



SII SDK for Windows Application Programmer's Guide

Rev.03

[Products]

DSP-A01 Series

Seiko Instruments Inc.

Rev.01 July 2019
Rev.02 January 2020
Rev.03 November 2024


Copyright©2019-2024 by Seiko Instruments Inc.
All rights reserved.

Microsoft® and Windows® are registered trademarks of Microsoft Corporation in the U.S., Japan, and other countries.

Bluetooth® is a registered trademark of Bluetooth SIG, Inc.

Seiko Instruments Inc. (hereinafter referred to as "SII") has prepared this manual for use by SII personnel, licensees, and customers. The information contained herein is the property of SII and shall not be reproduced in whole or in part without the prior written approval of SII.

SII reserves the right to make changes without notice to the specifications and materials contained herein and shall not be responsible for any damages (including consequential) caused by reliance on the materials presented, including but not limited to typographical, arithmetic, or listing errors.

SII  is a trademark of Seiko Instruments Inc.

Introduction

This manual describes display functions of "SII SDK for Windows" (hereinafter referred to as the "SDK") provided by Seiko Instruments Inc. (hereinafter referred to as "SII").
The SDK runs on the printer driver stated in "Target Product".

Target Product

The product covered by this manual is as follows.

Product	Interface	Description in This Manual
DSP-A01	USB	Display

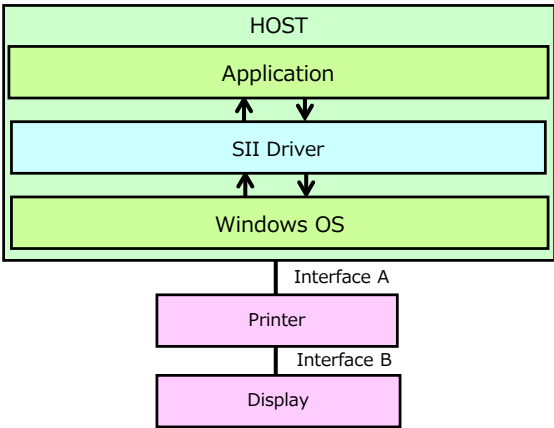
Supported Driver

SII driver is required for using Display.
Supported SII drivers are described below.
There are 2 types for connecting Display.

1. When Display is used by connecting to SII printer (hereinafter described as "use via a printer")

SII Driver	Interface A	Printer	Interface B
SII Printer Driver for Windows	USB Bluetooth TCP/IP	RP-F10 series	USB ^{*1}

^{*1}: USB Standard: Full-Speed



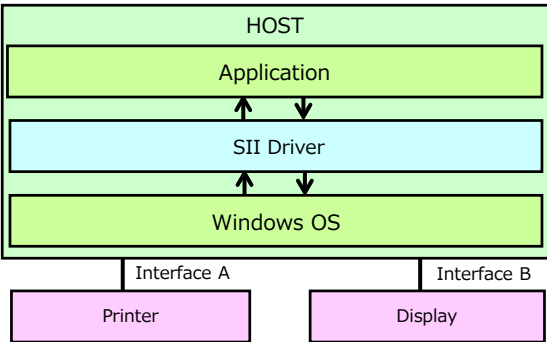
Configuration image

2. When Display is used alone (hereinafter described as "use alone")

SII Driver	Interface A	Printer ^{*1}	Interface B
SII Printer Driver for Windows	-	-	USB ^{*2}

^{*1}: SII printer that supports Display is as follows.
· RP-F10 series

^{*2}: USB Standard: Full-Speed



Configuration image

Notation in This Manual

The notation in this manual is described.

Terms

The terms used in this manual are defined below.

Printer

Term	Description
Technical Reference for Printer	The following Technical Reference ·RP-F10 SERIES THERMAL PRINTER TECHNICAL REFERENCE
Printer command	Command for controlling the printer described in "Technical Reference for Printer".
ASB setting command (ASB: Automatic Status Back)	Printer command "Enable/Disable Automatic Status Back" See the chapter of "Command Functions" in "Technical Reference for printer" for details of printer commands.

Display

Term	Description
Technical Reference for Display	The following Technical Reference. ·DSP-A01 SERIES CUSTOMER DISPLAY TECHNICAL REFERENCE
Display command	Command for controlling Display described in "Technical Reference for Display".
Slide	The image data of the screen size (Width 480 px × Height 272 px). Displays as a standby screen and as a backscreen superimposed on templates.
Template	The stylized form having elements that can set attributes such as drawing areas and mapping positions. The elements include text elements (text data), img elements (image data), barcode elements (barcode data), and qr elements (QR Code data). To register templates, define a map ID each for an element to map. Registered image data or text data is shown on Display by updating the screen after selecting a template and specifying its map ID. The data in the template is required to be specified XML file format. The maximum size of template data is width 480 px × height 272 px.
Map ID	An ID defined to an element which is holding positional information or modification information when a template is registered.
Macro	A function to register multiple APIs in order of execution, and execute automatically when an event occurs.

Term	Description
Contents	Data that can be registered in Display's memory area, such as slide data, image data, templates, and macros. ID information and data names can be retrieved except for macros.
Event	An event which is defined by "Event Notification" in Display commands.

Chapter 1	Overview	1-1
1.1	Operating Conditions	1-1
1.2	Registered Data at Factory	1-1

Chapter 2	Installation	2-1
------------------	---------------------	------------

Chapter 3	Win32 API	3-1
3.1	Development Language	3-1
3.2	Library File	3-1
3.3	API List	3-2
3.3.1	Common API	3-2
3.3.2	Specific API	3-2
3.4	API Details	3-4
3.4.1	Common API	3-4
	OpenMonPrinter	3-4
	CloseMonPrinter	3-5
	LockPrinter	3-5
	UnlockPrinter	3-6
	DirectIO	3-7
	DirectIOEx	3-8
3.4.2	Specific API	3-10
	Screen Display	3-10
	DispShowTemplate	3-10
	DispShowSlide	3-11
	DispTurnOnScreen	3-12
	DispEnterStandbyMode	3-13
	DispExecuteMacro	3-13
	Template	3-15
	DispRegisterTemplate	3-15
	DispUnregisterTemplate	3-16
	DispSelectTemplate	3-17
	DispSetTemplateImageData	3-18
	DispSelectTemplateTextObject	3-19
	DispSetTemplateTextAlignment	3-20
	DispSetTemplateTextLeftMargin	3-21
	DispSetTemplateTextBold	3-22

DispSetTemplateTextUnderline	3-23
DispSetTemplateTextSize	3-24
DispSetTemplateTextFont.....	3-25
DispSetTemplateTextRegisteredFont	3-26
DispSetCodePage	3-27
DispSetInternationalCharacter	3-28
DispSetTemplateTextRightSpacing	3-29
DispSetTemplateTextColor	3-30
DispSetTemplateTextLineSpacing	3-31
DispSetTemplateTextData	3-32
DispSetTemplateBarcodeData	3-33
DispSetTemplateQrCodeData	3-34
Image Registration.....	3-36
DispRegisterImageData.....	3-36
DispUnregisterImageData	3-37
DispRegisterSlideData.....	3-38
DispUnregisterSlideData	3-39
Character Registration	3-40
DispRegisterUserDefinedCharacter	3-40
DispUnregisterUserDefinedCharacter	3-41
DispRegisterOptionFont	3-41
DispUnregisterOptionFont	3-43
Macro	3-44
DispControlMacroRegistration.....	3-44
Memory Operation	3-46
DispDefragment.....	3-46
DispGetMemory	3-47
DispInitializeMemoryArea.....	3-47
Response Function	3-49
DispGetRegisteredStateList	3-49
DispGetObjectLabel.....	3-50
DispGetResponseRequest	3-51

Chapter 4 .NET API	4-1
4.1 Development Language	4-1
4.2 Library File	4-1
4.3 API List	4-2
4.3.1 Common API.....	4-2
4.3.2 Specific API.....	4-2

4.4	API Details	4-4
4.4.1	Property.....	4-4
	Property.....	4-4
	LastError	4-4
	IsValid	4-4
	Method	4-5
	OpenMonPrinter	4-5
	CloseMonPrinter	4-6
	LockPrinter	4-6
	UnlockPrinter	4-7
	DirectIOEx.....	4-7
4.4.2	Specific API.....	4-9
	Screen Display.....	4-9
	ShowTemplate	4-9
	ShowSlide	4-10
	TurnOnScreen	4-11
	EnterStandbyMode	4-11
	ExecuteMacro	4-12
	Template.....	4-13
	RegisterTemplate	4-13
	UnregisterTemplate.....	4-14
	SelectTemplate	4-14
	SetTemplateImageData	4-15
	SelectTemplateTextObject	4-16
	SetTemplateTextAlignment	4-16
	SetTemplateTextLeftMargin	4-17
	SetTemplateTextBold	4-18
	SetTemplateTextUnderline.....	4-19
	SetTemplateTextSize	4-20
	SetTemplateTextFont	4-21
	SetTemplateTextRegisteredFont	4-21
	SetCodePage.....	4-22
	SetInternationalCharacter.....	4-23
	SetTemplateTextRightSpacing.....	4-24
	SetTemplateTextColor.....	4-24
	SetTemplateTextLineSpacing	4-25
	SetTemplateTextData.....	4-26
	SetTemplateBarcodeData.....	4-27
	SetTemplateQrCodeData	4-28
	Image Registration.....	4-30

RegisterImageData	4-30
UnregisterImageData	4-31
RegisterSlideData	4-31
UnregisterSlideData	4-32
Character Registration	4-33
RegisterUserDefinedCharacter	4-33
UnregisterUserDefinedCharacter	4-34
RegisterOptionFont	4-34
UnregisterOptionFont	4-35
Macro	4-36
ControlMacroRegistration	4-36
Memory Operation	4-38
Defragment	4-38
GetMemory	4-38
InitializeMemoryArea	4-39
Response Function	4-40
GetRegisteredStateList	4-40
GetObjectLabel	4-41
GetResponseRequest	4-41

Chapter 5	Error Code List	5-1
------------------	------------------------	------------

5.1	Error Code List	5-1
-----	-----------------------	-----

Chapter 6	Argument Information	6-1
------------------	-----------------------------	------------

6.1	Contents Type	6-1
6.2	Alignment	6-1
6.3	Character Size	6-2
6.4	Character Font	6-2
6.5	Registered Font	6-2
6.6	Codepage	6-3
6.7	International Character	6-4
6.8	Character Color	6-4
6.9	Module Size	6-5
6.10	Error Correction Level	6-6
6.11	Data Mode	6-6
6.12	Macro Processing Selection	6-6
6.13	Memory Area	6-7

Chapter 1 Overview

This chapter describes the overview of the SDK.

The SDK includes the library file to directly control Displays provided for developers. And also, the SDK is provided with the driver and uses the driver to work.

Using the SDK allows to directly control Displays in an application development and to design the application independent of the port type.

The SDK includes the following library files:

- SDK for Win32 development environment (hereinafter, Win32 API)
- SDK for .NET development environment (hereinafter, .NET API)

For usage sample of the SDK, see the sample program provided for each development language.

1.1 Operating Conditions

The operating conditions of the SDK basically follow the operating environment of the driver, and the Display settings and limitations of the printer. For details of the operating environment of the driver, see "SII Printer Driver for Windows User's Guide" for the printer and Display described in "Supported Driver".

In addition, the following operating conditions must be met.

- When using the .NET API, the SDK requires .NET Framework Version 2.0 or later. When .NET Framework is uninstalled from the computer, .NET API cannot be used.
- All functions of the SDK are only available when the bidirectional support function is enabled (excluding the Display driver) and the printer spool function is disabled.

1.2 Registered Data at Factory

Registered data at the factory may be added or changed without prior notice for quality improvement.

In addition, it is necessary to use a template that has an appropriate encoding specified depending on the language setting and character code to be specified. See our web site for details of data to be registered at the factory.

<https://www.sii-ps.com/dspa01/>

Chapter 2 Installation

For the installation, see "SII Software Package for Windows Installation Guide".

Chapter 3 Win32 API

This chapter describes the Win32 API.

3.1 Development Language

The following development language is covered.

- Visual C++

3.2 Library File

The library file of the Win32 API is a dynamic link library format.

The library file has the following file name:

- SPSWL_API.dll

The library file is stored in the Windows system folder.

Use the library file without moving it from the folder. Except in special cases, there is no need to set a path to the folder where the library file is saved.

If the library file is moved to another location, the library file will not be updated properly when upgrading the driver.

3.3 API List

The APIs implemented in the Win32 API are as follows.

3.3.1 Common API

API	Function Summary
OpenMonPrinterA ^{*1} OpenMonPrinterW ^{*1}	Starts using the Win32 API and returns the API handle.
CloseMonPrinter	Ends using the Win32 API.
LockPrinter	Prohibits (Locks) all data transmission and hardware reset requests from other processes to the device.
UnlockPrinter	Unlocks the access prohibition (lock) from other processes by LockPrinter .
DirectIO	Transmits and receives binary data. (Received data does not include responses of the ASB setting command.)
DirectIOEx	Transmits and receives binary data.

^{*1}: Specifies arguments of strings with MBCS (MultiByte Character Set) or UNICODE (Unicode). Use API added the suffix 'A' for MBCS or 'W' for Unicode. Note that the suffix of 'A' or 'W' is omitted in the following descriptions.

3.3.2 Specific API

API	Function Summary
DispShowTemplate	Displays the template on the screen.
DispShowSlide	Displays the slide on the screen.
DispTurnOnScreen	Sets the screen backlight on/off.
DispEnterStandbyMode	Changes Display to Standby mode.
DispExecuteMacro	Executes the registered macro.
DispRegisterTemplateA ^{*1} DispRegisterTemplateW ^{*1}	Registers a template.
DispUnregisterTemplate	Deletes a registered template.
DispSelectTemplate	Selects a template.
DispSetTemplateImageData	Sets image data to show on a selecting template.
DispSelectTemplateTextObject	Selects the text element to start editing.
APIs that are valid for the text element selected a template with DispSelectTemplate and/or specified with DispSelectTemplateTextObject .	
DispSetTemplateTextAlignment	Sets alignment to text data.
DispSetTemplateTextLeftMargin	Sets left margin to text data.
DispSetTemplateTextBold	Sets bold character to text data.
DispSetTemplateTextUnderline	Sets underline to text data.
DispSetTemplateTextSize	Sets character size to text data.
DispSetTemplateTextFont	Sets a character font to text data.
DispSetTemplateTextRegisteredFont	Sets the registered font used for text data.

