SII SDK for Windows
Application Programmer's Guide

Rev.03

[Products]

SLP720RT Series

SLP721RT Series

Seiko Instruments Inc.

Rev.01 March 2022
Rev.02 October 2022
Rev.03 April 2023

Copyright©2022-2023 by Seiko Instruments Inc.
All rights reserved.

Microsoft® and Windows® are registered trademarks of Microsoft Corporation in the U.S., Japan, and other countries.
All other trademarks are the properties of their respective companies.

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 "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 Products".

Target Products

The products covered by this manual are as follows.

Printer	Interface	Printer Driver
SLP720RT Series SLP721RT Series*1	USB	"SII Printer Driver for Windows" for SLP720RT Series
	TCP/IP	

*1: The sales destination is Japan only.

Notation in This Manual

The notation in this manual is described.

Terms

The terms used in this manual are defined as below.

Term	Description
Technical Reference	Technical Reference shown as follows: ·SLP720RT SERIES THERMAL PRINTER TECHNICAL REFERENCE
User's Guide	User's Guide shown as follows: ·SLP720RT SERIES THERMAL PRINTER User's Guide
Printer command	Instruction to control the printer, described in "Technical Reference".
ASB setting command (ASB: Automatic Status Back)	Printer command "Automatic Status Back Enable/Disable". See "Chapter 6 Command Functions" in "Technical Reference" for details of printer commands.
Printer status	Printer status information that can be retrieved by the SDK. This information includes the status to respond to the printer command "Automatic Status Back Enable/Disable" and extended status. See "6.1 Printer Status List" for details of the printer status.

Symbols

The symbols used in this manual are described below.

Caution

◆ Notes and limitations are described.

Reference

• Supplemental information and related matters are described.

Chapter 1 Overview 1-1

1.1 Operating Conditions 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

3.4 API Details 3-4

OpenMonPrinter 3-4

CloseMonPrinter 3-5

LockPrinter 3-5

UnlockPrinter 3-6

DirectIO 3-7

DirectIOEx 3-8

Reset 3-9

GetStatus 3-10

SetStatusBackFunction 3-11

SetStatusBackWnd 3-12

CancelStatusBack 3-13

PowerOff 3-14

GetCounter 3-14

ResetCounter 3-15

GetType 3-15

GetPrnCapability 3-16

OpenDrawer 3-17

ControlFeature 3-18

SendDataFile 3-19

DirectSendRead 3-19

GetPrintSmartLabelMode 3-21

SetPrintSmartLabelMode 3-21

SelectSmartLabelFile 3-22

PrintSmartLabelImageData 3-22

SetSmartLabelCharacterSet 3-23

ReplaceSmartLabelTextData 3-24

ReplaceSmartLabelImageData 3-26

ReplaceSmartLabelBarcodeData 3-27
GetProperty 3-28
SetProperty 3-29

Chapter 4 .NET API 4-1

4.1 Development Language 4-1
4.2 Library File 4-1
4.3 Printer 4-2
4.3.1 API List 4-2
Common API 4-2
Specific API 4-2
4.3.2 Common Property 4-4
LastError 4-4
IsValid 4-4
4.3.3 Common Method 4-5
OpenMonPrinter 4-5
CloseMonPrinter 4-5
4.3.4 Specific Property 4-6
Status 4-6
PrintSmartLabelMode 4-6
4.3.5 Specific Method 4-7
LockPrinter 4-7
UnlockPrinter 4-7
DirectIOEx 4-8
ResetPrinter 4-9
SetStatusBack 4-10
CancelStatusBack 4-10
PowerOff 4-11
GetCounter 4-11
ResetCounter 4-12
GetType 4-12
GetPrnCapability 4-13
OpenDrawer 4-14
ControlFeature 4-15
SendDataFile 4-15
DirectSendRead 4-16
SelectSmartLabelFile 4-17
PrintSmartLabelImageData 4-18
ReplaceSmartLabelTextData 4-19

ReplaceSmartLabelImageData 4-20
ReplaceSmartLabelBarcodeData 4-21
GetProperty 4-22
SetProperty 4-23
4.3.6 Event 4-24
StatusCallback 4-24

Chapter 5 Error Code List 5-1

5.1 Error Code List 5-1

Chapter 6 Argument Information 6-1

6.1 Printer Status List 6-1
6.2 Counter ID 6-2
6.3 Printer ID 6-3
6.4 Type ID 6-3
6.5 Target Drawers 6-4
6.6 Drawer Drive Time 6-4
6.7 Printer Function 6-4
6.8 Paper ID when Printing Label File 6-5
6.9 Property ID 6-5

Chapter 7 Command Definition File 7-1

7.1 Overview 7-1
7.2 Format 7-1
7.3 How to Use 7-3

Chapter 8 Printing Label Function 8-1

8.1 Structure of Label File 8-2
8.1.1 Types of Objects and Support in SDK 8-2
8.1.2 Precautions for Printing Label File Using SDK 8-3
8.2 Method for Using Label File 8-5
8.2.1 Print Label File as It Is from Library 8-5
8.2.2 Replace Object Data in Label File and Print 8-5

Chapter 1 Overview

This chapter describes the overview of the SDK.

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

Using the SDK allows to directly control printers in an application development and to design the application independent of the port type. Also, some values of members in the private section of the DEVMODE can be retrieved or changed.

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 sample program provided for each development language.

1.1 Operating Conditions

The operating conditions of the SDK basically follow the operating environment of the printer driver, and the settings and limitations of the printer. For details of the operating environment of the printer driver, see "SII Printer Driver for Windows User's Guide" for SLP720RT/SLP721RT series.

In addition, the following operating conditions must be met.

- When using the .NET API, the SDK requires .NET Framework Version 3.5 or 4.0 to 4.8. 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 and the printer spool function is disabled.

Chapter 2 Installation

This software is installed at the same time with the printer driver.

For the installation procedure of the printer driver, see the installation part of "SII Printer Driver for Windows User's Guide" for SLP720RT/SLP721RT series.

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:

- SlpRtApi.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 printer driver.

3.3 API List

The APIs implemented in the Win32 API are as follows.

API	Function Summary
OpenMonPrinterA[*1] **OpenMonPrinterW**[*1]	Starts using the Win32 API and returns the API handle.
CloseMonPrinter	Ends using the Win32 API.
LockPrinter	Locks all data transmission and hardware reset requests from other processes to the printer.
UnlockPrinter	Unlocks the access prohibition (lock) from other processes by **LockPrinter**.
DirectIO	Sends and receives binary data. (Receive data does not include responses of the ASB setting command.)
DirectIOEx	Sends and receives binary data.
Reset	Performs hardware reset of the printer.
GetStatus	Gets the latest printer status.
SetStatusBackFunction	Registers the callback function called when a change in the printer status is detected.
SetStatusBackWnd	Registers the window handle of a button for which the click event is called and the variable to set the printer status when a change in the printer status is detected.
CancelStatusBack	Unregisters the callback function registered by **SetStatusBackWnd** or **SetStatusBackFunction**.
PowerOff	Turns off the printer power.
GetCounter	Gets the maintenance counter.
ResetCounter	Initializes the maintenance counter.
GetType	Gets various IDs of the printer.
GetPrnCapability	Gets the printer information.
OpenDrawer	Drives the specified drawer. This API is supported only by SLP721RT.
ControlFeature	Controls functions of the printer other than printing. This API is supported only by SLP721RT.
SendDataFileA[*1] **SendDataFileW**[*1]	Registers the contents of the command definition file to the memory in the SDK.
DirectSendReadA[*1] **DirectSendReadW**[*1]	Sends transmission data corresponding to the specified command name with respect to the command definition file registered by **SendDataFile**.
GetPrintSmartLabelMode	Gets the paper when printing label file.
SetPrintSmartLabelMode	Specifies the paper when printing label file.
SelectSmartLabelFileA[*1] **SelectSmartLabelFileW**[*1]	Specifies a label file (*.sl) to use the printing label function.
PrintSmartLabelImageData	Prints labels selected by **SelectSmartLabelFile**.
SetSmartLabelCharacterSet	Specifies the codepage of the text data to replace by **ReplaceSmartLabelTextDataA** or **ReplaceSmartLabelBarcodeDataA**.

API	Function Summary
ReplaceSmartLabelTextDataA[*1] **ReplaceSmartLabelTextDataW**[*1]	Replaces the value of the text object of the label file (*.sl) selected by **SelectSmartLabelFile**.
ReplaceSmartLabelImageDataA[*1] **ReplaceSmartLabelImageDataW**[*1]	Replaces the value of the image object of the label file (*.sl) selected by **SelectSmartLabelFile**.
ReplaceSmartLabelBarcodeDataA[*1] **ReplaceSmartLabelBarcodeDataW**[*1]	Replaces the value of the barcode object of the label file (*.sl) selected by **SelectSmartLabelFile**.
GetProperty	Gets a part of the print settings.
SetProperty	Changes a part of the print settings.

*1 : Specify 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.

Caution

◆ SLP720RT/SLP721RT do not support the APIs of Display or the barcode scanner.

3.4 API Details

OpenMonPrinter

Starts using the Win32 API and returns the API handle.

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

Parameters

i_type
Open type
2 (fixed)

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

Return value

On success:	Returns the API handle to identify the printer.
On failure:	Returns an error code. See "5.1 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 connection destination of the printer driver to USB or TCP/IP.
- This API succeeds even when the printer is not connected or the printer power is turned off.

CloseMonPrinter

Ends using the Win32 API.

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 "5.1 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

Locks all data transmission and hardware reset requests from other processes to the printer.

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 "5.1 Error Code List" for details of the error code.

Remarks

- When any APIs that directly access the printer are called from other processes during the time from calling this API until **UnlockPrinter** is called, those APIs fail.
- Only other processes are locked by this API. Therefore, calls from the process itself are not locked regardless of API handles or threads.
- By this API, multiple locking is possible up to 99 times with the same API handle. In order to release the lock, call **UnlockPrinter** the same number of times as for this API.
- When **LockPrinte**r is being called by another process, calling this API fails.
- For using the TCP/IP connection, release the lock before the time of no transmission exceeds the TCP/IP receive timeout period. If it is locked, it may cause the data to be missed or the data sent from other hosts to interrupt the printer.
  For the receive timeout period, see "LAN communication settings" in "3.2.3 Setting Screen for Each Connection Method" stated in the "SII Communication Setting Utility for Windows User's Guide" for SLP720RT/SLP721RT series.

