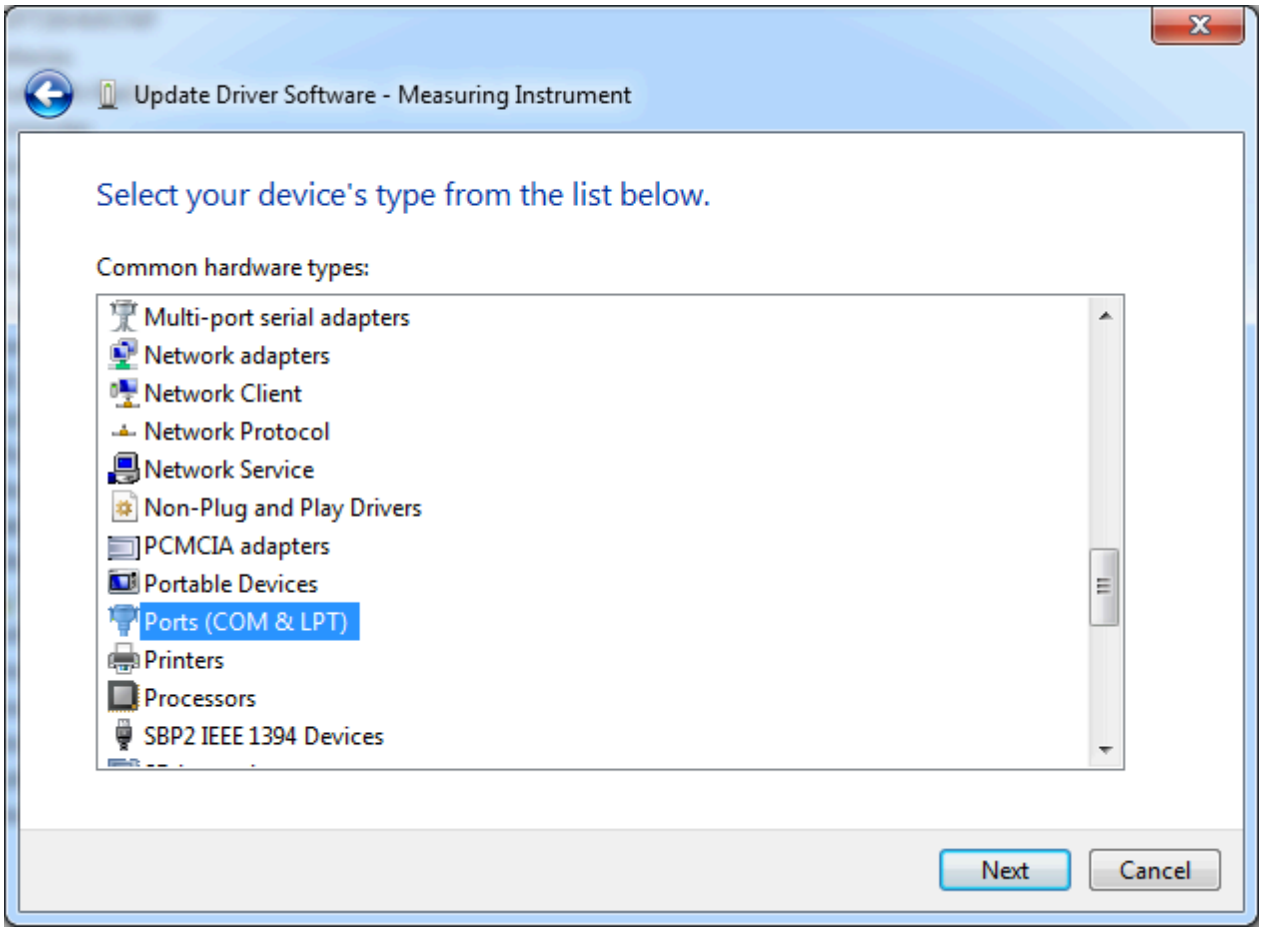
Click [Browse my computer for driver software].