API	Function Summary
DispSetCodePage	Sets the codepage used for text data.
DispSetInternationalCharacter	Sets the international character used for text data.
DispSetTemplateTextRightSpacing	Sets the amount of right space to text data.
DispSetTemplateTextColor	Sets the character color used for text data.
DispSetTemplateTextLineSpacing	Sets line spacing to text data.
DispSetTemplateTextDataA *1 DispSetTemplateTextDataW *1 DispSetTemplateTextBDData *2	Inputs text data to show on the text element specified by DispSelectTemplateTextObject .
DispSetTemplateBarcodeData	Specifies a map ID of the barcode element on a selecting template, and inputs barcode data.
DispSetTemplateQrCodeData	Specifies a map ID of the qr element on a selecting template, and inputs QR Code data.
DispRegisterImageDataA *1 DispRegisterImageDataW *1	Registers image data.
DispUnregisterImageData	Deletes registered image data.
DispRegisterSlideDataA *1 DispRegisterSlideDataW *1	Registers slide data.
DispUnregisterSlideData	Deletes registered slide data.
DispRegisterUserDefinedCharacter	Registers user-defined characters.
DispUnregisterUserDefinedCharacter	Deletes registered user-defined characters.
DispRegisterOptionFontA *1 DispRegisterOptionFontW *1	Registers optional fonts.
DispUnregisterOptionFont	Deletes registered optional fonts.
DispControlMacroRegistration	Specifies start and end of macro registration.
DispDefragment	Optimizes memory area.
DispGetMemory	Gets the remaining capacity of memory area.
DispInitializeMemoryArea	Initializes memory area.
DispGetRegisteredStateList	Gets the presence or absence of contents registration.
DispGetObjectLabelA ※1 DispGetObjectLabelW ※1	Gets the label names of registered contents.
DispGetResponseRequest	Gets the specified response codes when this API is executed.

*1: Specifies arguments of strings by MBCS (MultiByte Character Set) or UNICODE (Unicode). Use API added the suffix 'A' for MBCS or 'W' for Unicode. Note that the suffix of 'A' or 'W' is omitted in the following descriptions.

*2: Specifies arguments of strings by binary data.

Reference

- In order to use the Display API, template registration is required to be specified in XML format, and it is necessary to refer to "Technical Reference for Display" for details on the necessary elements.
It is also recommended that read well "Technical Reference for Display" regarding how to use slides, templates, and macros, and state transitions in Display status.

3.4 API Details

3.4.1 Common API

OpenMonPrinter

Starts using the Win32 API and returns the API handle.

```
INT OpenMonPrinter(  
    INT i_type,  
    LPCTSTR i_prt )
```

Parameters

i_type
Open type
3 (fixed)
Specify 3 in both cases when Display is used via a printer and when Display is used alone.

i_prt
Name of the device that uses the Win32
Specifies the device name (friendly name).

Return value

On success: Returns the API handle to identify the device.
On failure: Returns an error code.
See "Chapter 5 Error Code List" for details of the error code.

Remarks

- The number of API handles that can be retrieved simultaneously in 1 process is up to 8. The number is up to 126 in total processes.
- When the API handle retrieved by this API is no longer used, be sure to disable it by **CloseMonPrinter**.
- Set the driver to the following connection destination.
When Display is used via a printer:
Specify the printer port.
When Display is used alone and the printer driver is used:
Specify the USB.
- This API succeeds even when the device is not connected or the Display's power is turned off.

CloseMonPrinter

Ends the Win32 API in use.

```
INT CloseMonPrinter(  
    INT i_hdl )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Remarks

- When the API handle specified by this API is being used in another API, control of this API does not return until the processing is completed.
- All settings and data that have been associated with the API handle are discarded by this API.

LockPrinter

Prohibits (Locks) all data transmission and hardware reset requests from other processes to the device.

```
INT LockPrinter(  
    INT i_hdl,  
    DWORD i_timeout )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_timeout

Timeout period

Specifies the time to wait for success of this API in milliseconds (ms).

The valid range is from 3000 to 90000.

When the value is specified less than 3000, the time is set to 3000 ms.

When the value is specified more than 90000, the time is set to 90000 ms.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Remarks

- When any APIs that directly access to the device are called from other processes during the time until when **UnlockPrinter** is called after calling this API, those APIs fail.
- Only other processes are locked by this API. Therefore, calls from its own processes are out of target to be locked regardless of API handles or threads.
- By this API, a multiple locking is possible up to 99 times with the same API handle. In order to release the lock, call **UnlockPrinter** as the same times as this API was called.
- When **LockPrinter** is being called by another process, and this API is called, the API fails.

UnlockPrinter

Unlocks the access prohibition (lock) from other processes by **LockPrinter**.

```
INT UnlockPrinter(  
    INT i_hdl )
```

Parameters

i_hdl
API handle
Specifies the API handle retrieved by **OpenMonPrinter**.

Return value

On success: Returns 0.
On failure: Returns an error code.
See "0Chapter 5 Error Code List" for details of the error code.

Remarks

- When **LockPrinter** has been called multiple times, this API must be called as the same times as **LockPrinter** is called to release the lock.

DirectIO

Transmits and receives binary data.

```
INT DirectIO(  
    INT i_hdl,  
    BYTE i_wlen,  
    LPBYTE i_wcmd,  
    LPBYTE io_rlen,  
    LPBYTE o_rbuf,  
    DWORD i_timeout,  
    BOOL i_flag )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_wlen

Size of data to transmit

Specifies the size of data to transmit.

i_wcmd

Buffer of data to transmit

Specifies the buffer in which the data to transmit is stored.

io_rlen

Size of data to receive

Specifies the maximum length of data to receive from the device.

Specifies 0 when data retrieving is not necessary.

When the control returns from the API, the retrieved receive data size is stored.

o_rbuf

Buffer of data to receive

Specifies the buffer that stores the data to get.

i_timeout

Timeout period

Specifies the time to wait for success of this API in milliseconds (ms).

The valid range is from 3000 to 90000.

When the value is specified less than 3000, the time is set to 3000 ms.

When the value is specified more than 90000, the time is set to 90000 ms.

i_flag

Receive operation flag

Specifies the receive operation.

TRUE: Continues receiving until any data is received or timeout occurs.

FALSE: Continues receiving until the receive data size is received or timeout occurs.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Remarks

- In the case of sending commands, data, and image data that do not allow interrupt of other data, call this API after calling **LockPrinter**. When **LockPrinter** is not called, data from other processes may interrupt.
- When **LockPrinter** is being called by another process and this API is called, the API fails.

- When the device is not connected or in a communication disabled state, this API fails.

DirectIOEx

Transmits and receives binary data.

```
INT DirectIOEx(
    INT i_hdl,
    DWORD i_wlen,
    LPBYTE i_wcmd,
    LPDWORD io_rlen,
    LPBYTE o_rbuf,
    DWORD i_timeout,
    BOOL i_flag,
    BYTE i_op )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_wlen

Size of data to transmit

Specifies the size of the data to transmit.

i_wcmd

Buffer of data to transmit

Specifies the buffer in which the data to transmit is stored.

io_rlen

Size of data to receive

Specifies the maximum length of data to receive from the device.

The maximum receive data size is 4096 bytes.

When the value is specified more than 4096, the size is set to 4096 bytes.

Specify 0 when data retrieving is not necessary.

When the control returns from the API, the received data size is stored.

o_rbuf

Buffer of data to receive

Specifies the buffer that stores the data to get.

i_timeout

Timeout period

Specifies the time to wait for success of this API in milliseconds (ms).

The valid range is from 3000 to 90000.

When the value is specified less than 3000, the time is set to 3000 ms.

When the value is specified more than 90000, the time is set to 90000 ms.

i_flag

Receive operation flag

Specifies the receive operation.

TRUE: Continues receiving until any data is received or timeout occurs.

FALSE: Continues receiving until the receive data size is received or timeout occurs.

i_op

Receive target option

Specifies the data to receive from the following.

When Display is used via a printer:

0: Gets the data excluding the response of the ASB setting command.

1: Gets the data including the response of the ASB setting command.

When Display is used alone and the printer driver is used:

Specify 0.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Remarks

- In the case of sending commands, data, and image data that do not allow interrupt of other data, call this API after calling **LockPrinter**. When **LockPrinter** is not called, data from other processes may interrupt.
- When **LockPrinter** is being called by another process, calling this API fails.
- When the device is not connected or in a communication disabled state, this API fails.

3.4.2 Specific API

Screen Display

DispShowTemplate

Displays the template on the screen.

```
INT DispShowTemplate(  
    INT i_hdl,  
    INT i_time )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_time

Display time

Specifies display time on the screen in milliseconds (ms).

The valid range is 0 to 25500.

When the value that exceeds 0 and less than 100 is specified, the time is set to 100 ms.

When the value exceeding 25500 is specified, the time is set to 25500 ms.

When the macro is registered by **DispControlMacroRegistration**:

When 0 is specified with *i_time*, a template is shown continuously.

When other than 0 is specified with *i_time*, a next template is shown after the display time is elapsed.

When the macro is not registered by **DispControlMacroRegistration**:

When 0 is specified with *i_time*, a template is shown continuously.

When other than 0 is specified with *i_time*, the display time returns to a previous template after the display time is elapsed.

In case of the previous screen has been updated with the display time other than 0, the screen is traced back to the template which was updated with the display time 0.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Remarks

- Updates the screen, and displays data being specified with the following APIs.
 - **DispSetTemplateImageData**
 - **DispSelectTemplateTextObject**
 - **DispSetTemplateTextAlignment**
 - **DispSetTemplateTextLeftMargin**
 - **DispSetTemplateTextBold**
 - **DispSetTemplateTextUnderline**

- **DispSetTemplateTextSize**
- **DispSetTemplateTextFont**
- **DispSetRegisteredFont**
- **DispSetCodePage**
- **DispSetInternationalCharacter**
- **DispSetTemplateTextRightSpacing**
- **DispSetTemplateTextColor**
- **DispSetTemplateTextLineSpacing**
- **DispSetTemplateTextData**
- **DispSetTemplateBarcodeData**
- **DispSetTemplateQrCodeData**

- In use via a printer, this API is ignored when Display is not connected to the printer.

DispShowSlide

Displays the slide on the screen.

```
INT DispShowSlide(
    INT i_hdl,
    INT i_slideID,
    INT i_time )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_slideID

Slide ID

Specifies the ID of slide data which was registered in **DispRegisterSlideData**.
The valid range is 0 to 91.

This API is ignored when slide data is not registered in the specified ID.

i_time

Display time

Specifies display time on the screen in milliseconds (ms).

The valid range is 0 to 25500.

When the value that exceeds 0 and less than 100 is specified, the time is set to 100 ms.

When the value exceeding 25500 is specified, the time is set to 25500 ms.

When the macro is registered by **DispControlMacroRegistration**:

When 0 is specified with *i_time*, a slide is shown continuously.

When other than 0 is specified with *i_time*, a next slide is shown after the display time is elapsed.

When the macro is not registered by **DispControlMacroRegistration**:

When 0 is specified with *i_time*, a slide is shown continuously.

When other than 0 is specified with *i_time*, the display time returns to a previous slide after the display time is elapsed.

In case of the previous screen has been updated with the display time other than 0, the screen is traced back to the slide which was updated with the display time 0.

Return value

- On success: Returns 0.
On failure: Returns an error code.
See "Chapter 5 Error Code List" for details of the error code.

Remarks

- A selecting template is deselected when this API is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispTurnOnScreen

Sets the screen backlight on/off.

```
INT DispTurnOnScreen(  
    INT i_hdl,  
    BOOL i_turnon )
```

Parameters

- i_hdl*
API handle
Specifies the API handle retrieved by **OpenMonPrinter**.
- i_turnon*
Operation flag
Specifies the operation.
TRUE: Turn on the screen backlight.
FALSE: Turn off the screen backlight.

Return value

- On success: Returns 0.
On failure: Returns an error code.
See "Chapter 5 Error Code List" for details of the error code.

Remarks

- In use via a printer, this API is ignored when Display is not connected to the printer.

DispEnterStandbyMode

Changes Display to Standby mode.

```
INT DispEnterStandbyMode(  
    INT i_hdl )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

Remarks

- This API is ignored during Standby mode or Guide mode is being displayed.
- A selecting template is deselected when this API is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispExecuteMacro

Executes the registered macro.

```
INT DispExecuteMacro(  
    INT i_hdl,  
    INT i_macroID,  
    INT i_cnt )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_macroID

Macro ID

Specifies the ID of macro to execute.

The valid range is 0 to 127.

This API is ignored when the macro is not registered in the specified ID.

i_cnt

The number of execution times

Specifies the number of times to execute macro.

The valid range is 0 to 255.

Continue the repeating when 0 is specified.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Remarks

- A selecting template is deselected.
- In use via a printer, this API is ignored when Display is not connected to the printer.

Template

DispRegisterTemplate

Registers a template.

```
INT DispRegisterTemplate(  
    INT i_hdl,  
    INT i_templateID,  
    LPCTSTR i_label,  
    LPCTSTR i_fname )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_templateID

Template ID

Specifies the ID of template to register.

The valid range is 0 to 127.

Do not specify the template ID of 127 because it is being used for the system.

i_label

Template name

Specifies a template name.

The characters that can be specified are ASCII characters 0x20 (space) to 0x7E (tilde) such as alphanumeric characters ('0' to '9', 'A' to 'Z', 'a' to 'z').

Do not use Unicode 0x00A5 ('¥').

The available number of characters is 0 to 32.

i_label is optional. Specify NULL when the template name is not registered.

i_fname

File path

A file specified by *i_fname* is registered as template data to the ID specified by *i_templateID*.

Register template data in XML format.

The maximum data size that can be registered is 8192 bytes.

See "Technical Reference for Display" for details of the template data to register.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Notes

Registered data at the factory may be added or changed without prior notice for quality improvement.

Remarks

- Display is changed to Standby mode when this API is executed. A selecting template is deselected.
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispUnregisterTemplate

Deletes a registered template.

```
INT DispUnregisterTemplate(  
    INT i_hdl,  
    INT i_templateID )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_templateID

Template ID

Specifies the ID of template to delete.

The valid range is 0 to 127.