UnlockPrinter

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

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 "5.1 Error Code List" for details of the error code.

Remarks

- When **LockPrinter** has been called multiple times, this API must be called the same number of times as for **LockPrinter** in order to release the lock.

DirectIO

Sends 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 send
Specifies the size of data to send.

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

io_rlen
Size of data to receive
Specifies the maximum length of data to be received from the printer.
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 flag to specify the receive operation from the following:
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 "5.1 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.

- In the data to send, do not include the ASB setting command that disables the response. The API that gets the printer status will not work properly.
- Receive data does not include the response of the ASB setting command. To get data that includes the response of the ASB setting command, call **DirectIOEx**.
- This API can be aborted by **Reset**.
- When **LockPrinter** is being called by another process, calling this API fails.
- When the printer is not connected or in a communication disabled state, this API fails.

DirectIOEx

Sends 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 send
Specifies the size of the data to send.

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

io_rlen
Size of data to receive
Specifies the maximum length of data to be received from the printer.
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 flag to specify the receive operation from the following:
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:
0: Gets the data excluding the response of the ASB setting command.
1: Gets the data including the response of the ASB setting command.

Return value

On success: Returns 0.

On failure: Returns an error code.
See "5.1 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.
- In the data to send, do not include the ASB setting command that disables the response. The API that gets the printer status will not work properly.
- This API can be aborted by **Reset**.
- When **LockPrinter** is being called by another process, calling this API fails.
- When the printer is not connected or in a communication disabled state, this API fails.

Reset

Performs a hardware reset of the printer.

INT **Reset**(
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 "5.1 Error Code List" for details of the error code.

Remarks

- Performs hardware reset of the printer using the communication protocol (without using printer commands).
- After called this API, wait a few seconds to send the data. If data transmission is performed right after calling this API, it may cause the data to be missed.
- When this API is called, the following APIs are aborted.
  - **DirectIO**
  - **DirectIOEx**
  - **DirectSendRead**
- During the call of this API, the printer status response will be "No response". See "6.1 Printer Status List" for details of the printer status.
- When **LockPrinter** is being called by another process, calling this API fails.
- When the printer is not connected or in a communication disabled state, this API fails.

GetStatus

Gets the latest printer status.

INT **GetStatus**(
INT *i_hdl*,
LPDWORD *o_status*)

Parameters

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

o_status
Printer status variable
Specifies a variable to store the printer status.

Return value

On success: Returns 0.
On failure: Returns an error code.
See "5.1 Error Code List" for details of the error code.

Remarks

- When reconnection to the printer is detected, the printer status is the status received last at the time.
- See "6.1 Printer Status List" for details of the printer status.

SetStatusBackFunction

Registers the callback function called when a change in the printer status is detected.

INT **SetStatusBackFunction**(
INT *i_hdl*,
INT (CALLBACK EXPORT **lpStatusCB*) (DWORD *o_st*))

Parameters

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

lpStatusCB
Callback function address
Specifies the address of an application-defined callback function that receives the printer status.
When NULL is specified, monitoring of the printer status is canceled.

o_st
Printer status variable
Specifies the variable in which the printer status is stored.

Return value

On success: Returns 0.

On failure: Returns an error code.
See "5.1 Error Code List" for details of the error code.

Remarks

- When a callback function is registered by this API, the callback function is called with the current printer status.
- Even when the printer status is received, the callback function will not be called when the printer status has not changed from when it was last received.
- When reconnection to the printer is detected, the printer status is the status received last at the time.
- When this API is called in the state that the callback function is registered, the already registered function becomes disabled, and a new callback function is registered.
- Even when this API is called specifying the already registered and enabled callback function again, the printer status response immediately after it is performed.
- The callback function registered by this API is unregistered by the following APIs:
  - **CloseMonPrinter**
  - **CancelStatusBack**
- The following APIs cannot be called from the registered callback function with the same API handle:
  - **CloseMonPrinter**
  - **SetStatusBackFunction**
  - **SetStatusBackWnd**
  - **CancelStatusBack**
- The return value of the callback function is ignored.
- The time from receiving the printer status until calling the callback function is not guaranteed.

- See "6.1 Printer Status List" for details of the printer status.

SetStatusBackWnd

Registers the window handle of a button for which the click event is called and the variable to set the printer status when a change in the printer status is detected.

INT **SetStatusBackWnd**(
INT *i_hdl*,
HANDLE *i_Wnd*,
LPDWORD *o_status*)

Parameters

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

i_Wnd
Window handle
Specifies the window handle of the button that sends the click event.
When NULL is specified, this API is canceled.

o_status
Printer status variable
Specifies the variable to store the printer status.

Return value

On success: Returns 0.

On failure: Returns an error code.
See "5.1 Error Code List" for details of the error code.

Remarks

- When the button window handle is registered by this API, the click event is called with the current printer status.
- Even when the printer status is received, the click event will not be called when the printer status has not changed from when it was last received.
- When reconnection to the printer is detected, the printer status is the status received last at the time.
- When this API is called in the state that the button window handle is registered, the already registered button window handle becomes disabled and a new button window handle is registered.
- Even when this API is called specifying the already registered and enabled button window handle again, the printer status response immediately after it is performed.
- This API is canceled by the following APIs:
  - **CloseMonPrinter**
  - **CancelStatusBack**

- The following APIs cannot be called from the processing routine of the called click event with the same API handle:
  - **CloseMonPrinter**
  - **SetStatusBackFunction**
  - **SetStatusBackWnd**
  - **CancelStatusBack**
- The return value of the click event is ignored.
- The time from receiving the printer status until calling the click event is not guaranteed.
- See "6.1 Printer Status List" for details of the printer status.

CancelStatusBack

Unregisters the callback function which was called by **SetStatusBackWnd** or **SetStatusBackFunction.**

INT **CancelStatusBack**(
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 "5.1 Error Code List" for details of the error code.

Remarks

- This API succeeds even when registration has not been performed by either **SetStatusBackFunction** or **SetStatusBackWnd**.

PowerOff

Turns off the printer power.

INT **PowerOff**(
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 "5.1 Error Code List" for details of the error code.

GetCounter

Gets the maintenance counter.

INT **GetCounter**(
INT *i_hdl*,
WORD *i_readno*,
LPDWORD *o_readcounter*)

Parameters

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

i_readno
Counter ID
Specifies the counter ID of the maintenance counter to get.
See "6.2 Counter ID" for specifiable values.

o_readcounter
Counter variable
Specifies a variable to store the retrieved counter value.

Return value

On success: Returns 0.
On failure: Returns an error code.
See "5.1 Error Code List" for details of the error code.

Remarks

- When **LockPrinter** is being called by another process, calling this API fails.
- When the printer is not connected or in a communication disabled state, this API fails.

ResetCounter

Initializes the maintenance counter.

INT **ResetCounter**(
INT *i_hdl*,
WORD *i_readno*)

Parameters

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

i_readno
Counter ID
Specifies the counter ID of the maintenance counter to initialize the counter value.
See "6.2 Counter ID" for specifiable values.

Return value

On success: Returns 0.
On failure: Returns an error code.
See "5.1 Error Code List" for details of the error code.

Remarks

- Confirm with **GetCounter** that the value of the counter ID specified by this API has been initialized.
- When **LockPrinter** is being called by another process, calling this API fails.
- When the printer is not connected or in a communication disabled state, this API fails.

GetType

Gets various IDs of the printer.

INT **GetType**(
INT *i_hdl*,
LPBYTE *o_typeID*
LPBYTE *o_fontID*
LPBYTE *o_exrom*
LPBYTE *o_special*)

Parameters

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

o_typeID
ID1
3 is stored.

o_fontID
ID2
2 is stored.

o_exrom
Reserved
Specify NULL.

o_special
Reserved
Specify NULL.

Return value

On success: Returns 0.
On failure: Returns an error code.
See "5.1 Error Code List" for details of the error code.

Remarks

- When **LockPrinter** is being called by another process, calling this API fails.
- When the printer is not connected or in a communication disabled state, this API fails.

GetPrnCapability

Gets the printer information.

INT **GetPrnCapability**(
INT *i_hdl*,
BYTE *i_id*,
LPBYTE *io_datsize*,
LPBYTE *o_dat*)

Parameters

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

i_id
Printer ID
Specifies the printer ID of the printer information to get.
See "6.3 Printer ID" for specifiable values.

io_datsize
Size of data to receive
Specifies the buffer size to store the printer information to get.
When the control returns from the API, the data size of the retrieved printer information is stored.
When the specified buffer size is smaller than the response size of the printer information to get, this API fails and the specified response size of the printer information is stored.

o_dat
Buffer of data to receive
Specifies the buffer that stores the retrieved printer information.

Return value

On success: Returns 0.

On failure: Returns an error code.
See "5.1 Error Code List" for details of the error code.

Remarks

- When LockPrinter is being called by another process, calling this API fails.
- When the printer is not connected or in a communication disabled state, this API fails.

OpenDrawer

Drives the specified drawer.
Supported only by SLP721RT.

INT **OpenDrawer**(
INT *i_hdl*,
BYTE *i_drawer*,
BYTE *i_pulse*)

Parameters

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

i_drawer
Target drawer
Specifies the target drawer to drive.
See "6.5 Target Drawers" for specifiable values.

i_pulse
Drawer drive time
Specifies ON/OFF time of the drawer driving pulse. The ON time and the OFF time are specified as the same.
See "6.6 Drawer Drive Time" for specifiable values.

Return value

On success: Returns 0.

On failure: Returns an error code.
See "5.1 Error Code List" for details of the error code.

Remarks

- When **LockPrinter** is being called by another process, calling this API fails.
- When the printer is not connected or in a communication disabled state, this API fails.
- As for the drawer control time, follow the specifications of your drawer.