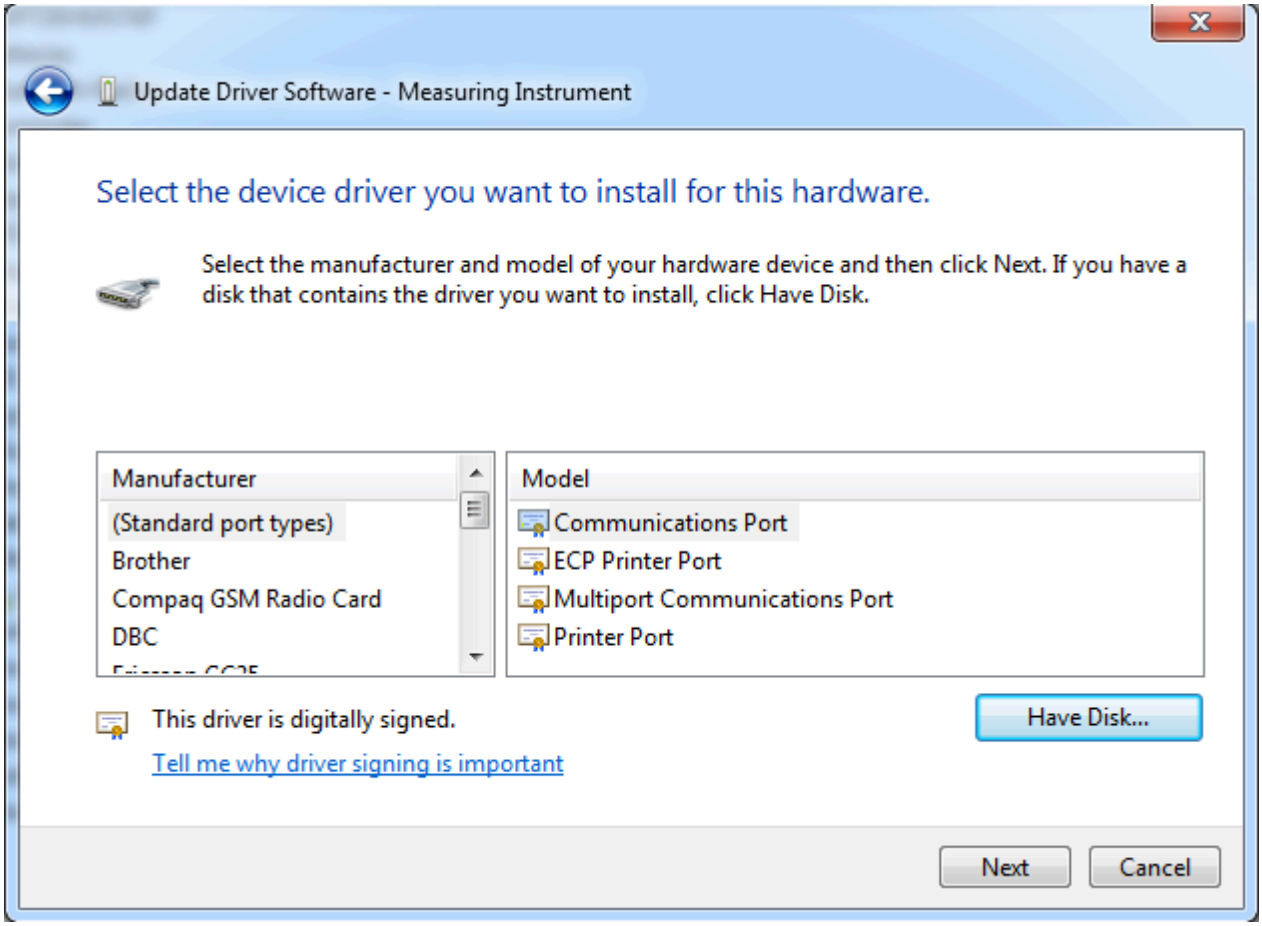
Click [Let me pick from a list of device drivers on my computer].



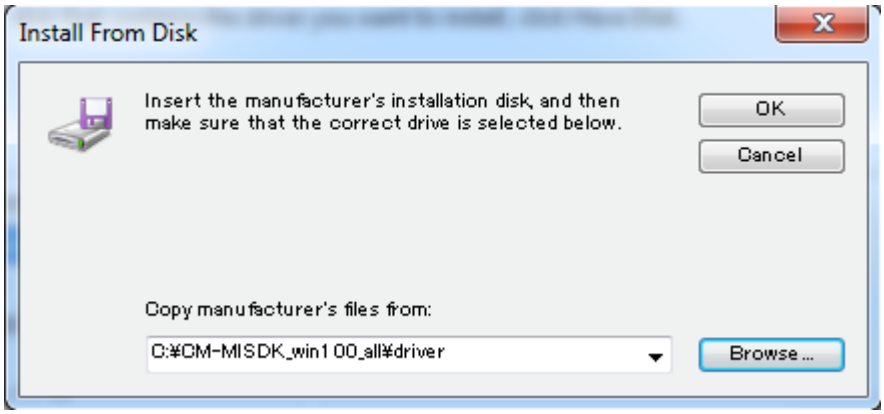
From the list of common hardware types, click [Ports (COM & LPT)], and then click [Next].



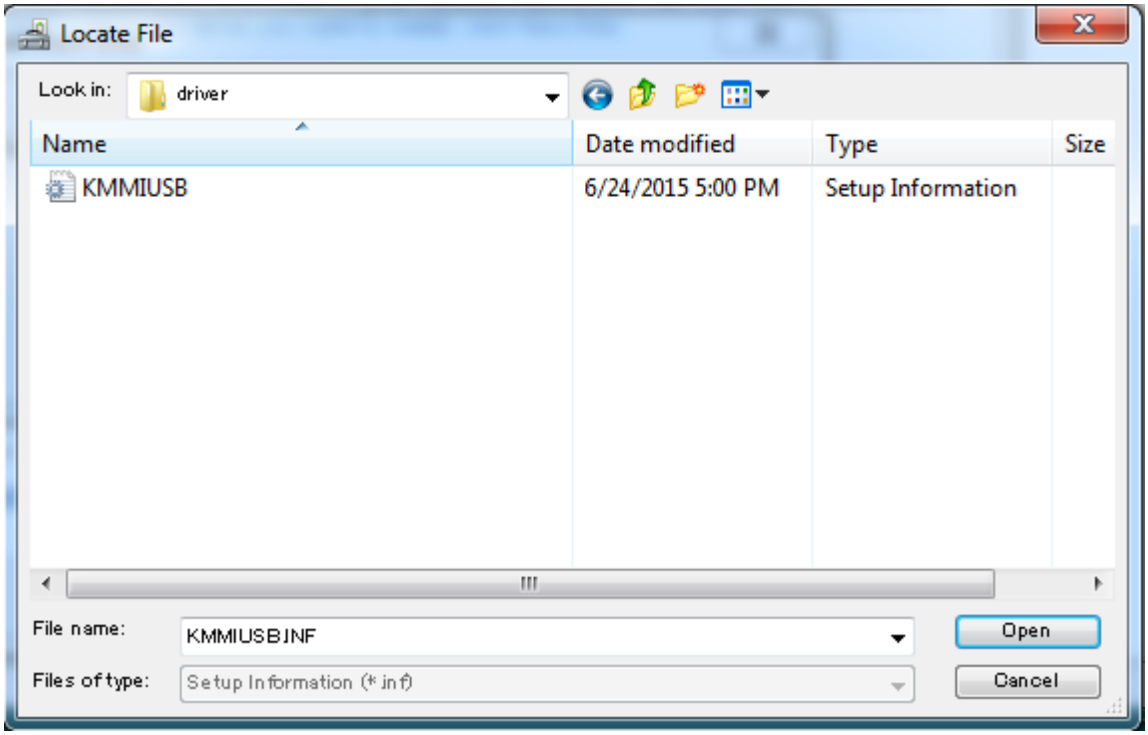
Click [Have Disk].