This API is ignored when a template is not registered in the specified template.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Remarks

- Display is changed to Standby mode when this API is executed. A selecting template is deselected.
- Used memory is not released even the template is deleted. The used memory can be reused after executing **DispDefragment**.
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispSelectTemplate

Selects a template.

```
INT DispSelectTemplate(  
    INT i_hdl,  
    INT i_templateID,  
    INT i_slideID )
```

Parameter

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_templateID

Template ID

Specifies the ID of template to select.

The valid range is 0 to 127.

This API is ignored when a template is not registered in the specified ID.

i_slideID

Slide ID

Specifies the ID of slide data to show as background.

The valid range is -1 to 91.

The background is not shown when -1 is specified.

This API is ignored when slide data is not registered in the specified ID.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Remarks

- The data on the template is cleared when this API is executed.
- The template selected by this API is displayed to the screen when **DispShowTemplate** is executed.
- The template selected by this API is deselected under the following conditions.
 - When the display time is specified at *i_time* of **DispShowTemplate**.
 - When **DispShowSlide**, **DispEnterStandbyMode**, or **DispExecuteMacro** is executed.
 - When this API is newly executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispSetTemplateImageData

Sets image data to show on a selecting template.

```
INT DispSetTemplateImageData(  
    INT i_hdl,  
    INT i_mapID,  
    INT i_imageID )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_mapID

Map ID

Specifies a map ID of img element defined in the selecting template.

The valid range is 0 to 63.

This API is ignored when a specified map ID is not defined in the template.

i_imageID

Image ID

Specifies the ID of image data to show in specified *i_mapID*.

The valid range is 0 to 63.

This API is ignored when image data is not registered in specified ID.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Remarks

- This API setting is cleared under the following conditions.
 - When **DispSelectTemplate** is executed.
 - When other than 0 is specified at *i_time* of **DispShowTemplate** and the specified display time has elapsed.
 - When **DispShowTemplate** registered in **DispExecuteMacro** is executed.
- This API is ignored when no template is selected.
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispSelectTemplateTextObject

Selects the text element to start editing.

```
INT DispSelectTemplateTextObject(  
    INT i_hdl,  
    INT i_mapID )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_mapID

Map ID

Specifies a map ID of text element defined in the selecting template.

The valid range is 0 to 63.

This API is ignored when a specified map ID is not defined in the template.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Remarks

- After specifying the map ID of the selecting template with this API, start to edit characters.
- When a scroll is set at the text element of the specified map ID and this API is executed after **DispShowTemplate**, scroll is executed.
When a scroll is not set at the text element of the specified map ID and this API is executed after **DispShowTemplate**, the input text data is discarded.
- This API setting is cleared under the following conditions.
 - When **DispSelectTemplate** is executed.
 - When **DispShowTemplate** is executed.
- This API is ignored when no template is selected.
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispSetTemplateTextAlignment

Sets alignment to text data.

```
INT DispSetTemplateTextAlignment(  
    INT i_hdl,  
    INT i_halign )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_halign

Alignment

Sets alignment.

See "6.2 Alignment" for specifiable values.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Remarks

- Alignment is valid only as following cases.
 - Text data is not entered in the specified map ID at **DispSelectTemplateTextObject**.
 - Text data entered in the map ID which is specified at **DispSelectTemplateTextObject** is registered right after a line feed.
- Specify the map ID at **DispSelectTemplateTextObject** before executing this API.
- Input text data at **DispSetTemplateTextData** after executing this API.
- The input text data is displayed to the screen when **DispShowTemplate** is executed.
- This API setting is cleared under the following conditions.
 - When **DispSelectTemplate** is executed.
 - When other than 0 is specified with *i_time* of **DispShowTemplate** and the specified display time has elapsed.
 - When **DispShowTemplate** registered in **DispExecuteMacro** is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispSetTemplateTextLeftMargin

Sets left margin to text data.

```
INT DispSetTemplateTextLeftMargin(  
    INT i_hdl,  
    INT i_margin )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_margin

Left margin

Sets left margin in pixel (px).

The valid range is 0 to 479.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Remarks

- The left margin setting is valid only as following cases.
 - Text data is not entered in the specified map ID at **DispSelectTemplateTextObject**.
 - Text data entered in the map ID which is specified at **DispSelectTemplateTextObject** is registered right after a line feed.
- Specify the map ID at **DispSelectTemplateTextObject** before executing this API.
- Input text data at **DispSetTemplateTextData** after executing this API.
- The input text data is displayed to the screen when **DispShowTemplate** is executed.
- This API setting is cleared under the following conditions.
 - When **DispSelectTemplate** is executed.
 - When other than 0 is specified at *i_time* of **DispShowTemplate** and the specified display time has elapsed.
 - When **DispShowTemplate** registered in **DispExecuteMacro** is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispSetTemplateTextBold

Sets bold character to text data.

```
INT DispSetTemplateTextBold(  
    INT i_hdl,  
    BOOL i_bold )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_bold

Bold character specifying/canceling flag

Sets bold character.

TRUE: Specifies bold character.

FALSE: Cancels bold character.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Remarks

- From the text data after this API is executed, the bold characters are applied. The bold character can be set one by one.
- Specify the map ID at **DispSelectTemplateTextObject** before executing this API.
- Input text data at **DispSetTemplateTextData** after executing this API.
- The input text data is displayed to the screen when **DispShowTemplate** is executed.
- This API setting is cleared under the following conditions.
 - When **DispSelectTemplate** is executed.
 - When other than 0 is specified with *i_time* of **DispShowTemplate** and the specified display time has elapsed.
 - When **DispShowTemplate** registered in **DispExecuteMacro** is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispSetTemplateTextUnderline

Sets underline to text data.

```
INT DispSetTemplateTextUnderline(  
    INT i_hdl,  
    BOOL i_underline )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_underline

Underline specifying/canceling flag

Sets underline.

TRUE: Specifies underline.

FALSE: Cancels underline.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Remarks

- From the text data after this API is executed, the underline is applied.
The underline can be set one by one.
- Specify the map ID at **DispSelectTemplateTextObject** before executing this API.
- Input text data at **DispSetTemplateTextData** after executing this API.
- The input text data is displayed to the screen when **DispShowTemplate** is executed.
- This API setting is cleared under the following conditions.
 - When **DispSelectTemplate** is executed.
 - When other than 0 is specified with *i_time* of **DispShowTemplate** and the specified display time has elapsed.
 - When **DispShowTemplate** registered in **DispExecuteMacro** is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispSetTemplateTextSize

Sets character size to text data.

```
INT DispSetTemplateTextSize(  
    INT i_hdl,  
    INT i_hscale,  
    INT i_vscale )
```

Parameters

i_hdl
API handle
Specifies the API handle retrieved by **OpenMonPrinter**.

i_hscale
Horizontal scale
Sets the horizontal character size.
See "6.3 Character Size" for specifiable values.

i_vscale
Vertical scale
Sets the vertical character size.
See "6.3 Character Size" for specifiable values.

Return value

On success: Returns 0.
On failure: Returns an error code.
See "Chapter 5 Error Code List" for details of the error code.

Remarks

- From the text data after this API is executed, the character size is applied.
The character size can be set one by one.
- Specify the map ID at **DispSelectTemplateTextObject** before executing this API.
- Input text data at **DispSetTemplateTextData** after executing this API.
- The input text data is displayed to the screen when **DispShowTemplate** is executed.
- This API setting is cleared under the following conditions.
 - When **DispSelectTemplate** is executed.
 - When other than 0 is specified with *i_time* of **DispShowTemplate** and the specified display time has elapsed.
 - When **DispShowTemplate** registered in **DispExecuteMacro** is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispSetTemplateTextFont

Sets a character font to text data.

```
INT DispSetTemplateTextFont(  
    INT i_hdl,  
    INT i_font )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_font

Character font

Sets a character font.

See "6.4 Character Font" for specifiable values.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Remarks

- From the text data after this API is executed, the character font is applied. The character font can be set one by one.
- Specify the map ID at **DispSelectTemplateTextObject** before executing this API.
- Input text data at **DispSetTemplateTextData** after executing this API.
- The input text data is displayed to the screen when **DispShowTemplate** is executed.
- This API setting is cleared under the following conditions.
 - When **DispSelectTemplate** is executed.
 - When other than 0 is specified with *i_time* of **DispShowTemplate** and the specified display time has elapsed.
 - When **DispShowTemplate** registered in **DispExecuteMacro** is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispSetTemplateTextRegisteredFont

Sets the registered font used for text data.

```
INT DispSetTemplateTextRegisteredFont(  
    INT i_hdl,  
    INT i_font )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_font

Registered font

Sets the registered font.

See "6.5 Registered Font" for specifiable values.

The registered font specifying is ignored when the optional font is not registered.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Remarks

- From the text data after this API is executed, the registered font is applied.
The registered font can be set one by one.
- Input text data at **DispSetTemplateTextData** after executing this API.
- The input text data is displayed to the screen when **DispShowTemplate** is executed.
- This API setting is cleared under the following conditions.
 - When **DispSelectTemplate** is executed.
 - When other than 0 is specified with *i_time* of **DispShowTemplate** and the specified display time has elapsed.
 - When **DispShowTemplate** registered in **DispExecuteMacro** is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispSetCodePage

Sets the codepage used for text data.

```
INT DispSetCodePage(  
    INT i_hdl,  
    INT i_codepage )
```

Parameters

i_hdl
API handle
Specifies the API handle retrieved by **OpenMonPrinter**.

i_codepage
Codepage
Sets the codepage.
See "6.6 Codepage" for specifiable values.

Return value

On success: Returns 0.
On failure: Returns an error code.
See "Chapter 5 Error Code List" for details of the error code.

Remarks

- From the text data after this API is executed, the character code table is applied.
The character code table can be set one by one.
- When the character code table is not set by this API, the following code page is set by the language setting of Windows OS.
 - When a language setting is Japanese: Katakana character set
 - When a language setting is other than Japanese: Codepage 1252
- The characters 0x80 to 0x9F and 0xE0 to 0xFF cannot be displayed when the template encoding specifying is Shift_JIS. Use a template whose encoding specifying is ISO-2022-JP to display.
- Specify the map ID at **DispSelectTemplateTextObject** before executing this API.
- This API setting is cleared under the following conditions.
 - When **DispSelectTemplate** is executed.
 - When other than 0 is specified with *i_time* of **DispShowTemplate** and the specified display time has elapsed.
 - When **DispShowTemplate** registered in **DispExecuteMacro** is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispSetInternationalCharacter

Sets the international character used for text data.

```
INT DispSetInternationalCharacter(  
    INT i_hdl,  
    INT i_intlchar )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_intlchar

International character

Sets the international character.

See "6.7 International Character" for specifiable values.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Remarks

- From the text data after this API is executed, the international character is applied. The international character can be set one by one.
- Specify the map ID at **DispSelectTemplateTextObject** before executing this API.
- This API setting is cleared under the following conditions.
 - When **DispSelectTemplate** is executed.
 - When other than 0 is specified with *i_time* of **DispShowTemplate** and the specified display time has elapsed.
 - When **DispShowTemplate** registered in **DispExecuteMacro** is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispSetTemplateTextRightSpacing

Sets the amount of right space to text data.

```
INT DispSetTemplateTextRightSpacing(  
    INT i_hdl,  
    INT i_spacing )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_spacing

The amount of character right space

Sets the amount of right space in pixel (px).

The valid range is 0 to 255.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Remarks

- From the text data after this API is executed, the amount of character right space is applied. The amount of right space can be set one by one.
- Specify the map ID at **DispSelectTemplateTextObject** before executing this API.
- Input text data at **DispSetTemplateTextData** after executing this API.
- The input text data is displayed to the screen when **DispShowTemplate** is executed.
- This API setting is cleared under the following conditions.
 - When **DispSelectTemplate** is executed.
 - When other than 0 is specified with *i_time* of **DispShowTemplate** and the specified display time has elapsed.
 - When **DispShowTemplate** registered in **DispExecuteMacro** is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispSetTemplateTextColor

Sets the character color used for text data.

```
INT DispSetTemplateTextColor(  
    INT i_hdl,  
    INT i_color )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_color

Character code table

Sets the character color.

See "6.8 Character Color" for specifiable values.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Remarks

- From the text data after this API is executed, the character color is applied.
The character color can be set one by one.
- Specify the map ID at **DispSelectTemplateTextObject** before executing this API.
- Input text data at **DispSetTemplateTextData** after executing this API.
- The input text data is displayed to the screen when **DispShowTemplate** is executed.
- This API setting is cleared under the following conditions.
 - When **DispSelectTemplate** is executed.
 - When other than 0 is specified with *i_time* of **DispShowTemplate** and the specified display time has elapsed.
 - When **DispShowTemplate** registered in **DispExecuteMacro** is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispSetTemplateTextLineSpacing

Sets line spacing to text data.

```
INT DispSetTemplateTextLineSpacing(  
    INT i_hdl,  
    INT i_spacing )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_spacing

Line spacing

Sets line spacing in pixel (px).

The valid range is 0 to 255.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Remarks

- Line spacing setting is valid only as following cases.
 - Text data is not entered in the specified map ID at **DispSelectTemplateTextObject**.
 - Text data entered in the map ID which is specified at **DispSelectTemplateTextObject** is registered right after a line feed.
- When a scroll is set with the text element of the specified map ID, specifying this API is not reflected.
- Specify the map ID at **DispSelectTemplateTextObject** before executing this API.
- Input text data at **DispSetTemplateTextData** after executing this API.
- The input text data is displayed to the screen when **DispShowTemplate** is executed.
- This API setting is cleared under the following conditions.
 - When **DispSelectTemplate** is executed.
 - When other than 0 is specified with *i_time* of **DispShowTemplate** and the specified display time has elapsed.
 - When **DispShowTemplate** registered in **DispExecuteMacro** is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispSetTemplateTextData

Inputs text data to show on the text element specified at **DispSelectTemplateTextObject**.
The method of syntax (a) inputs the text data by multi-byte code or Unicode.
The method of syntax (b) inputs the text data by binary data.