ControlFeature

Controls functions of the printer other than printing.
Supported only by SLP721RT.

INT **ControlFeature**(
INT *i_hdl*,
BYTE *i_feature*,
INT *i_arg1*,
INT *i_arg2*)

Parameters

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

i_feature
Printer function
Specifies the function of the printer to control.
See "6.7 Printer Function" for specifiable values.

i_arg1
Argument 1
Specifies the argument for the specified printer function.
See "6.7 Printer Function" for specifiable values.

i_arg2
Argument 2
Specifies the argument for the specified printer function.
See "6.7 Printer Function" for specifiable values.

Return value

On success: Returns 0.
On failure: Returns an error code.
See "5.1 Error Code List" for details of the error code.

Remarks

- When **LockPrinter** is being called by another process, calling this API fails.
- When the printer is not connected or in a communication disabled state, this API fails.

SendDataFile

Registers the contents of the command definition file in the internal memory of the SDK.

INT **SendDataFile**(
INT *i_hdl*,
LPCTSTR *i_fname*)

Parameters

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

i_fname
Command definition file name
Specifies the name of a command definition file created in the predefined format.

Return value

On success: Returns 0.

On failure: Returns an error code.
See "5.1 Error Code List" for details of the error code.

Remarks

- The command definition registered by this API is discarded by **CloseMonPrinter**.
- See "Chapter 7 Command Definition File" for details of the command definition file.

DirectSendRead

Sends transmission data corresponding to the command name specified in the command definition file that has been registered by **SendDataFile**.

INT **DirectSendRead**(
INT *i_hdl*,
LPCTSTR *i_cname*,
LPCTSTR *i_rtype*,
LPDWORD *io_rlen*,
LPBYTE *o_rbuf*,
DWORD *i_timeout*,
BOOL *i_flag*)

Parameters

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

i_cname
Command name
Specifies the command name registered by **SendDataFile**.

i_rtype

Receive data type name

Specifies the type of data to receive from the followings.
ASB: Stores only the response of the ASB setting command in the receive data.
Other: Stores responses other than the above from the printer in the receive data.

io_rlen

Size of data to receive

Specifies the maximum length of data to be received from the printer.
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 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 flag to specify the receive operation from the followings.
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 "5.1 Error Code List" for details of the error code.

Remarks

- In the data to send, do not include the ASB setting command that disables the response. The API that gets the printer status will not work properly.
- This API can be aborted by **Reset**.
- When **LockPrinter** is being called by another process, calling this API fails.
- When the printer is not connected or in a communication disabled state, this API fails.

GetPrintSmartLabelMode

Gets the paper when printing label file.

INT **GetPrintSmartLabelMode**(
INT *i_hdl*,
DWORD *o_id*)

Parameters

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

o_id
Paper ID data buffer
Specify the buffer to store the paper ID when printing label file.

Return value

On success: Returns 0.

On failure: Returns an error code.
See "5.1 Error Code List" for details of the error code.

SetPrintSmartLabelMode

Specifies the paper when printing label file.

INT **SetPrintSmartLabelMode**(
INT *i_hdl*,
DWORD *i_id*)

Parameters

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

i_id
Paper ID when printing label file
Specify the paper when printing label file.
See "6.8 Paper ID when Printing Label File" for specifiable values.
When an invalid value is specified, it is ignored.

Return value

On success: Returns 0.

On failure: Returns an error code.
See "5.1 Error Code List" for details of the error code.

Remarks

- This API setting is valid only when printing label file by **PrintSmartLabelImageData**.
- Paper ID specifying by this API is discarded by **CloseMonPrinter**.

SelectSmartLabelFile

Specifies a label file (*.sl) to use the printing label function.

INT **SelectSmartLabelFile**(
INT *i_hdl*,
LPCTSTR *i_fname*)

Parameters

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

i_fname
Label name (*.sl) to use
Specify the label file created using the app with the path.

Return value

On success: Returns 0.

On failure: Returns an error code.
See "5.1 Error Code List" for details of the error code.

Remarks

- Only 1 file can be selected with this API for 1 API handle.
- Make the file specified by this API readable by all processes that call the API.
- The file selection by this API is discarded by **CloseMonPrinter**.
- There are restrictions on the label files that can be used. See "Chapter 8 Printing Label Function" for details of the restrictions.

PrintSmartLabelImageData

Prints labels selected by **SelectSmartLabelFile**.

INT **PrintSmartLabelImageData**(
INT *i_hdl*,
DWORD *i_timeout*,
BYTE *i_option*)

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.

i_option

Reserved

Specify 0.

Return value

On success: Returns 0.

On failure: Returns an error code.
See "5.1 Error Code List" for details of the error code.

Remarks

- When replacement of data for each object fails, this API fails.
- When a label file where an object is not mapped is specified, this API fails.
- When **LockPrinter** is being called by another process, calling this API fails.
- When the printer is not connected or in a communication disabled state, this API fails.
- When printing for the first time after starting the computer, it may take some time before printing starts.
- See "Chapter 8 Printing Label Function" for details of the printing using the label file.

SetSmartLabelCharacterSet

Specifies the codepage of the text data to replace by **ReplaceSmartLabelTextDataA** or **ReplaceSmartLabelBarcodeDataA**.

INT **SetSmartLabelCharacterSet**(
INT *i_hdl*,
BYTE *i_charSet*)

Parameters

i_hdl

API handle

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

i_charSet

Codepage to use

i_charSet to specify is shown below.

Character Set	Character Code Table	*i_charSet*
System character set	(System Locale) Depends on the system	0
Thai	Thai (Code Page874)	1
Japanese	Japanese (Code Page932 / Shift-JIS)	2
Simplified Chinese	Simplified Chinese (Code Page936)	3
Korean	Korean (Code Page949)	4
Traditional Chinese	Traditional Chinese (Code Page950)	5
Central European	Central European (Code Page1250)	6
Cyrillic	Cyrillic (Code Page1251)	7

Character Set	Character Code Table	*i_charSet*
Latin	Latin (Code Page1252)	8
Greek	Greek (Code Page1253)	9
Turkish	Turkish (Code Page1254)	10
Hebrew	Hebrew (Code Page1255)	11
Arabic	Arabic (Code Page1256)	12
Baltic	Baltic (Code Page1257)	13
Vietnamese	Vietnamese (Code Page1258)	14

Return value

On success: Returns 0.

On failure: Returns an error code.
See "5.1 Error Code List" for details of the error code.

Remarks

- The codepage applicable APIs after executing this API are shown below.
  - **ReplaceSmartLabelTextDataA**
  - **ReplaceSmartLabelBarcodeDataA**
- When the codepage is not set with this API, it is encoded with "System character set".
- Codepage specification by this API is canceled by **CloseMonPrinter**.

ReplaceSmartLabelTextData

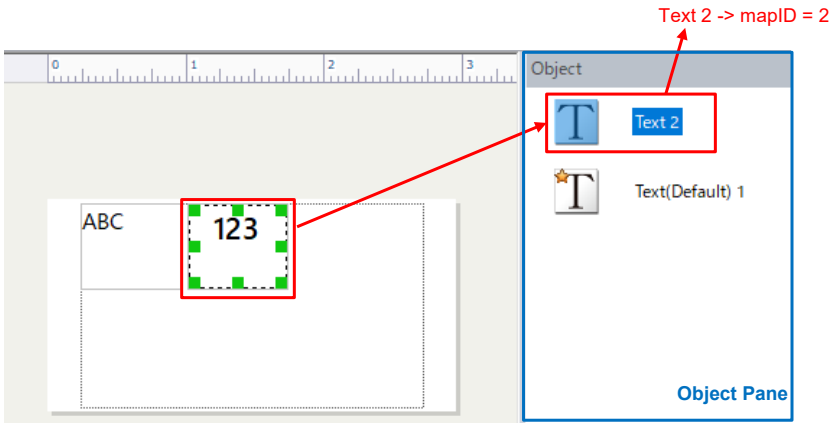
Replaces the value of the text object of the label file (*.sl) selected by **SelectSmartLabelFile**.

INT **ReplaceSmartLabelTextData**(
INT *i_hdl*,
INT *i_mapId*,
LPCTSTR *i_text*)

Parameters

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

i_mapId
ID of the text object
Specify the ID of the text object mapped on the label file (*.sl) of the app. The ID of the text object can be confirmed on the UI display of the app.



UI display of Smart Label Creator



UI display of SII Layout Editor

i_text

Text data to replace

Enter the text data to replace.
For the character string to be specified, be sure to terminate with NULL.
The maximum length of text data is 10000 characters including NULL.
If the text data is empty, the text data will not be printed in the text object specified by *i_mapId*.

Return value

On success: Returns 0.

On failure: Returns an error code.
See "5.1 Error Code List" for details of the error code.

Remarks

- When **SelectSmartLabelFile** is not called, this API fails.
- For **ReplaceSmartLabelTextDataA**, it is encoded with the language set by **SetSmartLabelCharacterSet**.
- See "Chapter 8 Printing Label Function" for details of the printing using the label file.

ReplaceSmartLabelImageData

Replaces the value of the image object of the label file (*.sl) selected by **SelectSmartLabelFile**.