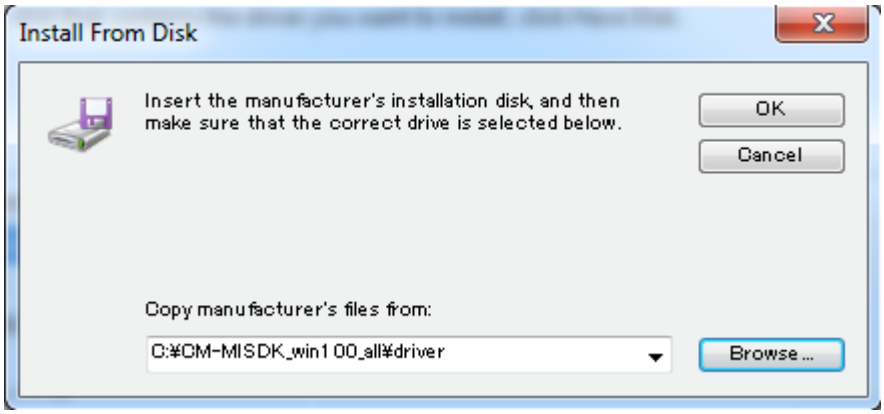
Click [Browse].



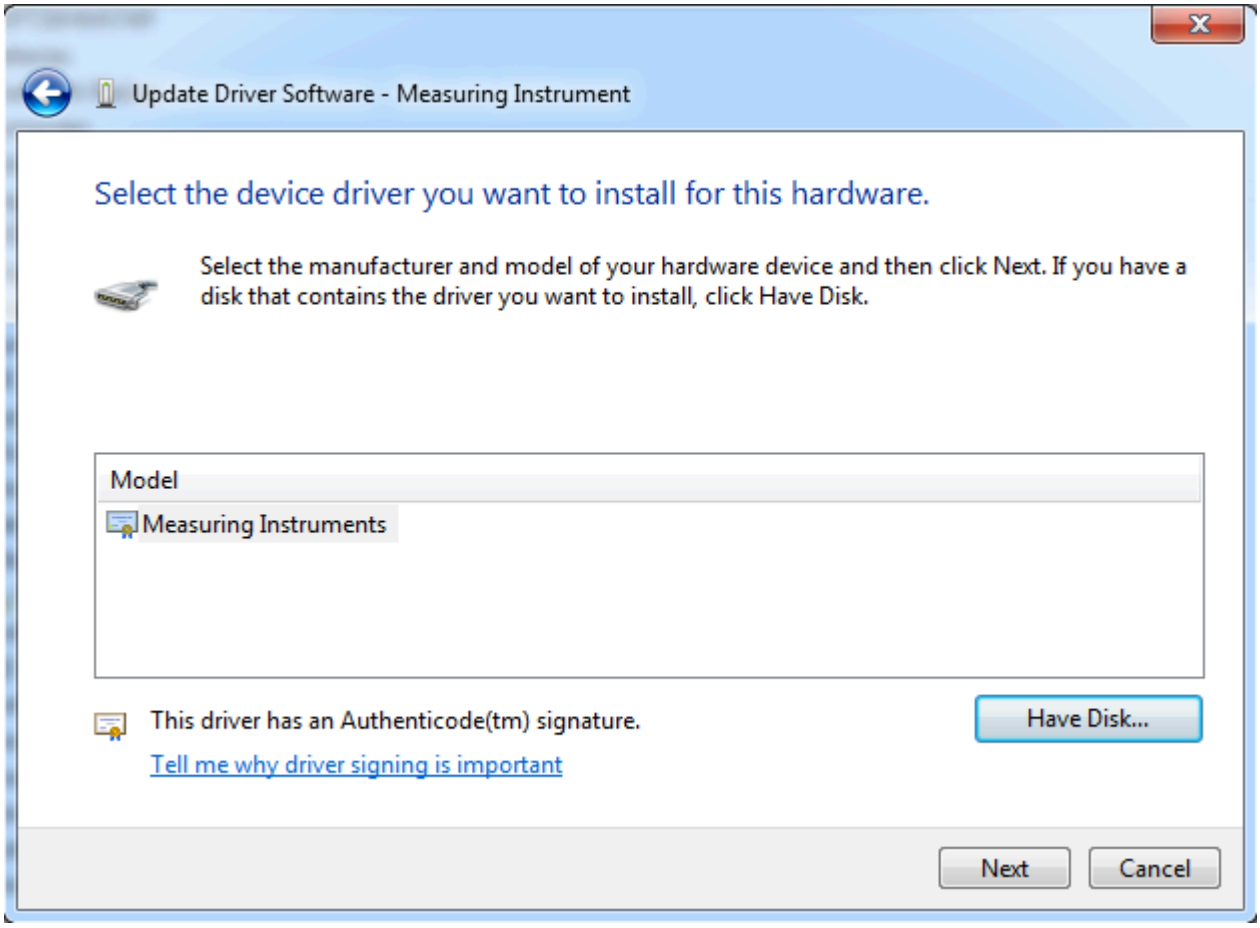
Select the "KMMIUSB.INF" file in "cm-misdk_verXXXrX/driver/", and then click [Open].