- (a) INT **DispSetTemplateTextData**(
 INT *i_hdl*,
 LPCTSTR *i_text*)
- (b) INT **DispSetTemplateTextData**(
 INT *i_hdl*,
 INT *i_len*,
 LPBYTE *i_data*)

Parameters

- i_hdl*
API handle
Specifies the API handle retrieved by **OpenMonPrinter**.
- i_len*
Data length
Specifies data length of text data.
- i_text (i_data)*
Text data
Inputs text data.
Data size which is able to be specified it once is 1 to 1020 bytes.

Return value

- On success: Returns 0.
- On failure: Returns an error code.
See "Chapter 5 Error Code List" for details of the error code.

Remarks

- This API encodes specified text data into text data which is enable to show on Display on the basis of settings at **DispSetInternationalCharacter** or **DispSetCodePage**, and displays. Text that could not be encoded is shown as '?'.
• Select the template in **DispSelectTemplate** before executing this API.
- After specifying the map ID of the selecting template with **DispSelectTemplateTextObject**, input text data with this API. This API is ignored when the map ID is not specified in **DispSelectTemplateTextObject**.
- The input text data is displayed to the screen when **DispShowTemplate** is executed.
- The input text data is cleared under the following conditions.
 - When **DispSelectTemplate** is executed.
 - When other than 0 is specified with *i_time* of **DispShowTemplate** and the specified display time has elapsed.
 - When **DispShowTemplate** registered in **DispExecuteMacro** is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispSetTemplateBarcodeData

Specifies a map ID of the barcode element on a selecting template, and inputs barcode data.

INT **DispSetTemplateBarcodeData**(

INT *i_hdl*,

INT *i_mapID*,

INT *i_len*,

LPBYTE *i_data*)

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_mapID

Map ID

Specifies map ID of a barcode element defined in a selecting template.

The valid range is 0 to 7.

This API is ignored when the specified map ID is not defined in the selecting template.

i_len

Data length

Specifies the data length of barcode data.

i_data

Barcode data

Inputs barcode data.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Remarks

- See the Display command "Input Barcode Data" described in "Technical Reference for Display" for details of data length and specification of barcode data.
- The input barcode data is displayed to the screen when **DispShowTemplate** is executed.
- The input barcode data is cleared under the following conditions.
 - When **DispSelectTemplate** is executed.
 - When other than 0 is specified with *i_time* of **DispShowTemplate** and the specified display time has elapsed.
 - When **DispShowTemplate** is executed.
- This API is ignored when no template is selected.
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispSetTemplateQrCodeData

Specifies a map ID of the qr element on a selecting template, and inputs QR Code data.

```
INT DispSetTemplateQrCodeData(  
    INT i_hdl,  
    INT i_mapID,  
    INT i_size,  
    INT i_errcrct,  
    INT i_mode,  
    BOOL i_quiet,  
    INT i_len,  
    LPBYTE i_data )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_mapID

Map ID

Specifies map ID of a qr element defined in a selecting template.

The valid range is 0 to 7.

This API is ignored when the specified map ID is not defined in the selecting template.

i_size

Module size

Specifies module size.

See "6.9 Module Size" for specifiable values.

i_errcrct

Error correction level

Specifies error correction level.

See "6.10 Error Correction Level" for specifiable values.

This parameter is ignored when 0 is specified at *i_size*.

i_mode

Data mode

Specifies the data mode.

See "6.11 Data Mode" for specifiable values.

This parameter is ignored when 0 is specified at *i_size*.

i_quiet

With/without quiet zone

Specifies whether to show with or without quiet zone.

TRUE: With quiet zone

FALSE: Without quiet zone

This parameter is ignored when 0 is specified at *i_size*.

i_len

Data length

Specifies the data length of QR Code data.

The valid range is 1 to 3909.

i_data

QR Code data

Inputs QR Code data.

Input data according to the data input format of QR Code for the input of QR Code data.

Return value

- On success: Returns 0.
- On failure: Returns an error code.
See "Chapter 5 Error Code List" for details of the error code.

Remarks

- The input QR Code data is displayed to the screen when **DispShowTemplate** is executed.
- The input QR Code data is cleared under the following conditions.
 - When **DispSelectTemplate** is executed.
 - When other than 0 is specified with *i_time* of **DispShowTemplate** and the specified display time has elapsed.
 - When **DispShowTemplate** is executed.
- This API is ignored when no template is selected.
- In use via a printer, this API is ignored when Display is not connected to the printer.

Image Registration

DispRegisterImageData

Registers image data.

```
INT DispRegisterImageData(  
    INT i_hdl,  
    INT i_imageID,  
    LPCTSTR i_label,  
    LPCTSTR i_fname )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_imageID

Image ID

Specifies the ID of image data to register.

The valid range is 0 to 63.

Do not specify the image IDs of 49 to 63 because they are being used for the system.

i_label

Label name

Specifies the label name.

The characters that can be specified are ASCII characters 0x20 (space) to 0x7E (tilde) such as alphanumeric characters ('0' to '9', 'A' to 'Z', 'a' to 'z').

Do not use Unicode 0x00A5 ('¥').

The available number of characters is 0 to 32.

i_label is optional. Specify NULL when the label name is not registered.

The specified label name can be retrieved by **DispGetObjectLabel**.

i_fname

File path

Registers image data to the ID specified at *i_imageID*.

Specifies the file name of image data created in the prescribed format.

Supported file extensions are .jpg, .jpeg and .png. However, even the supported extensions may not be registered depending on the format.

The maximum data size that can be registered is 480 horizontal × 272 vertical pixels (px).

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Notes

Registered data at the factory may be added or changed without prior notice for quality improvement.

Remarks

- Display is changed to Standby mode when this API is executed. A selecting template is deselected.
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispUnregisterImageData

Deletes registered image data.

```
INT DispUnregisterImageData(  
    INT i_hdl,  
    INT i_imageID )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_imageID

Image ID

Specifies the ID of image data to delete.

The valid range is 0 to 63.

This API is ignored when image data is not registered in the specified ID.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Remarks

- Display is changed to Standby mode when this API is executed. A selecting template is deselected.
- Used memory is not released even image data is deleted. The used memory can be reused after executing **DispDefragment**.
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispRegisterSlideData

Registers slide data.

```
INT DispRegisterSlideData(  
    INT i_hdl,  
    INT i_slideID  
    LPCTSTR i_label  
    LPCTSTR i_fname )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_slideID

Slide ID

Specifies the ID of slide data to register.

The valid range is 0 to 91.

Do not specify the slide IDs of 80 to 90 because they are being used for the system.

i_label

Label name

Specifies the label name.

The characters that can be specified are ASCII characters 0x20 (space) to 0x7E (tilde) such as alphanumeric characters ('0' to '9', 'A' to 'Z', 'a' to 'z').

Do not use Unicode 0x00A5 ('¥').

The available number of characters is 0 to 32.

i_label is optional. Specify NULL when the label name is not registered.

The specified label name can be retrieved by **DispGetObjectLabel**.

i_fname

File path

Registers slide data to the ID specified with *i_slideID*.

Specifies the file name of slide data created in the prescribed format.

Supported file extensions are .jpg, .jpeg and .png. However, even the supported extensions may not be registered depending on the format.

The data size that can be registered is fixed to 480 horizontal × 272 vertical pixels (px).

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Notes

Registered data at the factory may be added or changed without prior notice for quality improvement.

Remarks

- Display is changed to Standby mode when this API is executed. A selecting template is deselected.
- Execute **DispShowSlide** to show the registered slide data.
- Specify the slide ID at **DispShowTemplate** to use the registered slide data as a backscreen of the template.
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispUnregisterSlideData

Deletes registered slide data.

```
INT DispUnregisterSlideData(  
    INT i_hdl,  
    INT i_slideID )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_slideID

Slide ID

Specifies the ID of slide data to delete.

The valid range is 0 to 91.

This API is ignored when slide data is not registered in the specified ID.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Remarks

- Display is changed to Standby mode when this API is executed. A selecting template is deselected.
- In use via a printer, this API is ignored when Display is not connected to the printer.

Character Registration

DispRegisterUserDefinedCharacter

Registers user-defined characters.

```
INT DispRegisterUserDefinedCharacter(  
    INT i_hdl,  
    LPCTSTR i_fname )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_fname

File path

Specifies the file name of user-defined character data created in the prescribed format.

Supported file extension is .bin.

See the Display command "Register User-defined Character" described in "Technical Reference of Display" for user-defined character data.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Remarks

- Display is changed to Standby mode when this API is executed. A selecting template is deselected.
- User-defined characters are overwritten when this API is executed with user-defined character registered status.
- Use the template which encoding specifying is Shift_JIS for displaying user-defined characters.
- Specify 1 for *i_codepage* of **DispSetCodePage** before user-defined characters are displayed.
- Specify the character codes that can be specified for *i_text* of **DispSetTemplateTextData** or *i_data* of **DispSetTemplateTextBData** when user-defined characters are displayed. The character codes that can be specified are as follows.
 - For MBCS (MultiByte Character Set): 0xF040 to 0xF07E and 0xF080 to 0xF09E
 - For UNICODE (Unicode) : 0xE000 to 0xE05D
 - For binary data : 0xEC40 to 0xEC7E and 0xEC80 to 0xEC9E
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispUnregisterUserDefinedCharacter

Deletes registered user-defined characters.

```
INT DispUnregisterUserDefinedCharacter(  
    INT i_hdl )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Remarks

- Display is changed to Standby mode when this API is executed. A selecting template is deselected.
- All registered user-defined characters are deleted.
- Used memory is not released even the user-defined character is deleted. The used memory can be reused after executing **DispDefragment**.
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispRegisterOptionFont

Registers optional fonts.

```
INT DispRegisterOptionFont(  
    INT i_hdl,  
    INT i_startcode,  
    INT i_endcode,  
    INT i_width,  
    INT i_height,  
    LPCTSTR i_fname )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_startcode

Character code for registration starting

Specifies the character code to start registration.

The valid range is 0x20 to 0xFF in ASCII code.

i_endcode

Character code for registration ending
Specifies the character code to end registration.
The valid range is 0x20 to 0xFF in ASCII code.

i_width

Character width
Specifies the character width in pixel (px).

i_height

Character height
Specifies the character height in pixel (px).

i_fname

File path
Specifies the file name of optional font data created in the prescribed format.
Supported file extension is .bin.
See the Display command "Register Optional Font" described in "Technical Reference of Display" for optional font data.

Return value

On success: Returns 0.
On failure: Returns an error code.
See "Chapter 5 Error Code List" for details of the error code.

Remarks

- Display is changed to Standby mode when this API is executed. A selecting template is deselected.
- When this API is executed with optional font registered status, the registered optional fonts are deleted, and a new memory area is allocated to register optional fonts. Used memory is not released even the registered optional fonts are deleted. The used memory can be reused after executing **DispDefragment**.
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispUnregisterOptionFont

Deletes registered optional fonts.

```
INT DispUnregisterOptionFont(  
    INT i_hdl )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Remarks

- Display is changed to Standby mode when this API is executed. A selecting template is deselected.
- All registered optional fonts are deleted.
- Used memory is not released even the optional font is deleted. The used memory can be reused after executing **DispDefragment**.
- In use via a printer, this API is ignored when Display is not connected to the printer.

Macro

DispControlMacroRegistration

Specifies start and end of macro registration.

```
INT DispControlMacroRegistration(  
    INT i_hdl,  
    INT i_macroID,  
    INT i_mode )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_macroID

Macro ID

Specifies the ID of macro to register.

The valid range is -1 to 127.

Do not specify the macro IDs of 120 to 126 because they are being used for the system.

Macro ID	Contents
120	During Bluetooth pairing
121	Cutter error
122	Vp voltage error
123	Head temperature error
124	Hardware error
125	Cover open error
126	Out-of-paper error
127	Standby (Initial value)

i_mode

Macro processing selection

Selects macro processing.

See "6.12 Macro Processing Selection" for specifiable values.

The procedure of macro processing selection is as follows.

Starts macro registration:

Starts macro registration when 0 is specified to *i_mode*.

Specify -1 to *i_macroID*.

The APIs which are registration targets and their data are buffered in the internal memory of the SDK after registration is started.

The maximum data size that can register macro is 1024 bytes.

The APIs which are not registration targets, and their data are executed immediately without registration.

The APIs which are registration targets are as follows.

- **DispSelectTemplate**
- **DispSetTemplateImageData**
- **DispSelectTemplateTextObject**
- **DispSetTemplateTextData**
- **DispSetTemplateTextAlignment**
- **DispSetTemplateTextLeftMargin**
- **DispSetTemplateTextBold**
- **DispSetTemplateTextUnderline**
- **DispSetTemplateTextSize**
- **DispSetTemplateTextFont**
- **DispSetRegisteredFont**
- **DispSetCodePage**
- **DispSetInternationalCharacter**
- **DispSetTemplateTextRightSpacing**
- **DispSetTemplateTextColor**
- **DispSetTemplateTextLineSpacing**
- **DispSetTemplateBarcodeData**
- **DispSetTemplateQrCodeData**
- **DispShowTemplate**
- **DispShowSlide**

Ends after registering macro:

When 1 is specified to *i_mode*, specifies a macro to specified macro ID and ends this API.

Specify the macro ID to register to *i_macroID*.

Display is changed to Standby mode when this API is executed. A selecting template is deselected.

Ends without registering macro:

When 2 is specified to *i_mode*, discards the APIs which are registration targets, and their data that has been buffered, and ends this API. A selecting template is deselected.

Deletes the registered macro:

Specify 0 to *i_mode* and specify -1 to *i_macroID*, and call this API.

Specify 1 to *i_mode* and specify the macro ID to delete to *i_macroID*, and call this API.

Display is changed to Standby mode when this API is executed. A selecting template is deselected.

The registered macro can be executed in **DispExecuteMacro**.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Notes

Registered data at the factory may be added or changed without prior notice for quality improvement.

Remarks

- In use via a printer, this API is ignored when Display is not connected to the printer.

Memory Operation

DispDefragment

Optimizes memory area.

```
INT DispDefragment(  
    INT i_hdl  
    INT i_area )
```

Parameters

i_hdl
API handle
Specifies the API handle retrieved by **OpenMonPrinter**.

i_area
Memory area
Specify 1. Specifies user area.

Return value

On success: Returns 0.
On failure: Returns an error code.
See "Chapter 5 Error Code List" for details of the error code.

Remarks

- Display is changed to Standby mode when this API is executed. A selecting template is deselected.
- Do not turn off the device during optimization.
- Optimization may take several minutes.
- In use via a printer, this API is ignored when Display is not connected to the printer.