INT **ReplaceSmartLabelImageData**(
INT *i_hdl*,
INT *i_mapId*,
LPCTSTR *i_fname*)

Parameters

i_hdl

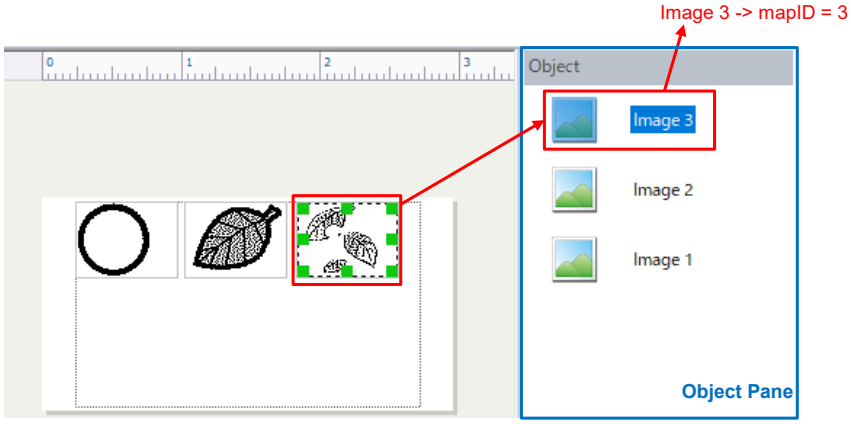
API handle

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

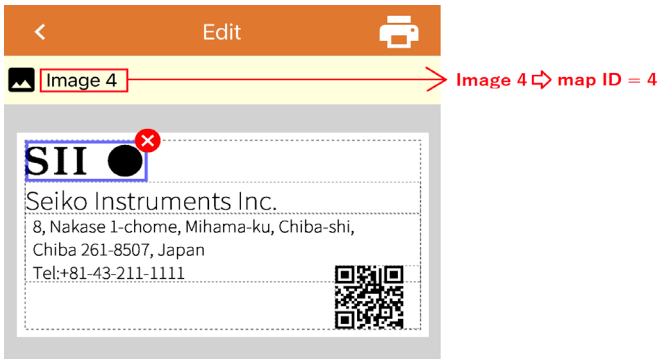
i_mapId

ID of the image object

Specify the ID of the image object mapped on the label file (*.sl) of the app. The ID of the image object can be confirmed on the UI display of the app.



UI display of Smart Label Creator



UI display of SII Layout Editor

i_fname

Image data to replace

Specify the file name of the image data to replace with the path.
Supported file extensions are .bmp, .gif, .jpg, .jpeg, .tiff, .tif, and .png.
However, even the supported extensions may not be registered depending on the format.

Return value

On success: Returns 0.

On failure: Returns an error code.
See "5.1 Error Code List" for details of the error code.

Remarks

- When **SelectSmartLabelFile** is not called, this API fails.
- Make the file specified by this API readable by all processes that call the API.
- See "Chapter 8 Printing Label Function" for details of the printing using the label file.

ReplaceSmartLabelBarcodeData

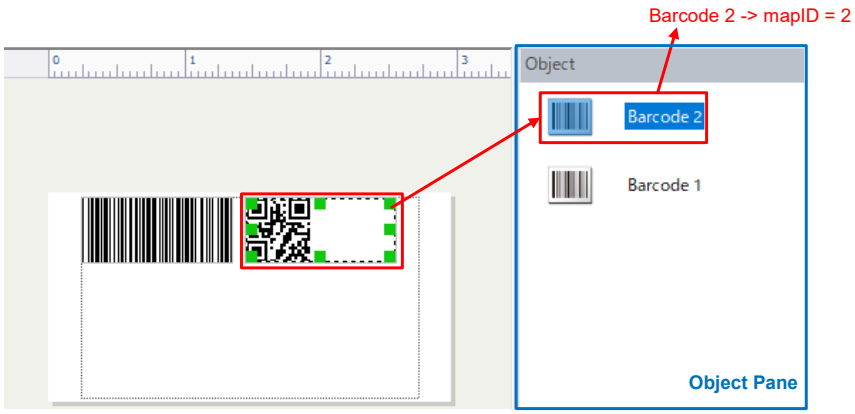
Replaces the value of the barcode object of the label file (*.sl) selected by **SelectSmartLabelFile**.

INT **ReplaceSmartLabelBarcodeData**(
INT *i_hdl*,
INT *i_mapId*,
LPCTSTR *i_barcodeData*)

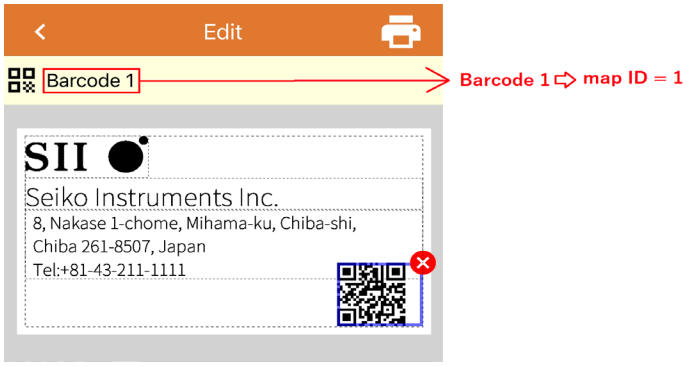
Parameters

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

i_mapId
ID of the barcode object
Specify the ID of the barcode object mapped on the label file (*.sl) of the app. The ID of the barcode object can be confirmed on the UI display of the app.



UI display of Smart Label Creator



UI display of SII Layout Editor

i_barcodeData

Barcode data to replace

Enter the barcode data to replace.
For the character string to be specified, be sure to terminate with NULL.
The maximum length of text data is 5000 characters including NULL. If barcode data exceeding 5000 characters is specified, characters after 5001 will be discarded.
If the barcode data is empty, the barcode data will not be printed in the barcode object specified by *i_mapId*.
Even if the character string to be replaced is invalid barcode data, an error is not caused. Make sure that the barcode data is valid before specifying it.

Return value

On success: Returns 0.

On failure: Returns an error code.
See "5.1 Error Code List" for details of the error code.

Remarks

- When **SelectSmartLabelFile** is not called, this API fails.
- For **ReplaceSmartLabelTextDataA**, it is encoded with the language set with **SetSmartLabelCharacterSet**.
- See "Chapter 8 Printing Label Function" for details of the printing using the label file.

GetProperty

Gets a part of the print settings.

INT **GetProperty**(
INT *i_hdl*,
LPDEVMODE *i_devmode*,
BYTE *i_pid*,
LPBYTE *o_dat*,
LPDWORD *io_size*)

Parameters

i_hdl

API handle

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

i_devmode

Devmode address

Specifies the Devmode address.

i_pid

Property ID

Specifies the property ID of the print settings to get.
See "6.8 Property ID" for specifiable values.

o_dat

Buffer of data to get

Specifies the buffer that stores the retrieved print settings.
When NULL is specified, this API fails and the specified data size of the print settings is stored in *io_size*.

io_size

Size of data to get

Specifies the buffer size to store the print settings.
When the control returns from the API, the data size of the retrieved print settings is stored.
When the specified buffer size is smaller than the data size of the print settings to get, this API fails and the specified data size of the print settings is stored.

Return value

On success: Returns 0.

On failure: Returns an error code.
See "5.1 Error Code List" for details of the error code.

SetProperty

Changes a part of the print settings.

INT **SetProperty**(
INT *i_hdl*,
LPDEVMODE *i_devmode*,
BYTE *i_pid*,
LPBYTE *i_dat*,
LPDWORD *i_size*)

Parameters

i_hdl

API handle

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

i_devmode

Devmode address

Specifies the Devmode address.

i_pid

Property ID

Specifies the property ID of the print settings to be changed.
See "6.8 Property ID" for specifiable values.

i_dat

Buffer of data to set

Specifies the buffer in which the print settings to be changed is stored.

i_size

Size of data to set

Specifies the buffer size in which the print settings to be changed is stored.

Return value

On success: Returns 0.

On failure: Returns an error code.
See "5.1 Error Code List" for details of the error code.

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.

- SlpRtCls.dll

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

4.3 Printer

4.3.1 API List

The APIs implemented in the .NET API are as follows.

- Namespace: SII.SDK.SlpRt
- Class name: StatusAPI

Common API

Category	API	Function Summary
Property	**LastError**	Gets the error value of the last executed API.
Property	**IsValid**	Gets the call status of **OpenMonPrinter**.
Method	**OpenMonPrinter**	Starts using the .NET API.
Method	**CloseMonPrinter**	Ends using the .NET API.

Specific API

Category	API	Function Summary
Property	**Status**	Gets the latest printer status.
Property	**PrintSmartLabelMode**	Specifies or gets the paper when printing label file.
Method	**LockPrinter**	Locks all data transmission and hardware reset requests from other processes to the printer.
Method	**UnlockPrinter**	Unlocks access prohibition (lock) from other processes by **LockPrinter**.
Method	**DirectIOEx**	Sends and receives binary data.
Method	**ResetPrinter**	Performs hardware reset of the printer.
Method	**SetStatusBack**	Starts the notification of the printer status by **StatusCallback** event.
Method	**CancelStatusBack**	Stops the notification of the printer status by **StatusCallback** event.
Method	**PowerOff**	Turns off the printer power.
Method	**GetCounter**	Gets the maintenance counter.
Method	**ResetCounter**	Initializes the maintenance counter.
Method	**GetType**	Gets various IDs of the printer.
Method	**GetPrnCapability**	Gets the printer information.
Method	**OpenDrawer**	Drives the specified drawer. This API is supported only by SLP721RT.
Method	**ControlFeature**	Controls functions of the printer other than printing. This API is supported only by SLP721RT.

Category	API	Function Summary
Method	**SendDataFile**	Registers the contents of the command definition file to the memory in the SDK.
Method	**DirectSendRead**	Executes the command definition registered by **SendDataFile**.
Method	**SelectSmartLabelFile**	Specifies a label file (*.sl) to use the printing label function.
Method	**PrintSmartLabelImageData**	Prints labels selected by **SelectSmartLabelFile**.
Method	**ReplaceSmartLabelTextData**	Replaces the value of the text object of the label file (*.sl) selected by **SelectSmartLabelFile**.
Method	**ReplaceSmartLabelImageData**	Replaces the value of the image object of the label file (*.sl) selected by **SelectSmartLabelFile**.
Method	**ReplaceSmartLabelBarcodeData**	Replaces the value of the barcode object of the label file (*.sl) selected by **SelectSmartLabelFile**.
Method	**GetProperty**	Gets a part of the print settings.
Method	**SetProperty**	Changes a part of the print settings.
Event	**StatusCallback**	Notifies the returned printer status.

4.3.2 Common Property

LastError

Gets the error value of the last called API.

ErrorCode LastError { get; }

Initial value

SUCCESS

Remarks

- See "5.1 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.

4.3.3 Common Method

OpenMonPrinter

Starts using the .NET API.