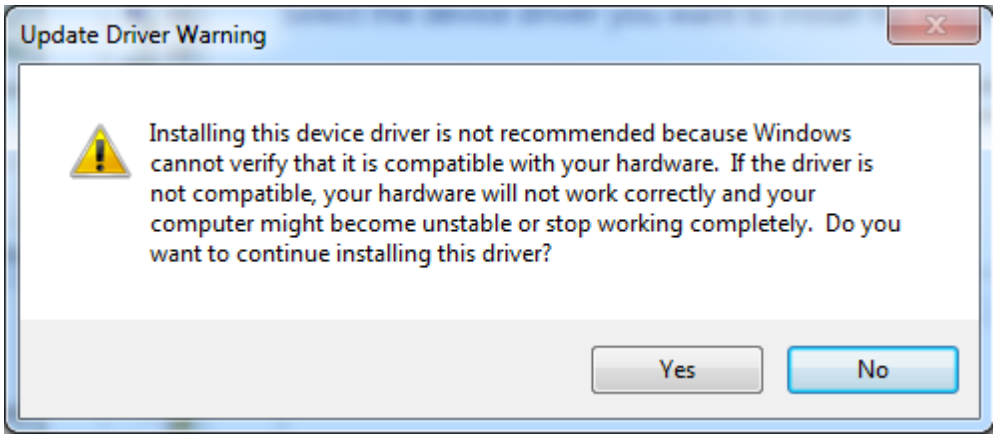
Click [OK].



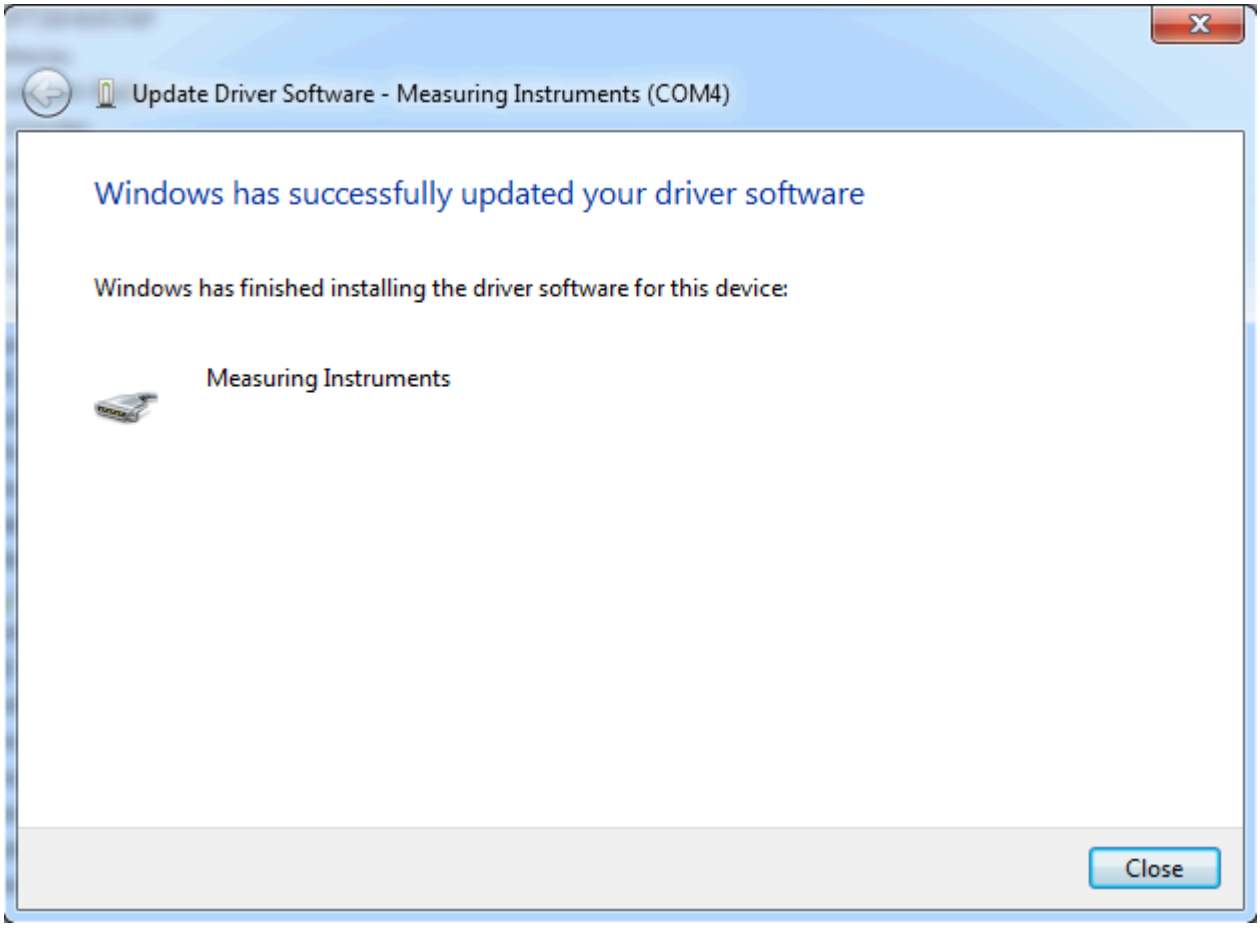
Click [Next].