DispGetMemory

Gets the remaining capacity of memory area.

```
INT DispGetMemory(  
    INT i_hdl  
    INT i_area  
    LPDWORD o_size )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_area

Memory area

Specify 1. Specifies user area.

o_size

Acquisition data size

Specifies the buffer size for storing the response of remaining capacity of memory area.

The data size of the remaining capacity of the acquired memory area is stored when control is returned from the API.

This API fails when the specified buffer size is smaller than the data size of the remaining capacity of the acquired memory area, and the data size of the remaining capacity of the specified memory area is stored.

Return value

On success: Returns the available remaining capacity excluding the released area.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

DispInitializeMemoryArea

Initializes memory area.

```
INT DispInitializeMemoryArea(  
    INT i_hdl  
    INT i_area )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_area

Memory area

Specify 1. Specifies memory area.

Return value

- On success: Returns 0.
- On failure: Returns an error code.
See "Chapter 5 Error Code List" for details of the error code.

Notes

The data registered by following APIs are deleted when 1 is specified to *i_area* and the memory area is initialized.

- **DispRegisterTemplate**
- **DispRegisterImageData**
- **DispControlMacroRegistration**
- **DispRegisterUserDefinedCharacter**
- **DispRegisterOptionFont**

Part of data which is registered at the factory to use for the system is deleted either. Therefore, when Display is used via a printer, the guide screen cannot be displayed if an error occurs in the printer.

When Display is used alone, the guide screen is not displayed.

The used memory area can be reused after executing **DispDefragment**.

Remarks

- Display is changed to Standby mode when this API is executed. A selecting template is deselected.
- When user area is initialized, the registered templates, image data, user-defined fonts, optional fonts and macro functions are all cleared, and the user area will be in the initial state.
- Do not turn off the device during initialization.
- Initialization may take several minutes.
- In use via a printer, this API is ignored when Display is not connected to the printer.

Response Function

DispGetRegisteredStateList

Gets the presence or absence of contents registration.

```
INT DispGetRegisteredStateList(  
    INT i_hdl  
    INT i_type  
    LPINT io_len  
    LPBYTE o_list )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_type

Contents type

Specifies the contents type to get the response.

See "6.1 Contents Type" for specifiable values.

io_len

Data length

Specifies the maximum length of the data to retrieve.

The retrieved data size is stored when control is returned from the API.

o_list

Buffer that stores the presence or absence of contents registration

Specifies the buffer that stores the presence or absence of contents registration.

The registration status of the contents type specified by *i_type* is stored in the buffer sequentially from the ID 0.

1 is stored when the data is registered, and 0 is stored when the data is not registered.

The buffer size to be specified depends on the contents type.

Contents Type	Buffer Size to be Specified
Template	128
Slide	24
Image	64

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

DispGetObjectLabel

Gets the label names of registered contents.

```
INT DispGetObjectLabel(  
    INT i_hdl,  
    INT i_type,  
    INT i_objectID,  
    LPINT io_len,  
    LPSTR o_data )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_type

Contents type

Specifies the contents type to get the response.

See "6.1 Contents Type" for specifiable values.

i_objectID

ID of contents

For template:

Specify *i_templateID* of the template registered in **DispRegisterTemplate**.

For slide data:

Specify *i_slideID* of slide data registered in **DispRegisterSlideData**.

For image data:

Specify *i_imageID* of image data registered in **DispRegisterImageData**.

io_len

Data length

Specifies the maximum length of the data to retrieve.

The retrieved data size is stored when control is returned from the API.

o_data

Received data

Returns NULL when the contents is not registered in the ID specified by *i_objectID*.

Return value

On success: Returns 0.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

DispGetResponseRequest

Gets the specified response codes when this API is executed.

```
INT DispGetResponseRequest(  
    INT i_hdl  
    INT i_id  
    INT i_timeout )
```

Parameters

i_hdl

API handle

Specifies the API handle retrieved by **OpenMonPrinter**.

i_id

Response code

Specifies the response code to be returned by execution of this API.

The valid range is 0 to 255.

However, the upper 4 bits are ignored.

i_timeout

Timeout period

Specifies the time to wait for success of this API in milliseconds (ms).

The valid range is from 3000 to 90000.

When the value is specified less than 3000, the time is set to 3000 ms.

When the value is specified more than 90000, the time is set to 90000 ms.

Return value

On success: Returns the response code specified with *i_id*.

The response format is a code from 0x40 to 0x4F, which is the logical sum of the lower 4 bits of specified with *i_id* and 0x40.

On failure: Returns an error code.

See "Chapter 5 Error Code List" for details of the error code.

Remarks

- By executing this API at the end of other APIs or transmission data and returning the response code specified by *i_id*, the completion of data processing execution can be confirmed from the outside.

Chapter 4 .NET API

This chapter describes the .NET API.

4.1 Development Language

The following development languages are covered.

- Visual Basic .NET
- Visual C#

4.2 Library File

The library file of .NET API is a class library format.

The library file has the following file name.

- SPSWL_Cls.dll

The library file is stored in the Global Assembly Cache (GAC) folder.

4.3 API List

The following APIs are implemented in the .NET API.

- Namespace: SII.SDK.PrinterDevice
- Class name: DisplayAPI

4.3.1 Common API

Category	API	Function Summary
Property	LastError	Gets the error value of the latest executed API.
Property	IsValid	Gets the call status of OpenMonPrinter .
Method	OpenMonPrinter	Starts using the .NET API.
Method	CloseMonPrinter	Ends using the .NET API.
Method	LockPrinter	Prohibits (Locks) all data transmission and hardware reset requests from other processes to the device.
Method	UnlockPrinter	Unlocks access prohibition (lock) from other processes by LockPrinter .
Method	DirectIOEx	Transmits and receives binary data.

4.3.2 Specific API

Category	API	Function Summary
Method	ShowTemplate	Displays the template on the screen.
Method	ShowSlide	Displays the slide on the screen.
Method	TurnOnScreen	Sets the screen backlight on/off.
Method	EnterStandbyMode	Changes Display to Standby mode.
Method	ExecuteMacro	Executes the registered macro.
Method	RegisterTemplate	Registers a template.
Method	UnregisterTemplate	Deletes a registered template.
Method	SelectTemplate	Selects a template.
Method	SetTemplateImageData	Sets image data to show on a selecting template.
Method	SelectTemplateTextObject	Selects the text element to start editing.
APIs that are valid for the text element selected a template with DispSelectTemplate and/or specified with DispSelectTemplateTextObject .		
Method	SetTemplateTextAlignment	Sets alignment to text data.
Method	SetTemplateTextLeftMargin	Sets left margin to text data.
Method	SetTemplateTextBold	Sets bold character to text data.
Method	SetTemplateTextUnderline	Sets underline to text data.
Method	SetTemplateTextSize	Sets character size to text data.
Method	SetTemplateTextFont	Sets a character font to text data.
Method	SetTemplateTextRegisteredFont	Sets the registered font used for text data.
Method	SetCodePage	Sets the codepage used for text data.

Category	API	Function Summary
Method	SetInternationalCharacter	Sets the international character used for text data.
Method	SetTemplateTextRightSpacing	Sets the amount of right space to text data.
Method	SetTemplateTextColor	Sets the character color used for text data.
Method	SetTemplateTextLineSpacing	Sets line spacing to text data.
Method	SetTemplateTextData ^{*1} SetTemplateTextBDData ^{*2}	Inputs text data to show on the text element specified by DispSelectTemplateTextObject .
Method	SetTemplateBarcodeData	Specifies a map ID of the barcode element on a selecting template, and inputs barcode data.
Method	SetTemplateQrCodeData	Specifies a map ID of the barcode element on a selecting template, and inputs QR Code data.
Method	RegisterImageData	Registers image data.
Method	UnregisterImageData	Deletes registered image data.
Method	RegisterSlideData	Registers slide data.
Method	UnregisterSlideData	Deletes registered slide data.
Method	RegisterUserDefinedCharacter	Registers user-defined characters.
Method	UnregisterUserDefinedCharacter	Deletes registered user-defined characters.
Method	RegisterOptionFont	Registers optional fonts.
Method	UnregisterOptionFont	Deletes registered optional fonts.
Method	ControlMacroRegistration	Specifies start and end of macro registration.
Method	Defragment	Optimizes memory area.
Method	GetMemory	Gets the remaining capacity of memory area.
Method	InitializeMemoryArea	Initializes memory area.
Method	GetRegisteredStateList	Gets the presence or absence of contents registration.
Method	GetObjectLabel	Gets the label names of registered contents.
Method	GetResponseRequest	Gets the specified response codes when this API is executed.

*1: Specifies arguments by strings.

*2: Specifies arguments by binary data.

Reference

- In order to use the Display API, template registration is required to be specified in XML format, and it is necessary to refer to "Technical Reference for Display" for details on the necessary elements.
It is also recommended that read well "Technical Reference for Display" regarding how to use slides, templates, and macros, and state transitions in Display status.

4.4 API Details

4.4.1 Property

Property

LastError

Gets the error value of the latest called API.

```
ErrorCode LastError { get; }
```

Initial value

SUCCESS

Remarks

- See "Chapter 5 Error Code List" for details of the error code.

IsValid

Gets the call status of **OpenMonPrinter**.

```
bool IsValid { get; }
```

Initial value

FALSE

Remarks

- TRUE: **OpenMonPrinter** is successful.
- FALSE: **OpenMonPrinter** is not successful.

Method

OpenMonPrinter

Starts using the .NET API.

```
ErrorCode OpenMonPrinter(  
    OpenType type,  
    string name )
```

Parameters

type

Open type

OpenType.TYPE_DISPLAY (fixed)

Specify OpenType.TYPE_DISPLAY in both cases when Display is used via a printer and when Display is used alone.

name

Name of the device that uses the .NET API

Specifies the device name (friendly name).

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- This API can be simultaneously opened up to 8 instances of each class of .NET API in 1 process.
- When the .NET API is no longer used, be sure to call **CloseMonPrinter**.
- Set the driver to the following connection destination.
When Display is used via a printer:
Specify the printer port.
When Display is used alone and the printer driver is used:
Specify the USB.
- This API succeeds even when the device is not connected or the device power is turned off.

CloseMonPrinter

Ends using the .NET API.

ErrorCode **CloseMonPrinter**()

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- All settings and data that have been associated with the API are discarded by this API.

LockPrinter

Prohibits (Locks) all data transmission and hardware reset requests from other processes to the device.

ErrorCode **LockPrinter**(
int *timeout*)

Parameters

timeout

Timeout period

Specifies the time to wait for success of this API in milliseconds (ms).

The valid range is from 3000 to 90000.

When the value is specified less than 3000, the time is set to 3000 ms.

When the value is specified more than 90000, the time is set to 90000 ms.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- By this API, a multiple locking is possible up to 99 times. In order to release the lock, call **UnlockPrinter** as the same times as this API was called.

UnlockPrinter

Unlocks the access prohibition (lock) from other processes by **LockPrinter**.

ErrorCode **UnlockPrinter**()

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- When **LockPrinter** is called multiple times, this API must be called as the same times as **LockPrinter** is called to release the lock.

DirectIOEx

With the syntax (a), the API gets the receive data as binary data from the device after transmitting binary data.

With the syntax (b), the API gets the receive data as string data from the device after transmitting binary data.

With the syntax (c), the API transmits binary data.

(a) ErrorCode **DirectIOEx**(
 byte[] *cmd*,
 ref byte[] *data*,
 int *timeout*,
 bool *readFlag*,
 byte *option*)

(b) ErrorCode **DirectIOEx**(
 byte[] *cmd*,
 out string *data*,
 int *timeout*,
 byte *option*)

(c) ErrorCode **DirectIOEx**(
 byte[] *cmd*,
 int *timeout*)

Parameters

cmd

Buffer of data to transmit
Specifies the buffer in which the data to transmit is stored.

data

Buffer of data to receive
Specifies the buffer that stores the data to get.
The maximum receive data size is 4096 bytes.
When the value is specified more than 4096, the size is set to 4096 bytes.
Specify 0 when data retrieving is not necessary.
The received data size is stored when the control is returned from the API.

timeout

Timeout period

Specifies the time to wait for success of this API in milliseconds (ms).

The valid range is from 3000 to 90000.

When the value is specified less than 3000, the time is set to 3000 ms.

When the value is specified more than 90000, the time is set to 90000 ms.

readFlag

Receive operation flag

Specifies the receive operation.

TRUE: Continues receiving until any data is received or timeout occurs.

FALSE: Continues receiving until the receive data size is received or timeout occurs.

option

Receive target option

Specifies the data to receive from the following.

When Display is used via a printer:

0: Gets the data excluding the response of the ASB setting command.

1: Gets the data including the response of the ASB setting command.

When Display is used alone:

Specify 0.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- When this API is succeeded, the parameter (A) ref byte[] data is resized to the received data with the limits of the specified size before being called.
- When continuous data of "0x10, 0x32" is included in the receive data with the syntax (B), "0x10, 0x32" is converted to "0x5f, 0x5f".

4.4.2 Specific API

Screen Display

ShowTemplate

Displays the template on the screen.

ErrorCode **ShowTemplate**(
int *time_ms*)

Parameters

time_ms

Display time

Specifies display time on the screen in milliseconds (ms).

The valid range is 0 to 25500.

When the value exceeds 0 and less than 100 is specified, the time is set to 100 ms.

When the value exceeding 25500 is specified, the time is set to 25500 ms.

When the macro is registered by **ControlMacroRegistration**:

When 0 is specified with *time_ms*, a template is shown continuously.

When other than 0 is specified with *time_ms*, a next template is shown after the display time is elapsed.

When the macro is not registered by **ControlMacroRegistration**:

When 0 is specified with *time_ms*, a template is shown continuously.

When other than 0 is specified with *time_ms*, the display time returns to a previous template after the display time is elapsed.

In case of the previous screen has been updated with the display time other than 0, the screen is traced back to the template which was updated with the display time 0.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- Updates the screen, and displays data being specified with the following APIs.

- **SetTemplateImageData**
- **SelectTemplateTextObject**
- **SetTemplateTextAlignment**
- **SetTemplateTextLeftMargin**
- **SetTemplateTextBold**
- **SetTemplateTextUnderline**
- **SetTemplateTextSize**
- **SetTemplateTextFont**
- **SetRegisteredFont**
- **SetCodePage**
- **SetInternationalCharacter**
- **SetTemplateTextRightSpacing**
- **SetTemplateTextColor**
- **SetTemplateTextLineSpacing**

- **SetTemplateTextData**
- **SetTemplateBarcodeData**
- **SetTemplateQrCodeData**

- In use via a printer, this API is ignored when Display is not connected to the printer.