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

Parameters

type
Open type
OpenType.TYPE_PRINTER (fixed)

name
Name of the printer that uses the .NET API
Specifies the printer name (friendly name).

Return value

Returns an error code. See "5.1 Error Code List" for details of the error code.

Remarks

- It can be opened up to 8 instances at the same time within 1 process combining each class in .NET API.
- When the .NET API is no longer used, be sure to call **CloseMonPrinter**.
- Set the connection destination of the printer driver to USB or TCP/IP.
- This API succeeds even when the printer is not connected or the printer power is turned off.

CloseMonPrinter

Ends using the .NET API.

ErrorCode **CloseMonPrinter**()

Return value

Returns an error code. See "5.1 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.

4.3.4 Specific Property

Status

Gets the latest printer status.

ASB Status { get; }

Initial value

ASB_NO_RESPONSE

Remarks

- When reconnection to the printer is detected, the printer status is the status received last at the time.
- When IsValid is FALSE, a correct value is not available.
- See "6.1 Printer Status List" for details of the printer status.

PrintSmartLabelMode

Specifies or gets the paper when printing label file.

PrintSmartLabelMode { set; get; }

Initial value

MODE_MARK

Remarks

- See "6.8 Paper ID when Printing Label File" for specifiable values.
- When IsValid is FALSE, a correct value is not available.

4.3.5 Specific Method

LockPrinter

Locks all data transmission and hardware reset requests from other processes to the printer.

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, time is set to 90000 ms.

Return value

Returns an error code. See "5.1 Error Code List" for details of the error code.

Remarks

- By this API, multiple locking is possible up to 99 times. In order to release the lock, call **UnlockPrinter** the same number of times as for this API.
- For using the TCP/IP connection, release the lock before the time of no transmission exceeds the TCP/IP receive timeout period. If it is locked, it may cause the data to be missed or the data sent from other hosts to interrupt the printer.
  For the receive timeout period, see "LAN communication settings" in "3.2.3 Setting Screen for Each Connection Method" stated in the "SII Communication Setting Utility for Windows User's Guide" for SLP720RT/SLP721RT series.

UnlockPrinter

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

ErrorCode **UnlockPrinter**()

Return value

Returns an error code. See "5.1 Error Code List" for details of the error code.

Remarks

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

DirectIOEx

With the syntax (a), the API gets the receive data as binary data from the printer after sending binary data.
With the syntax (b), the API gets the receive data as string data from the printer after sending binary data.
With the syntax (c), the API sends 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 send
Specifies the buffer in which the data to send 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.
When the control returns from the API, the received data size is stored.

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 flag to specify the receive operation from the following.
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 followings.
0: Gets the data excluding the response of the ASB setting command.
1: Gets the data including the response of the ASB setting command.

Return value

Returns an error code. See "5.1 Error Code List" for details of the error code.

Remarks

- When this API succeeds, *data* in syntax (a) is resized to the actual receive data size within the limits of specified receive data size as *data*.
- In the data to send, do not include the ASB setting command that disables the response. The API that gets the printer status will not work properly.
- This API can be aborted by **ResetPrinter**.
- When 0x02 is included in the receive data, it is converted to 0x5f.

ResetPrinter

Performs a hardware reset of the printer.

ErrorCode **ResetPrinter**()

Return value

Returns an error code. See "5.1 Error Code List" for details of the error code.

Remarks

- Performs hardware reset of the printer using the communication protocol (without using printer commands).
- After called this API, wait a few seconds to send the data. If data transmission is performed right after calling this API, it may cause the data to be missed.
- When this API is called, the following APIs are aborted.
  - **DirectIOEx**
  - **DirectSendRead**
- During the call of this API, the printer status response will be "No response". See "6.1 Printer Status List" for details of the printer status.

SetStatusBack

Starts the notification of the printer status by **StatusCallback** event.

ErrorCode **SetStatusBack**()

Return value

Returns an error code. See "5.1 Error Code List" for details of the error code.

Remarks

- When the notification of the printer status is started by this API, **StatusCallback** event is raised with the current printer status.
- When the event handler is not registered on **StatusCallback** event, this API fails.
- Even when **StatusCallback** event is already started and this API is called again, the immediate response of the printer status is performed.
- **StatusCallback** event started by this API is stopped by the following APIs:
  - **CloseMonPrinter**
  - **CancelStatusBack**
- The time from receiving the printer status to raising the event is not guaranteed.
- See "6.1 Printer Status List" for details of the printer status.

CancelStatusBack

Stops the notification of the printer status by **StatusCallback** event.

ErrorCode **CancelStatusBack**()

Return value

Returns an error code. See "5.1 Error Code List" for details of the error code.

Remarks

- This API succeeds even when the notification of the printer status has not been started by **SetStatusBack**.

PowerOff

Turns off the printer power.

ErrorCode **PowerOff**()

Return value

Returns an error code. See "5.1 Error Code List" for details of the error code.

Remarks

- When this API is executed, processing accompanying power-off is performed on the printer.

GetCounter

Gets the maintenance counter.
There are 2 types of syntax to get the maintenance counter: the syntax that uses the counter ID defined in CounterIndex, and the syntax that specifies it as a numerical value.

With the syntax (a), the API gets the counter value using the counter ID defined in CounterIndex.
With the syntax (b), the API gets the counter value by specifying the counter ID as a numerical value.

(a) ErrorCode **GetCounter**(
CounterIndex *index*,
bool *type*,
out int *data*)

(b) ErrorCode **GetCounter**(
byte *index*,
out int *data*)

Parameters

index
Counter ID
Specifies the counter ID of the maintenance counter to get.
Specify (a) when using the counter ID defined in CounterIndex.
Specify (b) when specifying the counter ID as a numerical value.
See "6.2 Counter ID" for specifiable values.

type
Maintenance counter type
Specifies the type of the maintenance counter to get from the following.
TRUE: Integration counter
FALSE: Initializable counter

data
Counter variable
Specifies the variable to store the retrieved counter value.

Return value

Returns an error code. See "5.1 Error Code List" for details of the error code.

ResetCounter

Initializes the maintenance counter.
There are 2 types of syntax to initialize the maintenance counter: the syntax that uses the counter ID defined in CounterIndex, and the syntax that specifies it as a numerical value.

With the syntax (a), the API initializes the counter value using the counter ID defined in CounterIndex.
With the syntax (b), the API initializes the counter value by specifying the counter ID as a numerical value.

(a) ErrorCode **ResetCounter**(
CounterIndex *index*)

(b) ErrorCode **ResetCounter**(
byte *index*)

Parameters

index
Counter ID
Specifies the counter ID of the maintenance counter to initialize the counter value.
Specify (a) when using the counter ID defined in CounterIndex.
Specify (b) when specifying the counter ID as a numerical value.
See "6.2 Counter ID" for specifiable values.

Return value

Returns an error code. See "5.1 Error Code List" for details of the error code.

Remarks

- Confirm with **GetCounter** that the value of the counter ID specified by this API has been initialized.

GetType

Gets various IDs of the printer.

ErrorCode **GetType**(
out byte *typeId*,
out byte *fontID*,
out byte *exrom*,
out byte *special*)

Parameters

typeId
ID1
3 is stored.

fontID
ID2
2 is stored.

exrom
Reserved
Specify null.
special
Reserved
Specify null.

Return value

Returns an error code. See "5.1 Error Code List" for details of the error code.

GetPrnCapability

Gets the printer information.
There are 2 types of syntax to get the printer information: the syntax that uses the printer ID defined in PrinterId, and the syntax that specifies it as a numerical value.

With the syntax (a), the API gets the printer information as binary data using the printer ID defined in PrinterId.

With the syntax (b), the API gets the printer information as string data using the printer ID defined in PrinterId.

With the syntax (c), the API gets the printer information as binary data by specifying the printer ID as a numerical value.

With the syntax (d), the API gets the printer information as string data by specifying the printer ID as a numerical value.

(a) ErrorCode **GetPrnCapability**(
PrinterId *id*,
out byte[] *data*)

(b) ErrorCode **GetPrnCapability**(
PrinterId *id*,
out string *data*)

(c) ErrorCode **GetPrnCapability**(
byte *id*,
out byte[] *data*)

(d) ErrorCode **GetPrnCapability**(
byte *id*,
out string *data*)

Parameters

id

Printer ID

Specifies the printer ID of the printer information to get.
Specify (a) or (b) when using the printer ID defined in PrinterId. (a) gets the response format of the printer information as a binary code, and (b) gets it as an ASCII character string.
Specify (c) or (d) when specifying the printer ID as a numerical value. (c) gets the response format of the printer information as a binary code, and (d) gets it as an ASCII character string.
See "6.3 Printer ID" for specifiable values.

data

Buffer of data to receive

Specifies the buffer that stores the retrieved printer information.

Return value

Returns an error code. See "5.1 Error Code List" for details of the error code.

OpenDrawer

Drives the specified drawer.
Supported only by SLP721RT.

ErrorCode **OpenDrawer**(
Drawer *drawer*,
Pulse *pulse*)

Parameters

drawer

Target drawer

Specifies the target drawer to drive.
See "6.5 Target Drawers" for specifiable values.

pulse

Drawer drive time

Specifies ON/OFF time of the drawer driving pulse. The ON time and the OFF time are specified as the same.
See "6.6 Drawer Drive Time" for specifiable values.

Return value

Returns an error code. See "5.1 Error Code List" for details of the error code.

Remarks

- As for the drawer control time, follow the specifications of your drawer.

ControlFeature

Controls functions of the printer other than printing.
Supported only by SLP721RT.

ErrorCode **ControlFeature**(
Feature *feature*,
int *arg1*,
int *arg2*)

Parameters

feature
Printer function
Specifies the function of the printer to control.
See "6.7 Printer Function" for specifiable values.

arg1
Argument 1
Specifies the argument for the specified printer function.
See "6.7 Printer Function" for specifiable values.

arg2
Argument 2
Specifies the argument for the specified printer function.
See "6.7 Printer Function" for specifiable values.

Return value

Returns an error code. See "5.1 Error Code List" for details of the error code.

SendDataFile

Registers the contents of the command definition file in the internal memory of the SDK.