Click [Yes].

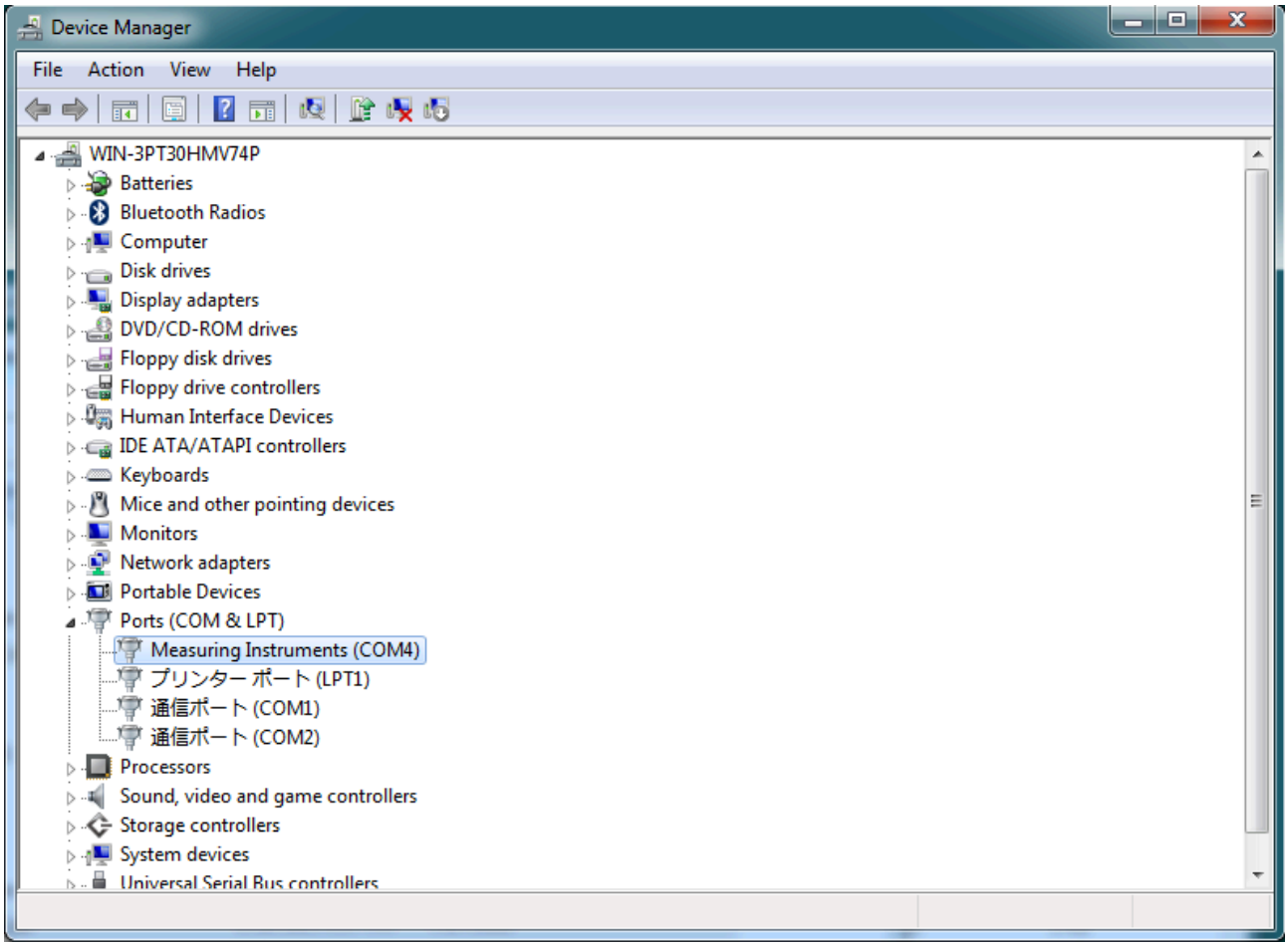


Click [Close] when you have confirmed that the installation has finished.



The warning symbol on [Measuring Instruments] should no longer be displayed. Next, confirm that the COM number is displayed, and then close the window by clicking x at the top right.

(COM11 is shown in the following screenshot, but the actual COM number will depend on the PC.)



This concludes installation of the driver.

Appendix C. List of parameters settable by instrument and version

Parameters that can be set on the instrument will depend on the instrument itself and its version. Refer to the following lists.

Warning status

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
KmWrBattery	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
KmWrCalibration	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
KmWrPreAnnualCalibraton	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
KmWrAnnualCalibraton	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
KmWrLampForColor	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
KmWrOutOfColorRange	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
KmWrOutOfGlossRange	✓	✓	×	×	✓	✓	×	×	×	×
KmWrLampForGloss	✓	✓	×	×	✓	✓	×	×	×	×

Calibration status

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
StatusZero	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
StatusWhite	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
StatusGloss	✓	✓	×	×	✓	✓	×	×	×	×
StatusMeasure	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
StatusMeasureWrn	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
StatusUser	✓	✓	×	×	✓	✓	✓	✓	✓	✓

Fluorescence adjustment conditions

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
UVADJ_PROFILE						✓		✓		
UVADJ_WI						✓		✓		
UVADJ_TINT						×		×		
UVADJ_WITINT						✓		✓		
UVADJ_BRIGHTNESS						✓		✓		