ShowSlide

Displays the slide on the screen.

```
ErrorCode ShowSlide(
    int slideID,
    int time_ms )
```

Parameters

slideID

Slide ID

Specifies the ID of slide data which was registered in **RegisterSlideData**.

The valid range is 0 to 91.

This API is ignored when slide data is not registered in the specified ID.

time_ms

Display time

Specifies display time on the screen in milliseconds (ms).

The valid range is 0 to 25500.

When the value exceeds 0 and less than 100 is specified, the time is set to 100 ms.

When the value exceeding 25500 is specified, the time is set to 25500 ms.

When the macro is registered by **ControlMacroRegistration**:

When 0 is specified with *time_ms*, a slide is shown continuously.

When other than 0 is specified with *time_ms*, a next slide is shown after the display time is elapsed.

When the macro is not registered by **ControlMacroRegistration**:

When 0 is specified with *time_ms*, a slide is shown continuously.

When other than 0 is specified with *time_ms*, the display time returns to a previous slide after the display time is elapsed.

In case of the previous screen has been updated with the display time other than 0, the screen is traced back to the slide which was updated with the display time 0.

Remarks

- A selecting template is deselected when this API is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

TurnOnScreen

Sets the screen backlight on/off.

ErrorCode **TurnOnScreen**(
 bool *turnon*)

Parameters

turnon

Operation flag

Specifies the operation.

TRUE: Turn on the screen backlight.

FALSE: Turn off the screen backlight.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- In use via a printer, this API is ignored when Display is not connected to the printer.

EnterStandbyMode

Changes Display to Standby mode.

ErrorCode **EnterStandbyMode**()

Remarks

- This API is ignored during Standby mode or Guide mode is being displayed.
- A selecting template is deselected when this API is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

ExecuteMacro

Executes the registered macro.

ErrorCode **ExecuteMacro**(
 int *macroID*,
 int *repeatCount*)

Parameters

macroID

Macro ID

Specifies the ID of macro to execute.

The valid range is 0 to 127.

This API is ignored when the macro is not registered in the specified ID.

repeatCount

The number of execution times

Specifies the number of times to execute macro.

The valid range is 0 to 255.

Continue the repeating when 0 is specified.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- A selecting template is deselected.
- In use via a printer, this API is ignored when Display is not connected to the printer.

Template

RegisterTemplate

Registers a template.

```
ErrorCode RegisterTemplate(  
    int templateID,  
    string label,  
    string fileName )
```

Parameters

templateID

Template ID

Selects an ID of template to register.

The valid range is 0 to 127.

Do not specify the template ID of 127 because it is being used for the system.

label

Template name

Specifies a template name.

The characters that can be specified are ASCII characters 0x20 (space) to 0x7E (tilde) such as alphanumeric characters ('0' to '9', 'A' to 'Z', 'a' to 'z').

Do not use Unicode 0x00A5 (¥).

The available number of characters is 0 to 32.

label is optional. Specify an empty string when the template name is not registered.

fileName

File path

A file specified by *fileName* is registered as template data to the ID specified by *templateID*.

Register template data in XML format.

The maximum data size that can be registered is 8192 bytes.

See "Technical Reference for Display" for details of the template data to register.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Notes

Registered data at the factory may be added or changed without prior notice for quality improvement.

Remarks

- Display is changed to Standby mode when this API is executed. A selecting template is deselected.
- In use via a printer, this API is ignored when Display is not connected to the printer.

UnregisterTemplate

Deletes a registered template.

ErrorCode **UnregisterTemplate**(
int *templateID*)

Parameters

templateID

Template ID

Specifies the ID of template to delete.

The valid range is 0 to 127.

This API is ignored when a template is not registered in the specified template.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- Display is changed to Standby mode when this API is executed. A selecting template is deselected.
- Used memory is not released even the template is deleted. The used memory can be reused after executing **Defragment**.
- In use via a printer, this API is ignored when Display is not connected to the printer.

SelectTemplate

Selects a template.

ErrorCode **SelectTemplate**(
int *templateID*,
int *slideID*)

Parameter

templateID

Template ID

Specifies the ID of template to select.

The valid range is 0 to 127.

This API is ignored when a template is not registered in the specified ID.

slideID

Slide ID

Specifies the ID of slide data to show as background.

The valid range is -1 to 91.

The background is not shown when -1 is specified.

This API is ignored when slide data is not registered in the specified ID.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- The data on the template is cleared when this API is executed.
- The template selected by this API is displayed to the screen when **ShowTemplate** is executed.
- The template selected by this API is deselected under the following conditions.
 - When the display time is specified with *time_ms* of **ShowTemplate**.
 - When **ShowSlide**, **EnterStandbyMode**, or **ExecuteMacro** is executed.
 - When this API is newly executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

SetTemplateImageData

Sets image data to show on a selecting template.

```
ErrorCode SetTemplateImageData(  
    int mapID,  
    int imageID )
```

Parameters

mapID

Map ID

Specifies a map ID of img element defined in the selecting template.
The valid range is 0 to 63.

This API is ignored when a specified map ID is not defined in the template.

imageID

Image ID

Specifies the ID of image data to show in specified *mapID*.
The valid range is 0 to 63.

This API is ignored when image data is not registered in specified ID.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- This API setting is cleared under the following conditions.
 - When **SelectTemplate** is executed.
 - When other than 0 is specified at *time_ms* of **ShowTemplate** and the specified display time has elapsed.
 - When **ShowTemplate** registered in **ExecuteMacro** is executed.
- This API is ignored when no template is selected.
- In use via a printer, this API is ignored when Display is not connected to the printer.

SelectTemplateTextObject

Selects the text element to start editing.

ErrorCode **SelectTemplateTextObject**(
int *mapID*)

Parameters

mapID

Map ID

Specifies a map ID of text element defined in the selecting template.
The valid range is 0 to 63.

This API is ignored when a specified map ID is not defined in the template.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- After specifying the map ID of the selecting template with this API, start to edit characters.
- When a scroll is set at the text element of the specified map ID and this API is executed after **ShowTemplate**, scroll is executed.
When a scroll is not set at the text element of the specified map ID and this API is executed after **ShowTemplate**, the input text data is discarded.
- This API setting is cleared under the following conditions.
 - When **SelectTemplate** is executed.
 - When **ShowTemplate** is executed.
- This API is ignored when no template is selected.
- In use via a printer, this API is ignored when Display is not connected to the printer.

SetTemplateTextAlignment

Sets alignment to text data.

ErrorCode **SetTemplateTextAlignment**(
PrintAlignment *alignment*)

Parameters

alignment

Alignment

Sets alignment.

See "6.2 Alignment" for specifiable values.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- Alignment is valid only as following cases.
 - Text data is not entered in the specified map ID at **SelectTemplateTextObject**.
 - Text data entered in the map ID which is specified at **SelectTemplateTextObject** is registered right after a line feed.
- Specify the map ID at **SelectTemplateTextObject** before executing this API.
- Input text data at **SetTemplateTextData** after executing this API.
- The input text data is displayed to the screen when **ShowTemplate** is executed.
- This API setting is cleared under the following conditions.
 - When **SelectTemplate** is executed.
 - When other than 0 is specified with *time_ms* of **ShowTemplate** and the specified display time has elapsed.
 - When **ShowTemplate** registered in **ExecuteMacro** is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

SetTemplateTextLeftMargin

Sets left margin to text data.

ErrorCode **SetTemplateTextLeftMargin**(
int *margin*)

Parameters

margin

Left margin

Sets left margin in pixel (px).
The valid range is 0 to 479.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- The left margin setting is valid only as following cases.
 - Text data is not entered in the specified map ID at **SelectTemplateTextObject**.
 - Text data entered in the map ID which is specified at **SelectTemplateTextObject** is registered right after a line feed.
- Specify the map ID at **SelectTemplateTextObject** before executing this API.
- Input text data at **SetTemplateTextData** after executing this API.
- The input text data is displayed to the screen when **ShowTemplate** is executed.

- This API setting is cleared under the following conditions.
 - When **SelectTemplate** is executed.
 - When other than 0 is specified with *time_ms* of **ShowTemplate** and the specified display time has elapsed.
 - When **ShowTemplate** registered in **ExecuteMacro** is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

SetTemplateTextBold

Sets bold character to text data.

ErrorCode **SetTemplateTextBold**(
 bool *bold*)

Parameters

bold

Bold character specifying/canceling flag

Sets bold character.

TRUE: Specifies bold character.

FALSE: Cancels bold character.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- From the text data after this API is executed, the bold character is applied.
 The bold character can be set one by one.
- Specify the map ID at **SelectTemplateTextObject** before executing this API.
- Input text data at **SetTemplateTextData** after executing this API.
- The input text data is displayed to the screen when **ShowTemplate** is executed.
- This API setting is cleared under the following conditions.
 - When **SelectTemplate** is executed.
 - When other than 0 is specified with *time_ms* of **ShowTemplate** and the specified display time has elapsed.
 - When **ShowTemplate** registered in **ExecuteMacro** is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

SetTemplateTextUnderline

Sets underline to text data.

ErrorCode **SetTemplateTextUnderline**(
 bool *underline*)

Parameters

underline

Underline specifying/canceling flag

Sets underline.

TRUE: Specifies underline.

FALSE: Cancels underline.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- From the text data after this API is executed, the underline is applied.
The underline can be set one by one.
- Specify the map ID at **SelectTemplateTextObject** before executing this API.
- Input text data at **SetTemplateTextData** after executing this API.
- The input text data is displayed to the screen when **ShowTemplate** is executed.
- This API setting is cleared under the following conditions.
 - When **SelectTemplate** is executed.
 - When other than 0 is specified with *time_ms* of **ShowTemplate** and the specified display time has elapsed.
 - When **ShowTemplate** registered in **ExecuteMacro** is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

SetTemplateTextSize

Sets character size to text data.

```
ErrorCode SetTemplateTextSize(  
    int horizontalScale,  
    int verticalScale )
```

Parameters

horizontalScale

Horizontal scale

Sets the horizontal character size.

See "6.3 Character Size" for specifiable values.

verticalScale

Vertical scale

Sets the vertical character size.

See "6.3 Character Size" for specifiable values.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- From the text data after this API is executed, the character size is applied.
The character size can be set one by one.
- Specify the map ID at **SelectTemplateTextObject** before executing this API.
- Input text data at **SetTemplateTextData** after executing this API.
- The input text data is displayed to the screen when **ShowTemplate** is executed.
- This API setting is cleared under the following conditions.
 - When **SelectTemplate** is executed.
 - When other than 0 is specified with *time_ms* of **ShowTemplate** and the specified display time has elapsed.
 - When **ShowTemplate** registered in **ExecuteMacro** is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

SetTemplateTextFont

Sets a character font to text data.

ErrorCode **SetTemplateTextFont**(
CharacterFont *font*)

Parameters

font

Character font

Sets a character font.

See "6.4 Character Font" for specifiable values.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- From the text data after this API is executed, the character font is applied.
The character font can be set one by one.
- Specify the map ID at **SelectTemplateTextObject** before executing this API.
- Input text data at **SetTemplateTextData** after executing this API.
- The input text data is displayed to the screen when **ShowTemplate** is executed.
- This API setting is cleared under the following conditions.
 - When **SelectTemplate** is executed.
 - When other than 0 is specified with *time_ms* of **ShowTemplate** and the specified display time has elapsed.
 - When **ShowTemplate** registered in **ExecuteMacro** is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

SetTemplateTextRegisteredFont

Sets the registered font used for text data.

ErrorCode **SetTemplateTextRegisteredFont**(
RegisteredFont *font*)

Parameters

font

Registered font

Sets the registered font.

See "6.5 Registered Font" for specifiable values.

The registered font specifying is ignored when the optional font is not registered.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- From the text data after this API is executed, the registered font is applied. The registered font can be set one by one.
- Specify the map ID at **SelectTemplateTextObject** before executing this API.
- Input text data at **SetTemplateTextData** after executing this API.
- The input text data is displayed to the screen when **ShowTemplate** is executed.
- This API setting is cleared under the following conditions.
 - When **SelectTemplate** is executed.
 - When other than 0 is specified with *time_ms* of **ShowTemplate** and the specified display time has elapsed.
 - When **ShowTemplate** registered in **ExecuteMacro** is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

SetCodePage

Specifies the codepage used for text data.

```
ErrorCode SetCodePage(  
    Codepage codepage )
```

Parameters

```
codepage  
Codepage  
    Sets the codepage.  
    See "6.6 Codepage" for specifiable values.
```

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- From the text data after this API is executed, the character code table is applied. The character code table can be set one by one.
- When the character code table is not set by this API, the following code page is set by the language setting of Windows OS.
 - When a language setting is Japanese: Katakana character set
 - When a language setting is other than Japanese: Codepage 1252
- The characters 0x80 to 0x9F and 0xE0 to 0xFF cannot be displayed when the template encoding specifying is Shift_JIS. Use a template whose encoding specifying is ISO-2022-JP to display.
- Specify the map ID at **SelectTemplateTextObject** before executing this API.
- This API setting is cleared under the following conditions.
 - When **SelectTemplate** is executed.

- When other than 0 is specified with *time_ms* of **ShowTemplate** and the specified display time has elapsed.
- When **ShowTemplate** registered in **ExecuteMacro** is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

SetInternationalCharacter

Sets the international character used for text data.

ErrorCode **SetInternationalCharacter**(
InternationalCharacter *internationalCharacter*)

Parameters

internationalCharacter

International character

Sets the international character.

See "6.7 International Character" for specifiable values.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- From the text data after this API is executed, the international character is applied. The international character can be set one by one.
- Specify the map ID at **SelectTemplateTextObject** before executing this API.
- This API setting is cleared under the following conditions.
 - When **SelectTemplate** is executed.
 - When other than 0 is specified with *time_ms* of **ShowTemplate** and the specified display time has elapsed.
 - When **ShowTemplate** registered in **ExecuteMacro** is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

SetTemplateTextRightSpacing

Sets the amount of right space to text data.