ErrorCode **SendDataFile**(
string *fileName*)

Parameters

fileName
Command definition file name
Specifies the name of a command definition file created in the predefined format.

Return value

Returns an error code. See "5.1 Error Code List" for details of the error code.

Remarks

- The command definition registered by this API is discarded by **CloseMonPrinter**.
- See "Chapter 7 Command Definition File" for details of the command definition file.

DirectSendRead

Sends transmission data corresponding to the command name specified in the command definition file that has been registered by **SendDataFile**.

With the syntax (a), the API gets the receive data as binary data from the printer after sending the transmission data corresponding to the command name specified in the command definition file that has been registered by **SendDataFile**.

With the syntax (b), the API gets the receive data as string data from the printer after sending the transmission data corresponding to the command name specified in the command definition file that has been registered by **SendDataFile**.

With the syntax (c), the API sends the transmission data corresponding to the command name specified in the command definition file that has been registered by **SendDataFile**.

(a) ErrorCode **DirectSendRead**(
string *cmdName*,
string *readType*,
ref byte[] *data*,
int *timeout*,
bool *readFlag*)

(b) ErrorCode **DirectSendRead**(
string *cmdName*,
string *readType*,
out *string data*,
int *timeout*)

(c) ErrorCode **DirectSendRead**(
string *cmdName*,
string *readType*,
int *timeout*)

Parameters

cmdName
Command name
Specifies the command name defined by **SendDataFile**.

readType
Receive data type name
Specifies the type of data to receive from the followings.
ASB: Stores only the response of the ASB setting command in the receive data.
Other: Stores responses other than the above from the printer in the receive data.

data
Buffer of data to receive
Specifies the buffer that stores the data to get.

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 flag to specify the receive operation from the followings.
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

Returns an error code. See "5.1 Error Code List" for details of the error code.

Remarks

- This API can be aborted by **ResetPrinter**.
- When 0x02 is included in the receive data, it is converted to 0x5f.

SelectSmartLabelFile

Specifies a label file (*.sl) to use the printing label function.

ErrorCode **SelectSmartLabelFile**(
string *fileName*)

Parameters

fileName
Label name (*.sl) to use
Specify the file name of the label file created using the app.

Return value

Returns an error code. See "5.1 Error Code List" for details of the error code.

Remarks

- Only 1 file can be selected with this API for 1 instance.
- Make the file specified by this API readable by all processes that call the API.
- The file selection by this API is discarded by **CloseMonPrinter**.
- There are restrictions on the label files that can be used. See "Chapter 8 Printing Label Function" for details of the restrictions.

PrintSmartLabelImageData

Prints labels selected by **SelectSmartLabelFile**.

ErrorCode **PrintSmartLabelImageData**(
int *timeout*,
byte *option*)

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.

option

Reserved

Specify 0.

Return value

Returns an error code. See "5.1 Error Code List" for details of the error code.

Remarks

- When replacement of data for each object fails, this API fails.
- When a label file where an object is not mapped is specified, this API fails.
- When printing for the first time after starting the computer, it may take some time before printing starts.
- See "Chapter 8 Printing Label Function" for details of the printing using the label file.

ReplaceSmartLabelTextData

Replaces the value of the text object of the label file (*.sl) selected by **SelectSmartLabelFile**.

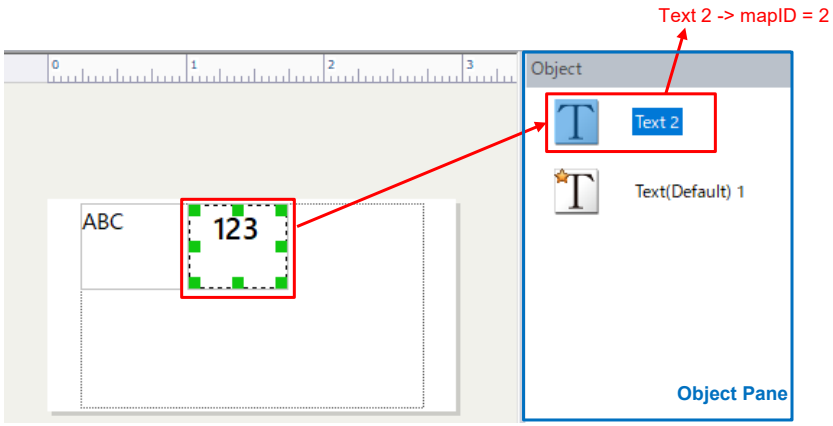
ErrorCode **ReplaceSmartLabelTextData**(
int *mapId*,
string *text*)

Parameters

mapId

ID of the text object

Specify the ID of the text object mapped on the label file (*.sl) of the app. The ID of the text object can be confirmed on the UI display of the app.



UI display of Smart Label Creator



UI display of SII Layout Editor

text

Text data to replace

Enter the text data to replace.
For the character string to be specified, be sure to terminate with NULL.
The maximum length of text data is 10000 characters including NULL.
If the text data is empty, the text data will not be printed in the text object specified by *mapId*.

Return value

Returns an error code. See "5.1 Error Code List" for details of the error code.

Remarks

- When **SelectSmartLabelFile** is not called, this API fails.
- See "Chapter 8 Printing Label Function" for details of the printing using the label file.

ReplaceSmartLabelImageData

Replaces the value of the image object of the label file (*.sl) selected by **SelectSmartLabelFile**.

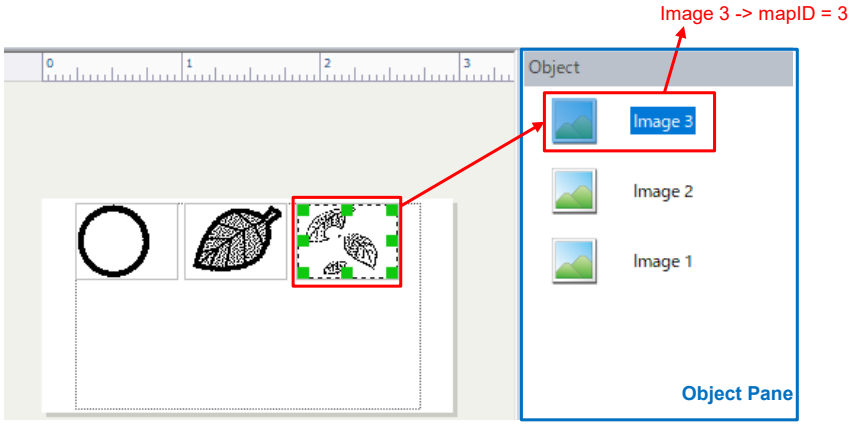
ErrorCode **ReplaceSmartLabelImageData**(
int *mapId*,
string *fileName*)

Parameters

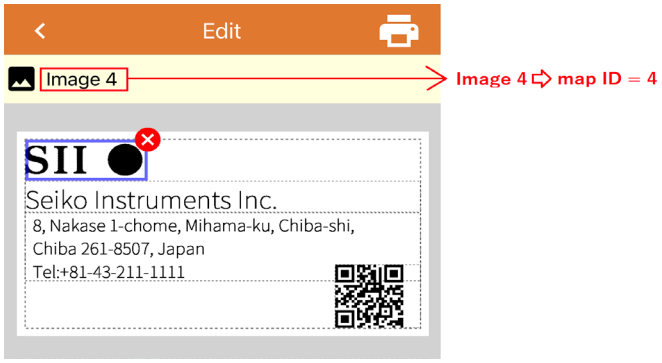
mapId

ID of the image object

Specify the ID of the image object mapped on the label file (*.sl) of the app. The ID of the image object can be confirmed on the UI display of the app.



UI display of Smart Label Creator



UI display of SII Layout Editor

fileName

Image data to replace

Specify the file name of the image data to replace.
Supported file extensions are .bmp, .gif, .jpg, .jpeg, .tiff, .tif, and .png.
However, even the supported extensions may not be registered depending on the format.

Return value

Returns an error code. See "5.1 Error Code List" for details of the error code.

Remarks

- When **SelectSmartLabelFile** is not called, this API fails.
- Make the file specified by this API readable by all processes that call the API.
- See "Chapter 8 Printing Label Function" for details of the printing using the label file.

ReplaceSmartLabelBarcodeData

Replaces the value of the barcode object of the label file (*.sl) selected by **SelectSmartLabelFile**.

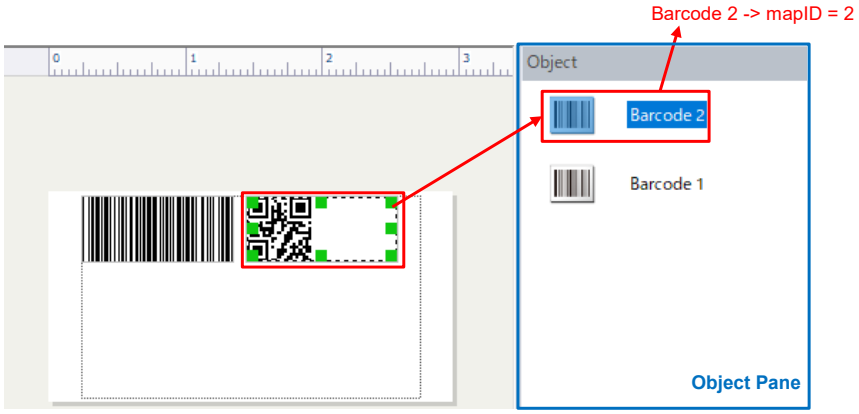
ErrorCode **ReplaceSmartLabelBarcodeData**(
int *mapId*,
string *barcodeData*)

Parameters

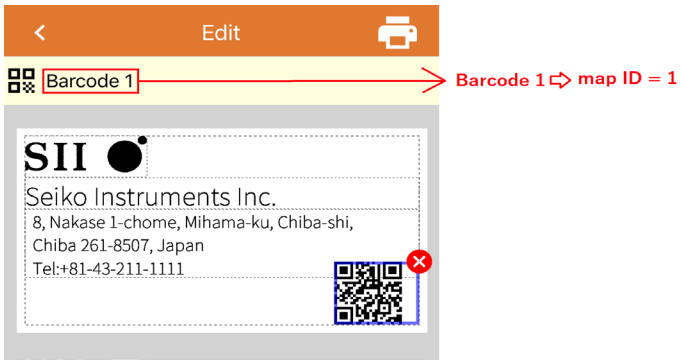
mapId

ID of the barcode object

Specify the ID of the barcode object mapped on the label file (*.sl) of the app. The ID of the barcode object can be confirmed on the UI display of the app.