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
UVADJ_GG						✓		✓		

Fluorescence coefficient data type

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
UVADJ_DATATYPE_SCI						✓		✓		
UVADJ_DATATYPE_SCE						✓		✓		

Measurement area

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
AREA_MAV	✓	✓			✓	✓	✓	✓		
AREA_SAV	✓	✓			✓	✓	✓	✓		
AREA_LAV	×	×			×	×	×	×		
AREA_LMAV	×	×			×	×	×	×		

Measurement type

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
MEASTYPE_REF										
MEASTYPE_TRA										

Measurement angle

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
MEAS_ANGLE_M15			✓	✓						
MEAS_ANGLE_15			✓	✓						
MEAS_ANGLE_25			✓	✓						
MEAS_ANGLE_45			✓	✓						
MEAS_ANGLE_75			✓	✓						

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
MEAS_ANGLE_110			✓	✓						

Tilt detection

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
OFF			✔	✔						
ON			✔	✔						

Measurement mode

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
MEASMODE_COLORANDGLOSS	✓	✓			✓	✓	×	✓	×	×
MEASMODE_COLORONLY	✓	✓			✓	✓	✓	✓	✓	✓
MEASMODE_GLOSSONLY	✓	✓			✓	✓	×	✓	×	×
MEASMODE_OPACITY	×	×			✓	✓	✓	✓	✓	✓

Specular component

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
SC_SCI					✓	✓	✓	✓	✓	✓
SC_SCE					✓	✓	✓	✓	✓	✓
SC_SCIE					✓	✓	✓	✓	✓	✓

UV condition

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
UV_100					✓	✓	✓	✓		
UV_CUT400					✓	✓	✓	✓		
UV_CUT420					×	×	×	×		
UV_CUT400N					×	(*)	×	(*)		

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
UV_CUT400L					×	×	×	×		
UV_CUT420N					×	×	×	×		
UV_CUT420L					×	×	×	×		
UV_100_CUT400					×	×	×	×		
UV_100_CUT420					×	×	×	×		
UV_100_CUT400N					×	✓	×	✓		
UV_100_CUT400L					×	×	×	×		
UV_100_CUT420N					×	×	×	×		
UV_100_CUT420L					×	×	×	×		

(*): * Used for data properties and jobs.

Auto average count

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
Minimum	1	1	1	1	1	1	1	1	1	1
Maximum	10	10	10	10	10	10	10	10	10	10

Manual average count

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
Minimum	1	1	1	1	1	1	1	1	1	1
Maximum	30	30	10	10	30	30	30	30	30	30

Manual averaging save mode

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
SAVEMODE_AUTO	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SAVEMODE_MANUAL	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

SMC setting

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
OFF					✓	✓	✓	✓	✓	
ON					✓	✓	✓	✓	✓	

SMC number of times

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
Minimum					3	3	3	3	3	
Maximum					10	10	10	10	10	

Display type

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
DISPTYPE_ABS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DISPTYPE_DIF	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DISPTYPE_ABSDIF	✓	✓	×	×	✓	✓	✓	✓	✓	✓
DISPTYPE_CUSTOM	✓	✓	×	×	✓	✓	✓	✓	✓	✓
DISPTYPE_GRAPH_ABS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DISPTYPE_GRAPH_DIF	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DISPTYPE_GRAPH_REF	✓	✓	×	×	✓	✓	✓	✓	✓	✓
DISPTYPE_PASS_FAIL	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DISPTYPE_MI	×	×	✓	✓	×	×	×	×	×	×
DISPTYPE_GRAPH_LINE	×	×	✓	✓	×	×	×	×	×	×
DISPTYPE_AUDI2000_EC	×	×	×	✓	×	×	×	×	×	×
DISPTYPE_AUDI2000_EP	×	×	×	✓	×	×	×	×	×	×

Observer

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
OBS_02	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
OBS_10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Illuminant

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
ILL_NONE	(*)	(*)	(*)	(*)	(*)	(*)	(*)	(*)	(*)	(*)
ILL_A	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ILL_C	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ILL_D50	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ILL_D65	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ILL_ID50	✓	✓	×	×	✓	✓	✓	✓	✓	✓
ILL_ID65	✓	✓	×	×	✓	✓	✓	✓	✓	✓
ILL_F2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ILL_F6	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ILL_F7	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ILL_F8	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ILL_F10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ILL_F11	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ILL_F12	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ILL_USER1	✓	✓	×	✓	✓	✓	✓	✓	✓	×

(*) Only the second illuminant can be set.

Color space

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
COLOR_LAB	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
COLOR_LCH	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
COLOR_HLAB	✓	✓	×	×	✓	✓	✓	✓	✓	×
COLOR_YXY	✓	✓	×	×	✓	✓	✓	✓	✓	✓
COLOR_XYZ	✓	✓	×	×	✓	✓	✓	✓	✓	✓
COLOR_MUNSELL_C	✓	✓	×	×	✓	✓	✓	✓	✓	✓

Color equation

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
EQUATION_DE1976	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EQUATION_CMC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EQUATION_DE1994	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EQUATION_DE2000	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EQUATION_DEH	✓	✓	×	×	✓	✓	✓	✓	✓	×
EQUATION_DEP	×	×	✓	✓	×	×	×	×	×	×
EQUATION_DEC	×	×	✓	✓	×	×	×	×	×	×
EQUATION_DE99o	×	✓	×	✓	✓	✓	✓	✓	✓	×