ErrorCode **SetTemplateTextRightSpacing**(
int *spacing*)

Parameters

spacing

The amount of character right space

Sets the amount of right space in pixel (px).

The valid range is 0 to 255.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- From the text data after this API is executed, the amount of character right space is applied.
The amount of right space can be set one by one.
- Specify the map ID at **SelectTemplateTextObject** before executing this API.
- Input text data at **SetTemplateTextData** after executing this API.
- The input text data is displayed to the screen when **ShowTemplate** is executed.
- This API setting is cleared under the following conditions.
 - When **SelectTemplate** is executed.
 - When other than 0 is specified with *time_ms* of **ShowTemplate** and the specified display time has elapsed.
 - When **ShowTemplate** registered in **ExecuteMacro** is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

SetTemplateTextColor

Sets the character color used for text data.

ErrorCode **SetTemplateTextColor**(
uint *color*)

Parameters

color

Character code table

Sets the character color.

See "6.8 Character Color" for specifiable values.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- From the text data after this API is executed, the character color is applied. The character color can be set one by one.
- Specify the map ID at **SelectTemplateTextObject** before executing this API.
- Input text data at **SetTemplateTextData** after executing this API.
- The input text data is displayed to the screen when **ShowTemplate** is executed.
- This API setting is cleared under the following conditions.
 - When **SelectTemplate** is executed.
 - When other than 0 is specified with *time_ms* of **ShowTemplate** and the specified display time has elapsed.
 - When **ShowTemplate** registered in **ExecuteMacro** is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

SetTemplateTextLineSpacing

Sets line spacing to text data.

ErrorCode **SetTemplateTextLineSpacing**(
int *spacing*)

Parameters

spacing

Line spacing

Sets line spacing in pixel (px).

The valid range is 0 to 255.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- Line spacing setting is valid only as following cases.
 - Text data is not entered in the specified map ID at **SelectTemplateTextObject**.
 - Text data entered in the map ID which is specified at **SelectTemplateTextObject** is registered right after a line feed.
- When a scroll is set to a text element of the specified map ID, specifying this API is not reflected.
- Specify the map ID at **SelectTemplateTextObject** before executing this API.
- Input text data at **SetTemplateTextData** after executing this API.
- The input text data is displayed to the screen when **ShowTemplate** is executed.

- This API setting is cleared under the following conditions.
 - When **SelectTemplate** is executed.
 - When other than 0 is specified with *time_ms* of **ShowTemplate** and the specified display time has elapsed.
 - When **ShowTemplate** registered in **ExecuteMacro** is executed.
- In use via a printer, this API is ignored when Display is not connected to the printer.

SetTemplateTextData

Inputs text data to show on the text element specified at **SelectTemplateTextObject**.

The method of syntax (a) inputs the text data by strings.

The method of syntax (b) inputs the text data by binary data.

- (a) ErrorCode **SetTemplateTextData**(
string *text*)
- (b) ErrorCode **DispSetTemplateTextData**(
byte[] *data*)

Parameters

text (data)

Text data

Inputs text data.

Data size which is able to be specified it once is 1 to 1020 bytes.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- This API encodes specified text data into text data which is enable to show on Display on the basis of settings at **SetInternationalCharacter** or **SetCodePage**, and displays. Text that could not be encoded is shown as '?'.
 - Select the template in **SelectTemplate** before executing this API.
 - After specifying the map ID of the selecting template with **SelectTemplateTextObject**, input text data with this API. This API is ignored when the map ID is not specified in **SelectTemplateTextObject**.
 - The input text data is displayed to the screen when **ShowTemplate** is executed.
 - The input text data is cleared under the following conditions.
 - When **SelectTemplate** is executed.
 - When other than 0 is specified with *time_ms* of **ShowTemplate** and the specified display time has elapsed.
 - When **ShowTemplate** registered in **ExecuteMacro** is executed.
 - This API is ignored when the no template is selected.
 - In use via a printer, this API is ignored when Display is not connected to the printer.

SetTemplateBarcodeData

Specifies a map ID of the barcode element on a selecting template, and inputs barcode data.

```
ErrorCode SetTemplateBarcodeData(  
    int mapID,  
    byte[] data )
```

Parameters

mapID

Map ID

Specifies map ID of a barcode element defined in a selecting template.

The valid range is 0 to 7.

This API is ignored when the specified map ID is not defined in the selecting template.

data

Barcode data

Inputs barcode data.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- See the Display command "Input Barcode Data" described in "Technical Reference for Display" for specification of barcode data.
- The input barcode data is displayed to the screen when **ShowTemplate** is executed.
- The input barcode data is cleared under the following conditions.
 - When **SelectTemplate** is executed.
 - When other than 0 is specified with *time_ms* of **ShowTemplate** and the specified display time has elapsed.
 - When **ShowTemplate** registered in **ExecuteMacro** is executed.
- This API is ignored when no template is selected.
- In use via a printer, this API is ignored when Display is not connected to the printer.

SetTemplateQrCodeData

Specifies a map ID of the qr element on a selecting template, and inputs QR Code data.

```
ErrorCode SetTemplateQrCodeData(  
    int mapID,  
    ModuleSize moduleSize,  
    ErrorCorrection errorCorrection,  
    QrDataMode mode,  
    bool qrQuietZoneFlag,  
    byte[] data )
```

Parameters

mapID

Map ID

Specifies map ID of a qr element defined in a selecting template.

The valid range is 0 to 7.

This API is ignored when the specified map ID is not defined in the selecting template.

moduleSize

Module size

Specifies module size.

See "6.9 Module Size" for specifiable values.

errorCorrection

Error correction level

Specifies error correction level.

See "6.10 Error Correction Level" for specifiable values.

This parameter is ignored when QR_MODULE_SIZE_0 is specified at *moduleSize*.

mode

Data mode

Specifies the data mode.

See "6.11 Data Mode" for specifiable values.

This parameter is ignored when QR_MODULE_SIZE_0 is specified at *moduleSize*.

qrQuietZoneFlag

Quiet zone flag

Specifies the presence/absence of quiet zone.

TRUE: With quiet zone

FALSE: Without quiet zone

This parameter is ignored when QR_MODULE_SIZE_0 is specified at *moduleSize*.

data

QR Code data

Inputs QR Code data.

The valid data length is 1 to 3909.

Input data according to the data input format of QR Code for the input of QR Code data.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- The input QR Code data is displayed to the screen when **ShowTemplate** is executed.
- The input QR Code data is cleared under the following conditions.
 - When **SelectTemplate** is executed.
 - When other than 0 is specified with *time_ms* of **ShowTemplate** and the specified display time has elapsed.
 - When **ShowTemplate** registered in **ExecuteMacro** is executed.
- This API is ignored when no template is selected.
- In use via a printer, this API is ignored when Display is not connected to the printer.

Image Registration

RegisterImageData

Registers image data.

```
ErrorCode RegisterImageData(  
    int imageID,  
    string label,  
    string fileName )
```

Parameters

imageID

Image ID

Specifies the ID of image data to register.

The valid range is 0 to 63.

Do not specify the image IDs of 49 to 63 because they are being used for the system.

label

Label name

Specifies the label name.

The characters that can be specified are ASCII characters 0x20 (space) to 0x7E (tilde) such as alphanumeric characters ('0' to '9', 'A' to 'Z', 'a' to 'z').

Do not use Unicode 0x00A5 ('¥').

The available number of characters is 0 to 32.

label is optional. Specify an empty string when the label name is not registered.

The specified label name can be retrieved by **GetObjectLabel**.

fileName

File path

Registers image data to the ID specified with *imageID*.

Specifies the file name of image data created in the prescribed format.

Supported file extensions are .jpg, .jpeg and .png. However, even the supported extensions may not be registered depending on the format.

The maximum file size that can be specified is 786432 bytes.

The maximum data size that can be registered is 480 horizontal × 272 vertical pixels (px).

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Notes

Registered data at the factory may be added or changed without prior notice for quality improvement.

Remarks

- Display is changed to Standby mode when this API is executed. A selecting template is deselected.
- In use via a printer, this API is ignored when Display is not connected to the printer.

UnregisterImageData

Deletes registered image data.

```
INT UnregisterImageData(  
    int imageID )
```

Parameters

imageID

Image ID

Specifies the ID of image data to delete.

The valid range is 0 to 63.

This API is ignored when image data is not registered in the specified ID.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- Display is changed to Standby mode when this API is executed. A selecting template is deselected.
- Used memory is not released even image data is deleted. The used memory can be reused after executing **Defragment**.
- In use via a printer, this API is ignored when Display is not connected to the printer.

RegisterSlideData

Registers slide data.

```
ErrorCode RegisterSlideData(  
    int slideID  
    string label  
    string fileName )
```

Parameters

slideID

Slide ID

Specifies the ID of slide data to register.

The valid range is 0 to 91.

Do not specify the slide IDs of 80 to 90 because they are being used for the system.

label

Label name

Registers slide data to the ID specified with *slideID*.

Specifies the label name.

The characters that can be specified are ASCII characters 0x20 (space) to 0x7E (tilde) such as alphanumeric characters ('0' to '9', 'A' to 'Z', 'a' to 'z').

Do not use Unicode 0x00A5 ('¥').

The available number of characters is 0 to 32.

label is optional. Specify an empty string when the label name is not registered.

The specified label name can be retrieved by **GetObjectLabel**.

fileName

File path

Specifies the file name of slide data created in the prescribed format.

Supported file extensions are .jpg, .jpeg and .png. However, even the supported extensions may not be registered depending on the format.

The maximum file size that can be specified is 786432 bytes.

The data size that can be registered is fixed to 480 horizontal × 272 vertical pixels (px).

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Notes

Registered data at the factory may be added or changed without prior notice for quality improvement.

Remarks

- Display is changed to Standby mode this API is executed. A selecting template is deselected.
- Execute **ShowSlide** to show the registered slide data.
- Specify the slide ID at **ShowTemplate** to use the registered slide data as a backscreen of the template.
- In use via a printer, this API is ignored when Display is not connected to the printer.

UnregisterSlideData

Deletes registered slide data.

ErrorCode **UnregisterSlideData**(
int *slideID*)

Parameters

slideID

Slide ID

Specifies the ID of slide data to delete.

The valid range is 0 to 91.

This API is ignored when slide data is not registered in the specified ID.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- Display is changed to Standby mode when this API is executed. A selecting template is deselected.
- In use via a printer, this API is ignored when Display is not connected to the printer.

Character Registration

RegisterUserDefinedCharacter

Registers user-defined characters.

ErrorCode **RegisterUserDefinedCharacter**(
string *fileName*)

Parameters

fileName

File path

Specifies the file name of user-defined character data created in the prescribed format.

Supported file extension is .bin.

See the Display command "Register User-defined Character" described in "Technical Reference of Display" for user-defined character data.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- Display is changed to Standby mode when this API is executed. A selecting template is deselected.
- User-defined characters are overwritten when this API is executed with user-defined character registered status.
- Use the template which encoding specifying is Shift_JIS for displaying user-defined characters.
- Specify CODE_PAGE_KATAKANA for *codePage* of **SetCodePage** before user-defined characters are displayed.
- Specify the character codes that can be specified for *text* of **SetTemplateTextData** or *data* of **SetTemplateTextBData** when user-defined characters are displayed.
The character codes that can be specified are as follows.
 - For character strings : 0xE000 to 0xE05D
 - For binary data : 0xEC40 to 0xEC7E and 0xEC80 to 0xEC9E
- In use via a printer, this API is ignored when Display is not connected to the printer.

UnregisterUserDefinedCharacter

Deletes registered user-defined characters.

ErrorCode **UnregisterUserDefinedCharacter()**

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- Display is changed to Standby mode when this API is executed. A selecting template is deselected.
- All registered user-defined characters are deleted.
- Used memory is not released even the user-defined character is deleted. The used memory can be reused after executing **DispDefragment**.
- In use via a printer, this API is ignored when Display is not connected to the printer.

RegisterOptionFont

Registers optional fonts.

ErrorCode **RegisterOptionFont**(
 int *startcode*,
 int *endcode*,
 int *width*,
 int *height*,
 string *fileName*)

Parameters

startcode

Character code for registration starting

Specifies the character code to start registration.

The valid range is 0x20 to 0xFF in ASCII code.

endcode

Character code for registration ending

Specifies the character code to end registration.

The valid range is 0x20 to 0xFF in ASCII code.

width

Character width

Specifies the character width in pixel (px).

height

Character height

Specifies the character height in pixel (px).

fileName

File path

Specifies the file name of optional font data created in the prescribed format. Supported file extension is .bin.

See the Display command "Register Optional Font" described in "Technical Reference of Display" for optional font data.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- Display is changed to Standby mode when this API is executed. A selecting template is deselected.
- When this API is executed with optional font registered status, the registered optional fonts are deleted, and a new memory area is allocated to register optional fonts. Used memory is not released even the registered optional fonts are deleted. The used memory can be reused after executing **Defragment**.
- In use via a printer, this API is ignored when Display is not connected to the printer.

UnregisterOptionFont

Deletes registered optional fonts.

ErrorCode **UnregisterOptionFont()**

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- Display is changed to Standby mode when this API is executed. A selecting template is deselected.
- All registered optional fonts are deleted.
- Used memory is not released even the optional font is deleted. The used memory can be reused after executing **Defragment**.
- In use via a printer, this API is ignored when Display is not connected to the printer.

Macro

ControlMacroRegistration

Specifies start and end of macro registration.

ErrorCode **ControlMacroRegistration**(
int *macroID*,
MacroRegistrationFunction *control*)

Parameters

macroID

Macro ID

Specifies the ID of macro to register.

The valid range is -1 to 127.

Do not specify the macro IDs of 120 to 126 because they are being used for the system.

Macro ID	Contents
120	During Bluetooth pairing
121	Cutter error
122	Vp voltage error
123	Head temperature error
124	Hardware error
125	Cover open error
126	Out-of-paper error
127	Standby (Initial value)

control

Macro processing selection

Selects macro processing.

See "6.12 Macro Processing Selection" for specifiable values.

The procedure of macro processing selection is as follows.

Starts macro registration:

Starts macro registration when ControlStart is specified to *control*.

Specify -1 to *macroID*.

The APIs which are registration targets and their data are buffered in the internal memory of the SDK after registration is started.

The maximum data size that can register macro is 1024 bytes.

The APIs which are not registration targets, and their data are executed immediately without registration.