UI display of Smart Label Creator



UI display of SII Layout Editor

barcodeData

Barcode data to replace

Enter the barcode data to replace.
For the character string to be specified, be sure to terminate with NULL.
The maximum length of text data is 5000 characters including NULL. If barcode data exceeding 5000 characters is specified, characters after 5001 will be discarded.
If the barcode data is empty, the barcode data will not be printed in the barcode object specified by *mapId*.
Even if the character string to be replaced is invalid barcode data, an error is not caused. Make sure that the barcode data is valid before specifying it.

Return value

Returns an error code. See "5.1 Error Code List" for details of the error code.

Remarks

- When **SelectSmartLabelFile** is not called, this API fails.
- See "Chapter 8 Printing Label Function" for details of the printing using the label file.

GetProperty

Gets a part of the print settings.
There are 2 types of syntax to get the print settings: the syntax that uses the property ID defined in PropertyId, and the syntax that specifies it as a numerical value.

With the syntax (a), the API gets a part of the print settings of the property using the property ID defined in PropertyId.
With the syntax (b), the API gets a part of the print settings of the property by specifying the property ID as a numerical value.

(a) ErrorCode **GetProperty**(
IntPtr *devmode*,
PropertyId *id*,
byte[] *data*,
ref uint *size*)

(b) ErrorCode **GetProperty**(
IntPtr *devmode*,
byte *id*,
byte[] *data*,
ref uint *size*)

Parameters

devmode
DevMode address
Specifies the Devmode address.

id
Property ID
Specifies the property ID of the print settings to get.
Specify (a) when using the property ID defined in PropertyId.
Specify (b) when specifying the property ID as a numerical value.
See "6.8 Property ID" for specifiable values.

data
Buffer of data to get
Specifies the buffer that stores the retrieved print settings.

size
Size of data to get
Specifies the buffer size to store the print settings.
When the control returns from the API, the data size of the retrieved print settings is stored.

Return value

Returns an error code. See "5.1 Error Code List" for details of the error code.

SetProperty

Changes a part of the print settings.
There are 2 types of syntax to change the print settings: the syntax that uses the property ID defined in PropertyId, and the syntax that specifies it as a numerical value.

With the syntax (a), the API changes a part of the print settings using the property ID defined in PropertyId.
With the syntax (b), the API changes a part of the print settings by specifying the property ID as a numerical value.

(a) ErrorCode **SetProperty**(
IntPtr *devmode*,
PropertyId *id*,
byte[] *data*,
ref uint *size*)

(b) ErrorCode **SetProperty**(
IntPtr *devmode*,
byte *id*,
byte[] *data*,
ref uint *size*)

Parameters

devmode
DevMode address
Specifies the Devmode address.

id
Property ID
Specifies the property ID of the print settings to be changed.
Specify (a) when using the property ID defined in PropertyId.
Specify (b) when specifying the property ID as a numerical value.
See "6.8 Property ID" for specifiable values.

data
Buffer of data to set
Specifies the buffer in which the print settings to be changed is stored.

size
Size of data to set
Specifies the buffer size in which the print settings to be changed is stored.

Return value

Returns an error code. See "5.1 Error Code List" for details of the error code.

4.3.6 Event

StatusCallback

Notifies the returned printer status.

event StatusCallbackHandler **StatusCallback**

delegate void **StatusCallbackHandler**(
ASB *status*)

Parameters

status
Printer status variable
The printer status is stored.
See "6.1 Printer Status List" for details of the printer status.

Remarks

- To start the notification of the printer status, call **SetStatusBack**.
- When the notification of the printer status is started by **SetStatusBack**, the event is raised with the current printer status.
- Register the event handler to handle the response printer status before starting the notification of the printer status.
- When the event handler is not registered, **SetStatusBack** fails.
- Even when the printer status is received, the event is not raised when the printer status has not changed from when it was last received.
- When reconnection to the printer is detected, the printer status is the status received last at the time.
- Call the following APIs to stop this API.
  - **CloseMonPrinter**
  - **CancelStatusBack**
- The following APIs cannot be called from the registered event handler.
  - **CloseMonPrinter**
  - **SetStatusBack**
  - **CancelStatusBack**
- The time from receiving the printer status to raising the event is not guaranteed.
- See "6.1 Printer Status List" for details of the printer status.

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. ・Printer or file cannot be accessed.
ERR_PARAM	-90	Parameter is incorrect.
ERR_NOT_SUPPORT	-100	Function is not supported.
ERR_OFFLINE	-110	Printer is disconnected or offline.
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_IMAGE_FAILED	-250	Label image has been failed to create.
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.

Macro Definition (Constant)	Value	Description
ERR_SPL_NOT_EXIST	-350	Spooler service or Smart Label Printer RT series service has not been started.
ERR_LOCKED	-1000	Printer is locked.
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 Printer Status List

The corresponding bits of the printer status responses are as follows. In addition, please refer to the printer status processing of our sample program.

Printer Status	Corresponding Bit		Description
Voltage error	ASB_VP_ERR	0x00000000	No voltage error
		0x00000001	Voltage error
Hardware error	ASB_HARDWARE_ERR	0x00000000	No hardware error
		0x00000002	Hardware error
Head temperature error	ASB_HEAD_TEMPERATUR_ERR	0x00000000	No head temperature error
		0x00000004	Head temperature error
Cutter error	ASB_AUTOCUTTER_ERR	0x00000000	No cutter error
		0x00000008	Cutter error
Out-of-paper error	ASB_RECEIPT_END	0x00000000	No out-of-paper error
		0x00000010	Out-of-paper error
Paper jam error while detecting mark	ASB_MARK_PAPER_JAM_ERR	0x00000000	No paper jam error
		0x00000040	Paper jam error
Cover open error	ASB_COVER_OPEN	0x00000000	Paper cover is closed
		0x00000080	Paper cover is open
FEED switch status	ASB_PAPER_FEED	0x00000000	FEED switch status is "Off"
		0x00000100	FEED switch status is "On"
Paper feed status	ASB_NOW_PRINTING	0x00000000	Stop
		0x00000400	Operating
Return-waiting status	ASB_RETURN_WAITING	0x00000000	-
		0x00000800	Recovery waiting state
Taken sensor status	ASB_TAKEN_SENSOR	0x00000000	Paper removed
		0x00002000	Paper removal waiting

Printer Status	Corresponding Bit		Description
Drawer switch input status	ASB_DRAWER_KICK	0x00000000	Low*1
		0x00008000	High
Rewriting FLASH memory	ASB_FLASH_MEMORY_REWRITING	0x00000000	-
		0x00010000	FLASH memory rewriting
Automatic recovery error	ASB_AUTORECOVER_ERR*2	0x00000000	No automatic recovery error
		0x20000000	Automatic recovery error
Unrecoverable error	ASB_UNRECOVER_ERR*2	0x00000000	No unrecoverable error
		0x40000000	Unrecoverable error
No response	ASB_NO_RESPONSE*2	0x00000000	Printer responds
		0x80000000	Not connected or communication error

*1: Low is fixed in SLP720RT.
*2: It is an extended status for the response of the ASB setting command.

6.2 Counter ID

The counter IDs and descriptions are as follows.

Counter ID		Description	Initialization by ResetCounter
.NET API	.NET API Win32 API		
ROLL_FEED_LINES	20	Number of paper feed lines	Enable
ROLL_HEAD_CHARGE	21	Number of thermal head activation times	Enable
PAPER CUT	50	Number of autocutter drive times	Enable
OPERATION_TIME	70	Drive time of printer	Enable
ROLL_FEED_LINES*1	148	Number of paper feed lines (integrated)	Disable
ROLL_HEAD_CHARGE*1	149	Number of thermal head activation times (integrated)	Disable
PAPER CUT*1	178	Number of autocutter drive times (integrated)	Disable
OPERATION_TIME*1	198	Drive time of printer (integrated)	Disable

*1: For the .NET API, specify *type* of **GetCounter** to get the counter value of the integration counter. See **GetCounter** of .NET API for details of the API.

6.3 Printer ID

The printer IDs and descriptions are as follows.

Printer ID		Description	Response Format
.NET API	.NET API Win32 API		
DEVICE	1	Printer model ID	Numerical value (1 byte) 0x28
TYPE	2	Type ID	Numerical value (1 byte) See "6.4 Type ID"
ROM_VER	3	ROM version ID	Numerical value (1 byte)
FW_VER_MAIN	65	Firmware version (main)	ASCII string "x.xx.xx"
MFG	66	Manufacturer	ASCII string "Seiko Instruments Inc."
MDL	67	Model name	ASCII string "SII SLP720RT Series."
FONT	69	Multi-language font type	ASCII string "KANJI JAPANESE"
FW_VER_BOOT	97	Firmware version (boot)	ASCII string "x.xx.xx"
FW_SUM_BOOT	98	Firmware checksum (boot)	Numerical value (2 bytes)
FW_SUM_MAIN	99	Firmware checksum (main)	Numerical value (2 bytes)
FW_SUM	100	Firmware checksum (main + boot)	Numerical value (2 bytes)

6.4 Type ID

The corresponding bits of responses for the type ID and descriptions are as follows.
The type ID is a part of the printer information retrieved by the printer command "Send Printer ID".

Corresponding Bit	Description	
0x01	Paper width selection	0: 58 mm 1: 80 mm
0x02	Reserved	Fixed to 1
0x04	SLP720RT (Reserved) SLP721RT (Reserved)	Fixed to 0 Fixed to 1
0x08	Undefined	Fixed to 0
0x10	Reserved	0: No 1: Yes
0x20	Reserved	0: No 1: Yes
0x40	Reserved	Fixed to 0
0x80	Undefined	Fixed to 0

The retrievable response is the sum of the above values.