Custom items

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
None	✓	✓			✓	✓	✓	✓	✓	✓
L*	✓	✓			✓	✓	✓	✓	✓	✓
a*	✓	✓			✓	✓	✓	✓	✓	✓
b*	✓	✓			✓	✓	✓	✓	✓	✓
C*	✓	✓			✓	✓	✓	✓	✓	✓
h	✓	✓			✓	✓	✓	✓	✓	✓
L(Hunter)	✓	✓			✓	✓	✓	✓	✓	×
a(Hunter)	✓	✓			✓	✓	✓	✓	✓	×
b(Hunter)	✓	✓			✓	✓	✓	✓	✓	×
X	✓	✓			✓	✓	✓	✓	✓	✓

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
Y	✓	✓			✓	✓	✓	✓	✓	✓
Z	✓	✓			✓	✓	✓	✓	✓	✓
x	✓	✓			✓	✓	✓	✓	✓	✓
y	✓	✓			✓	✓	✓	✓	✓	✓
H	✓	✓			✓	✓	✓	✓	✓	✓
V	✓	✓			✓	✓	✓	✓	✓	✓
C	✓	✓			✓	✓	✓	✓	✓	✓
WI(E313-73)	✓	✓			✓	✓	✓	✓	✓	✓
WI(CIE)	✓	✓			✓	✓	✓	✓	✓	×
Tint(CIE)	✓	✓			✓	✓	✓	✓	✓	×
YI(E313)	✓	✓			✓	✓	✓	✓	✓	×
YI(D1925)	✓	✓			✓	✓	✓	✓	✓	✓
B(ISO)	✓	✓			✓	✓	✓	✓	✓	×
GU	✓	✓			✓	✓	×	×	×	×
UserE1	✓	✓			✓	✓	✓	✓	✓	✓
UserC1	✓	✓			✓	✓	✓	✓	✓	✓
UserE2	✓	✓			✓	✓	✓	✓	✓	✓
UserC2	✓	✓			✓	✓	✓	✓	✓	✓
UserE3	✓	✓			✓	✓	✓	✓	✓	✓
UserC3	✓	✓			✓	✓	✓	✓	✓	✓
8° gloss	×	×			×	×	✓	✓	✓	×
WI(Ganz)	×	×			×	✓	×	✓	×	×
Tint(Ganz)	×	×			×	✓	×	✓	×	×
ΔL*	✓	✓			✓	✓	✓	✓	✓	✓
Δa*	✓	✓			✓	✓	✓	✓	✓	✓
Δb*	✓	✓			✓	✓	✓	✓	✓	✓
ΔC*	✓	✓			✓	✓	✓	✓	✓	✓
ΔH*	✓	✓			✓	✓	✓	✓	✓	✓
ΔL(Hunter)	✓	✓			✓	✓	✓	✓	✓	×
Δa(Hunter)	✓	✓			✓	✓	✓	✓	✓	×

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
Δb(Hunter)	✓	✓			✓	✓	✓	✓	✓	×
ΔX	✓	✓			✓	✓	✓	✓	✓	✓
ΔY	✓	✓			✓	✓	✓	✓	✓	✓
ΔZ	✓	✓			✓	✓	✓	✓	✓	✓
Δx	✓	✓			✓	✓	✓	✓	✓	✓
Δy	✓	✓			✓	✓	✓	✓	✓	✓
ΔWI(E313-73)	✓	✓			✓	✓	✓	✓	✓	✓
ΔWI(CIE)	✓	✓			✓	✓	✓	✓	✓	×
ΔTint(CIE)	✓	✓			✓	✓	✓	✓	✓	×
ΔYI(E313)	✓	✓			✓	✓	✓	✓	✓	×
ΔYI(D1925)	✓	✓			✓	✓	✓	✓	✓	✓
ΔB(ISO)	✓	✓			✓	✓	✓	✓	✓	×
ΔGU	✓	✓			✓	✓	×	×	×	×
MI	✓	✓			✓	✓	✓	✓	✓	✓
ΔE*ab	✓	✓			✓	✓	✓	✓	✓	✓
CMC	✓	✓			✓	✓	✓	✓	✓	✓
ΔE*94	✓	✓			✓	✓	✓	✓	✓	✓
ΔE00	✓	✓			✓	✓	✓	✓	✓	✓
ΔE(Hunter)	✓	✓			✓	✓	✓	✓	✓	×
ΔE99o	×	✓			✓	✓	✓	✓	✓	×
StrengthXYZ	×	×			✓	✓	✓	✓	✓	×
StrengthX	×	×			✓	✓	✓	✓	✓	×
StrengthY	×	×			✓	✓	✓	✓	✓	×
StrengthZ	×	×			✓	✓	✓	✓	✓	×
GreyScale	×	×			✓	✓	✓	✓	✓	×
ΔWI(Ganz)	×	×			×	✓	×	✓	×	×
ΔTint(Ganz)	×	×			×	✓	×	✓	×	×

Irradiation direction to display

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
DIRECTION_DP				✓						
DIRECTION_L				✓						

Irradiation direction

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
LDIRECTION_NONE			✓	✓						
LDIRECTION_L			✓	✓						
LDIRECTION_R			✓	✓						
LDIRECTION_DP			✓	✓						

Target filter

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
FILTER_OFF	✓	✓			✓	✓	✓	✓	✓	✓
FILTER_SAVE	✓	✓			✓	✓	✓	✓	✓	✓
FILTER_GROUP	✓	✓			✓	✓	✓	✓	✓	✓

Target protection

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
OFF	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ON	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Group name

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
Name size	30	30			30	30	30	30	30	30