The APIs which are registration targets are as follows.

- **SelectTemplate**
- **SetTemplateImageData**
- **SelectTemplateTextObject**
- **SetTemplateTextData**
- **SetTemplateTextAlignment**
- **SetTemplateTextLeftMargin**
- **SetTemplateTextBold**

- **SetTemplateTextUnderline**
- **SetTemplateTextSize**
- **SetTemplateTextFont**
- **SetRegisteredFont**
- **SetCodePage**
- **SetInternationalCharacter**
- **SetTemplateTextRightSpacing**
- **SetTemplateTextColor**
- **SetTemplateTextLineSpacing**
- **SetTemplateBarcodeData**
- **SetTemplateQrCodeData**
- **ShowTemplate**
- **ShowSlide**

Ends after registering macro:

When ControlRegist is specified to *control*, specifies a macro to specified macro ID and ends this API.
Specify the macro ID to register to *macroID*.
Display is changed to Standby mode when this API is executed. A selecting template is deselected.

Ends without registering macro:

When ControlCancel is specified to *control*, discards the APIs which are registration targets and their data that has been buffered, and ends this API.
A selecting template is deselected.

Deletes the registered macro:

Specify ControlStart to *control* and specify -1 to *macroID*, and call this API.
Specify ControlRegist to *control* and specify the macro ID to delete to *macroID*, and call this API.
Display is changed to Standby mode when this API is executed. A selecting template is deselected.

The registered macro can be executed in **ExecuteMacro**.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Notes

Registered data at the factory may be added or changed without prior notice for quality improvement.

Remarks

- In use via a printer, this API is ignored when Display is not connected to the printer.

Memory Operation

Defragment

Optimizes memory area.

ErrorCode **Defragment**(
MemoryArea *area*)

Parameters

area
Memory area
Specifies user area.
See "6.13 Memory Area" for specifiable values.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- Display is changed to Standby mode when this API is executed. A selecting template is deselected.
- Do not turn off the device during optimization.
- Optimization may take several minutes.
- In use via a printer, this API is ignored when Display is not connected to the printer.

GetMemory

Gets the remaining capacity of memory area.

ErrorCode **GetMemory**(
MemoryArea *area*
out unit *size*)

Parameters

area
Memory area
Specifies memory area.
See "6.13 Memory Area" for specifiable values.

size
Received data buffer
Specifies the buffer for storing data to retrieve.
The received data size is stored when the control is returned from the API.

Return value

- On success: Returns the available remaining capacity excluding the released area.
- On failure: Returns an error code.
See "Chapter 5 Error Code List" for details of the error code.

InitializeMemoryArea

Initializes memory area.

```
ErroroCode InitializeMemoryArea(  
    MemoryArea area )
```

Parameters

- area*
Memory area
Specifies memory area.
See "6.13 Memory Area" for specifiable values.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Notes

The data registered by following APIs are deleted when MEMORY_DISPLAY_USERMEMORY is specified to *area* and the memory area is initialized.

- RegisterTemplate
- RegisterImageData
- ControlMacroRegistration
- RegisterUserDefinedCharacter
- RegisterOptionFont

Part of data which is registered at the factory to use for the system is deleted either. Therefore, when Display is used via a printer, the guide screen cannot be displayed if an error occurs in the printer.

When Display is used alone, the guide screen is not displayed.

The used memory area can be reused after executing Defragment.

Remarks

- Display is changed to Standby mode when this API is executed. A selecting template is deselected.
- When user area is initialized, the registered templates, image data, user-defined fonts, optional fonts and macro functions are all cleared, and the user area will be in the initial state.
- Do not turn off the device during initialization.
- Initialization may take several minutes.
- In use via a printer, this API is ignored when Display is not connected to the printer.

Response Function

GetRegisteredStateList

Gets the presence or absence of contents registration.

```
ErrorCode GetRegisteredStateList(  
    ContentsType type  
    out RegisteredState[] data )
```

Parameters

type

Contents type

Specifies the contents type to get the response.

See "6.1 Contents Type" for specifiable values.

data

Received data buffer

Specifies the buffer that stores the presence or absence of contents registration.

The registration status of the contents type specified by *type* is stored in the buffer sequentially from the ID 0.

STATE_REGISTERED is stored when the data is registered, and

STATE_UNREGISTERED is stored when the data is not registered.

The buffer size to be stored depends on the contents type.

Contents Type	Buffer Size to be Stored
Template	128
Slide	24
Image	64

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

GetObjectLabel

Gets the label names of registered contents.

```
ErrorCode GetObjectLabel(  
    ContentsType type,  
    int objectID,  
    out string data )
```

Parameters

type
Contents type
Specifies the contents type to get the response.
See "6.1 Contents Type" for specifiable values.

objectID
ID of contents
For template:
Specify *templateID* of the template registered in **RegisterTemplate**.
For slide data:
Specify *slideID* of slide data registered in **RegisterSlideData**.
For image data:
Specify *imageID* of image data registered in **RegisterImageData**.

data
Received data buffer
Specifies the buffer that stores the data to retrieve.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

GetResponseRequest

Gets the specified response codes when this API is executed.

```
ErrorCode GetResponseRequest(  
    int id  
    int timeout )
```

Parameters

id
Response code
Specifies the response code to be returned by execution of this API.
The valid range is 0 to 255.
However, the upper 4 bits are ignored.

timeout
Timeout period
Specifies the time to wait for success of this API in milliseconds (ms).
The valid range is from 3000 to 90000.
When the value is specified less than 3000, the time is set to 3000 ms.
When the value is specified more than 90000, the time is set to 90000 ms.

Return value

Returns an error code. See "Chapter 5 Error Code List" for details of the error code.

Remarks

- By executing this API at the end of other APIs or transmission data and returning the response code specified by *id*, the completion of data processing execution can be confirmed from the outside.

Chapter 5 Error Code List

This chapter describes the error codes.

5.1 Error Code List

Major error codes are as follows:

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_TYPE	-10	Open type parameter error.
ERR_OPENED	-20	Specified printer has already been opened.
ERR_NO_PRINTER	-30	Specified printer driver does not exist.
ERR_HANDLE	-60	API handle value is incorrect.
ERR_TIMEOUT	-70	Timeout or busy state occurs.
ERR_ACCESS	-80	Printer cannot be accessed because it is not reachable on the network, or the service is stopped during access.
ERR_PARAM	-90	Parameter is incorrect.
ERR_NOT_SUPPORT	-100	Function is not supported.
ERR_OFFLINE	-110	Printer is disconnected or offlined.
ERR_NON_TARGETTED_DRIVER	-120	Specified printer driver is not supported.
ERR_DISK_FULL	-170	Printer is busy.
ERR_ENTRY_OVER	-190	Processing capacity is exceeded.
ERR_EXIST	-210	Existing module is called.
ERR_NOT_FOUND	-220	File cannot be found, or it is not registered.
ERR_WORKAREA_NO_MEMORY	-260	Specified memory size is insufficient.
ERR_WORKAREA_FAILED	-280	Memory cannot be reserved.
ERR_EXEC_FUNCTION	-310	Function is not available because it is being used by other thread or process.
ERR_SPL_NOT_EXIST	-350	Spooler service or SII Printer Software service has not been started.
ERR_LOCKED	-1000	Printer is locked.

Macro Definition (Constant)	Value	Description
ERR_UNLOCKED	-1010	UnlockPrinter was executed, even though printer is not locked.
ERR_INVALID_DATA	-1020	Invalid data is specified.
ERR_READ_FAULT	-1030	Data cannot be received from printer.
ERR_WRITE_FAULT	-1040	Data cannot be sent to printer.
ERR_CANCELLED	-1050	Function has been canceled.
ERR_UNKNOWN_PORT	-1070	Port is not supported.
ERR_INVALID_PRINTER_STATE	-1080	Printer status is abnormal.
ERR_BAD_ENVIRONMENT	-1090	Printer driver may not be installed normally.

Chapter 6 Argument Information

This chapter describes the arguments.

6.1 Contents Type

The contents types and descriptions are as follows.

Contents Type		Description
.NET API	Win32 API	DispGetObjectLabel GetObjectLabel DispGetRegisteredStateList GetRegisteredStateList
TYPE_TEMPLATE	1	Template
TYPE_SLIDE	2	Slide
TYPE_IMAGE	3	Image

6.2 Alignment

The alignment and descriptions are as follows.

Alignment		Description
.NET API	Win32 API	
ALIGNMENT_LEFT	0	Left aligned
ALIGNMENT_CENTER	1	Align center
ALIGNMENT_RIGHT	2	Right aligned

6.3 Character Size

The character size and descriptions are as follows.

Character Size		Description
.NET API	Win32 API	
1		Horizontal scale : 1x
2		Horizontal scale : 2 times
3		Horizontal scale : 3 times
4		Horizontal scale : 4 times
1		Vertical scale : 1x
2		Vertical scale : 2 times
3		Vertical scale : 3 times
4		Vertical scale : 4 times

6.4 Character Font

The character font and descriptions are as follows.

Character Font		Description	
.NET API	Win32 API	DispSetTemplateTextFont SetTemplateTextFont	DispRegisterUserDefinedCharacter RegisterUserDefinedCharacter
FONT_A	0	Font A (24 × 12)	Font A (24 × 24)
FONT_B	1	Font B (16 × 8)	Font A (16 × 16)

6.5 Registered Font

The registered font and descriptions are as follows.

Registered Font		Description
.NET API	Win32 API	
FONT_STANDARD	0	Standard font
FONT_OPTION	1	Optional font

6.6 Codepage

The codepage and descriptions are as follows.

Code page		Description
.NET API	Win32 API	
CODE_PAGE_437	0	USA, Standard Europe (Code Page437)
CODE_PAGE_KATAKANA	1	Katakana
CODE_PAGE_850	2	Multilingual (Code Page850)
CODE_PAGE_860	3	Portuguese (Code Page860)
CODE_PAGE_863	4	Canadian-French (Code Page863)
CODE_PAGE_865	5	Nordic (Code Page865)
CODE_PAGE_857	13	Turkish (Code Page857)*1
CODE_PAGE_737	14	Greek (Code Page737)
CODE_PAGE_1252	16	Latin (Code Page1252)
CODE_PAGE_866	17	Russian (Code Page866)
CODE_PAGE_852	18	Eastern Europe (Code Page852)
CODE_PAGE_858	19	Euro (Code Page858)
CODE_PAGE_855	34	Cyrillic (Code Page855)
CODE_PAGE_864	37	Arabic (Code Page864)*1*2
CODE_PAGE_1250	45	Central European (Code Page1250)
CODE_PAGE_1251	46	Cyrillic (Code Page1251)
CODE_PAGE_1253	47	Greek (Code Page1253)
CODE_PAGE_1254	48	Turkish (Code Page1254)

*1: Euro symbol cannot be displayed.

*2: Font B cannot be displayed.

6.7 International Character

The international character and descriptions are as follows.

International Character		Description
.NET API	Win32 API	
COUNTRY_USA	0	USA
COUNTRY_FRANCE	1	France
COUNTRY_GERMANY	2	Germany
COUNTRY_ENGLAND	3	United Kingdom
COUNTRY_DENMARK_1	4	Denmark I
COUNTRY_SWEDEN	5	Sweden
COUNTRY_ITALY	6	Italy
COUNTRY_SPAIN	7	Spain I
COUNTRY_JAPAN	8	Japan
COUNTRY_NORWAY	9	Norway
COUNTRY_DENMARK_2	10	Denmark II
COUNTRY_SPAIN_2	11	Spain II
COUNTRY_LATIN_AMERICA	12	Latin America
COUNTRY_ARABIA	17	Arabia

6.8 Character Color

The character color and descriptions are as follows.

Character Color	Description
.NET API Win32 API	
0x000000	Black
0x0000FF	Blue
0x00FF00	Green
0x00FFFF	Cyan
0xFF0000	Red

Character Color	Description
.NET API Win32 API	
0xFF00FF	Magenta
0xFFFF00	Yellow
0xFFFFFFFF	White
Optional	Any color can be specified. Input in RGB 24-bit notation. However, the lower 3 bits of each RGB color are ignored.

6.9 Module Size

The module size and descriptions are as follows.

Module Size		Description
.NET API	Win32 API	
QR_MODULE_SIZE_0	0	Module size : Applies the value of template.
QR_MODULE_SIZE_2	2	Module size : 2
QR_MODULE_SIZE_3	3	Module size : 3
QR_MODULE_SIZE_4	4	Module size : 4
QR_MODULE_SIZE_5	5	Module size : 5
QR_MODULE_SIZE_6	6	Module size : 6
QR_MODULE_SIZE_7	7	Module size : 7
QR_MODULE_SIZE_8	8	Module size : 8
QR_MODULE_SIZE_9	9	Module size : 9
QR_MODULE_SIZE_10	10	Module size : 10
QR_MODULE_SIZE_11	11	Module size : 11

6.10 Error Correction Level

The error correction level and descriptions are as follows.

Error Correction Level		Description
.NET API	.NET API Win32 API	
QR_ERROR_CORRECTION_L	76	Error correction level : 'L' (0x4C)
QR_ERROR_CORRECTION_M	77	Error correction level : 'M' (0x4D)
QR_ERROR_CORRECTION_Q	81	Error correction level : 'Q' (0x51)
QR_ERROR_CORRECTION_H	72	Error correction level : 'H' (0x48)

6.11 Data Mode

The data mode and descriptions are as follows.

Data Mode		Description
.NET API	.NET API Win32 API	
QRDATAMODE_NUMERIC	78	Data mode : 'N' (0x4E)
QRDATAMODE_ALPHANUMERIC	65	Data mode : 'A' (0x41)
QRDATAMODE_8BITBYTE	66	Data mode : 'B' (0x42)
QRDATAMODE_KANJI	75	Data mode : 'K' (0x4B)
QRDATAMODE_MIXTURE	77	Data mode : 'M' (0x4D)

6.12 Macro Processing Selection

The macro processing and descriptions are as follows.

Macro Processing		Description
.NET API	Win32 API	
MACRO_REGISTRATION_START	0	Starts macro registration
MACRO_REGISTRATION_REGIST	1	Ends macro registration
MACRO_REGISTRATION_CLEAR	2	Ends without macro registration

6.13 Memory Area

The memory area and description is as follows.

Memory Area		Description
.NET API	Win32 API	
MEMORY_DISPLAY_USERMEMORY	1	User area