Tolerance ID

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
ΔL*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Δa*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Δb*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ΔC*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ΔH*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ΔL(Hunter)	✓	✓	×	×	✓	✓	✓	✓	✓	×
Δa(Hunter)	✓	✓	×	×	✓	✓	✓	✓	✓	×
Δb(Hunter)	✓	✓	×	×	✓	✓	✓	✓	✓	×
ΔX	✓	✓	×	×	✓	✓	✓	✓	✓	✓
ΔY	✓	✓	×	×	✓	✓	✓	✓	✓	✓
ΔZ	✓	✓	×	×	✓	✓	✓	✓	✓	✓
Δx	✓	✓	×	×	✓	✓	✓	✓	✓	✓
Δy	✓	✓	×	×	✓	✓	✓	✓	✓	✓
ΔWI(E313-73)	✓	✓	×	×	✓	✓	✓	✓	✓	✓
ΔWI(CIE)	✓	✓	×	×	✓	✓	✓	✓	✓	×
ΔTint(CIE)	✓	✓	×	×	✓	✓	✓	✓	✓	×
ΔYI(E313)	✓	✓	×	×	✓	✓	✓	✓	✓	×
ΔYI(D1925)	✓	✓	×	×	✓	✓	✓	✓	✓	✓
ΔB(ISO)	✓	✓	×	×	✓	✓	✓	✓	✓	×
ΔGU	✓	✓	×	×	✓	✓	×	×	×	×
MI	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ΔE*ab	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CMC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ΔE*94	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ΔE00	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ΔE(Hunter)	✓	✓	×	×	✓	✓	✓	✓	✓	×
ΔEp(DIN6175)	×	×	✓	✓	×	×	×	×	×	×
ΔEc(DIN6175)	×	×	✓	✓	×	✓	×	✓	×	×
ΔFF	×	×	✓	✓	×	✓	×	✓	×	×

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
ΔE99o	×	✓	×	✓	✓	✓	✓	✓	✓	×
ΔEc(Audi2000)	×	×	×	✓	×	×	×	×	×	×
ΔEc average (Audi2000)	×	×	×	✓	×	×	×	×	×	×
ΔEc maximum (Audi2000)	×	×	×	✓	×	×	×	×	×	×
ΔEp(Audi2000)	×	×	×	✓	×	×	×	×	×	×
ΔEp average (Audi2000)	×	×	×	✓	×	×	×	×	×	×
ΔEp maximum (Audi2000)	×	×	×	✓	×	×	×	×	×	×
ΔStrengthXYZ	×	×	×	×	✓	✓	✓	✓	✓	×
ΔstrengthX	×	×	×	×	✓	✓	✓	✓	✓	×
ΔstrengthY	×	×	×	×	✓	✓	✓	✓	✓	×
ΔstrengthZ	×	×	×	×	✓	✓	✓	✓	✓	×
Opacity difference	×	×	×	×	✓	✓	✓	✓	✓	×
Grayscale difference	×	×	×	×	✓	✓	✓	✓	✓	×
ΔWI(Ganz)	×	×	×	×	×	✓	×	✓	×	×
ΔTint(Ganz)	×	×	×	×	×	✓	×	✓	×	×

Warning level

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
Minimum	0	0	0	0	0	0	0	0	0	0
Maximum	100	100	100	100	100	100	100	100	100	100

Instrument mode

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
INSTRUMENTMODE_NORMAL					✓	✓	✓	✓	✓	✓
INSTRUMENTMODE_SIMPLE					✓	✓	✓	✓	✓	✓

User type

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
USERTYPE_ADMIN	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
USERTYPE_WORKER	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Automatic printing

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
OFF	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ON	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Display brightness

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
Minimum	0	0	0	0	0	0	0	0	0	0
Maximum	4	4	4	4	4	4	4	4	4	4

Display direction

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
SCREENDIR_0			✓	✓	✓	✓	✓	✓	✓	✓
SCREENDIR_180			✓	✓	✓	✓	✓	✓	✓	✓

Sound

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
OFF	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ON	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Calibration interval

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
Minimum	1	1	1	1	1	1	1	1	1	1
Maximum	24	24	24	24	24	24	24	24	24	24

User calibration

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
OFF	✓	✓			✓	✓	✓	✓	✓	
ON	✓	✓			✓	✓	✓	✓	✓	

Periodical calibration notification

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
OFF	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ON	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Skip zero calibration on/off

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
OFF					✓	✓	✓	✓	✓	✓
ON					✓	✓	✓	✓	✓	✓

Date format

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
DF_YYYYMMDD	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DF_MMDDYYYY	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DF_DDMMYYYY	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Language

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
LANGUAGE_ENGLISH	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
LANGUAGE_JAPANESE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
LANGUAGE_GERMAN	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
LANGUAGE_FRENCH	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
LANGUAGE_SPANISH	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
LANGUAGE_ITALIAN	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
LANGUAGE_CHINESE_S	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
LANGUAGE_PORTUGUESE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
LANGUAGE_RUSSIAN	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
LANGUAGE_POLISH	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
LANGUAGE_TURKISH	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Power savings

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
Minimum		0		0	0	0	0	0	0	0
Maximum		60		60	60	60	60	60	60	60

Job

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
Job name (size)					20	20	20	20	20	20
Data name (size)					30	30	30	30	30	30
Comment (size)					100	100	100	100	100	100

Date/time type

	25cG 1.0x	25cG 1.1x	M6 1.0x	M6 1.1x	26dG 1.0x	26dG 1.1x	26d 1.0x	26d 1.1x	25d	23d
DATETYPE_COLOR	✓	✓			✓	✓				
DATETYPE_GLOSS	✓	✓			✓	✓				

CM-MISDK (ANSI C Version) Reference Manual