Example: Response value = 0x13

6.5 Target Drawers

The target drawers and descriptions are as follows.

Target Drawers		Description
.Net API	.NET API Win32 API	
DRAWER_1	1	Drives drawer 1
DRAWER_2	2	Drives drawer 2

6.6 Drawer Drive Time

The drawer drive time and descriptions are as follows.

Drawer Drive Time		Description
.NET API	.NET API Win32 API	
TIME_100	1	Drives drawer for 100 ms
TIME_200	2	Drives drawer for 200 ms
TIME_300	3	Drives drawer for 300 ms
TIME_400	4	Drives drawer for 400 ms
TIME_500	5	Drives drawer for 500 ms
TIME_600	6	Drives drawer for 600 ms
TIME_700	7	Drives drawer for 700 ms
TIME_800	8	Drives drawer for 800 ms

As for the drawer control time, follow the specifications of your drawer.

6.7 Printer Function

The printer function and descriptions are as follows.

Printer Function		Description	Argument 1	Argument 2
.NET API	.NET API Win32 API			
EXT_BUZZER	1	Sounds external buzzer	Buzzer pattern 0: Pattern 1 1: Pattern 2 2: Pattern 3 3: Pattern 4	Buzzer frequency*1 1 to 255 (times)

*1 : The external buzzer sounds will stop under one of the following conditions:
- Sounding for the set number of buzzers
- Opening the cover
- Executing the printer command "Stop External Buzzer"

6.8 Paper ID when Printing Label File

The paper IDs when printing label file and descriptions are as follows.

Paper selection with or without mark		Description
.NET API	.NET API Win32 API	
MODE_MARK	0	Feeds the paper to the print start position with feeding the paper backward at **PrintSmartLabelImageData** execution.
MODE_NONE_MARK	1	The paper is not fed to the print start position at **PrintSmartLabelImageData** execution.

6.9 Property ID

Caution

◆ When calling **SetProperty**, specify the data size to 1 byte except for the logo keycode and custom commands.

The property IDs and descriptions are as follows.
Inside [ ] the parentheses indicate the timing when the processing is performed in option settings. See "3.3.7 Setting of Option" in "SII Printer Driver for Windows User's Guide" for SLP720RT/SLP721RT series on the option settings.

Property ID		Description	
.NET API	.NET API Win32 API		
INIT	1	Initialize	0: Enabled 1: Disabled
SPEED	2	Speed	0: High[*1] 1: Middle[*1]
MARGIN	3	Margin	0: Minimum margin 1: Minimum top margin 2: Minimum bottom margin 3: Maximum margin
DENSITY	4	Density (percentage)	70 to 130
DIRECTION	5	Direction	0: Forward 1: Backward
REDUCTION	6	Reduction	0: None 15 to 100: Scale (percentage)

Property ID		Description	
.NET API	.NET API Win32 API		
CUT	7	Paper cut	0: No cut 1: Full cut (by jobs) 2: Partial cut (by jobs) 3: Full cut (by pages) 4: Partial cut (by pages) 5: Partial cut (between pages)
MARK_FEED	8	Marked paper form feed	0: No form feed 1: By pages 2: By jobs
CUT_FEED	9	Feed to cut position	0: Enabled 1: Disabled
START_DOC_LOGO	10	[Print Start] Logo	0: None 1: Left 2: Center 3: Right
START_DOC_LOGO_KEY	11	[Print Start] Logo keycode (2 bytes)*2	Each byte 0x20 to 0x39
START_DOC_DRAWER	12	[Print Start] Drawer*3	0: No use 1: Use drawer 1 2: Use drawer 2
START_DOC_DRAWER_ON	13	[Print Start] ON time (x 2 ms)*4	1 to 255
START_DOC_DRAWER_OFF	14	[Print Start] OFF time (x 2 ms)	1 to 255
START_DOC_CMD	15	[Print Start] Custom command (128 bytes)	Command data
START_DOC_BACK_FEED	16*5	[Print Start] Paper feed (backfeed) (dot)	0 to -68
START_DOC_FEED	17*5	[Print Start] Paper feed (feed) (dot)	0 to 255
START_DOC_EXT_BUZZER	110	[Print Start] External buzzer	0: Pattern 1 1: Pattern 2 2: Pattern 3 3: Pattern 4 255: OFF
START_DOC_EXT_BUZZER_RPT	111	[Print Start] External buzzer Frequency*6	1 to 255
START_PAGE_LOGO	20	[Page Start] Logo	0: None 1: Left 2: Center 3: Right

Property ID		Description	
.NET API	.NET API Win32 API		
START_PAGE_LOGO_KEY	21	[Page Start] Logo keycode (2 bytes)*2	Each byte 0x20 to 0x39
START_PAGE_CMD	25	[Page Start] Custom command (128 bytes)	Command data
START_PAGE_BACK_FEED	26*5	[Page Start] Paper feed (backfeed) (dot)	0 to -68
START_PAGE_FEED	27*5	[Page Start] Paper feed (feed) (dot)	0 to 255
START_PAGE_EXT_BUZZER	120	[Page Start] External buzzer	0: Pattern 1 1: Pattern 2 2: Pattern 3 3: Pattern 4 255: OFF
START_PAGE_EXT_BUZZER_RPT	121	[Page Start] External buzzer Frequency*6	1 to 255
END_PAGE_LOGO	30	[Page End] Logo	0: None 1: Left 2: Center 3: Right
END_PAGE_LOGO_KEY	31	[Page End] Logo keycode (2 bytes)*2	Each byte 0x20 to 0x39
END_PAGE_CMD	35	[Page End] Custom command (128 bytes)	Command data
END_PAGE_BACK_FEED	36*5	[Page End] Paper feed (backfeed) (dot)	0 to -68
END_PAGE_FEED	37*5	[Page End] Paper feed (feed) (dot)	0 to 255
END_PAGE_EXT_BUZZER	130	[Page End] External buzzer	0: Pattern 1 1: Pattern 2 2: Pattern 3 3: Pattern 4 255: OFF
END_PAGE_EXT_BUZZER_RPT	131	[Page End] External buzzer Frequency*6	1 to 255

Property ID		Description	
.NET API	.NET API Win32 API		
END_DOC_LOGO	40	[Print End] Logo	0: None 1: Left 2: Center 3: Right
END_DOC_LOGO_KEY	41	[Print End] Logo keycode (2 bytes)*2	Each byte 0x20 to 0x39
END_DOC_DRAWER	42	[Print End] Drawer*3	0: No use 1: Use drawer 1 2: Use drawer 2
END_DOC_DRAWER_ON	43	[Print End] ON time (x 2 ms)*4	1 to 255
END_DOC_DRAWER_OFF	44	[Print End] OFF time (x 2 ms)	1 to 255
END_DOC_CMD	45	[Print End] Custom command (128 bytes)	Command data
END_DOC_BACK_FEED	46*5	[Print End] Paper feed (backfeed) (dot)	0 to -68
END_DOC_FEED	47*5	[Print End] Paper feed (feed) (dot)	0 to 255
END_DOC_EXT_BUZZER	140	[Print End] External buzzer	0: Pattern 1 1: Pattern 2 2: Pattern 3 3: Pattern 4 255: OFF
END_DOC_EXT_BUZZER_RPT	141	[Print End] External buzzer Frequency*6	1 to 255
PAPER_SIZE	50	Paper size	0: Letter 1: A4 3: 58 mm (54 × 3276 mm) 7: 58 mm (52 × 3276 mm) 8: 40 mm (36 × 3276 mm) 9 to 255: Other paper than above*7
ORIENTATION	51	Orientation	0: Portrait 1: Landscape
COLOR_MODE	52	Color printing mode	0: System*8 1: Driver*8

Property ID		Description	
.NET API	.NET API Win32 API		
PRESET	60[*9]	Preset	1 (116) : 58 mm Receipt Setting[*10] 118 : 58 mm Mark Form Feed Setting (Label) 5 (119) : A4 -> 58 mm Reduction Setting[*10] 80 : 40 mm Receipt Setting 81 : 40 mm Marked Paper Setting 83 : A4 -> 40 mm Reduction Setting 6 (255) : User setting

*1: For details of the print speed, see "User's Guide".
*2: Stored in the order of the higher digit to the lower digit.
*3: Driving the drawer both at Print Start and at Print End is not possible.
*4: As for the drawer control time, follow the specifications of your drawer.
*5: For the same timing feed, the last setting is valid.
*6: The external buzzer sounds will stop under one of the following conditions:
・Sounding for the set number of buzzers
・Opening the cover
・Executing the printer command "Stop External Buzzer"
*7: It cannot be set. You can only get the value.
*8: For details of color printing mode, see "3.2 Paper/Quality" in "SII Printer Driver for Windows User's Guide" for SLP720RT/SLP721RT series.
*9: When this property ID is specified, some contents of other property IDs are ignored. For details of preset, see "3.3.4 Use of Preset" in "SII Printer Driver for Windows User's Guide" for SLP720RT/SLP721RT series.
*10: Both 2 types of parameters can available in the settings. The value without the parentheses is obtained.

Chapter 7 Command Definition File

7.1 Overview

A command definition file is a text file (*.txt) that describes transmission data in a predefined format and is saved in ANSI format and UNICODE format.

7.2 Format

The command definition file format is described.

Command definition

[Command name 1] = [Transmission data 1]#[Comment 1]
[Command name 2] = [Transmission data 2]#[Comment 2]
[Command name 3] = [Transmission data 3]#[Comment 3]
:
:

Name		Description
Command definition		Data in command name unit described according to the format. Be sure to describe it on 1 line for 1 command definition.
	Command name	Describe an arbitrary command name to be specified by **DirectSendRead**. ・ Use ASCII characters except for "=" and "#". When any character other than ASCII characters is written, only the character is ignored. ・ Command names are case-sensitive. ・ Registerable command name is up to 33 bytes. When 34 bytes or more are written, 33 bytes are registered, and characters after 34 bytes are ignored. ・ When describing the same command names in 1 command definition file, the first command name is registered. When the command name already registered is specified, it is ignored. ・ Write "=" between the command name and the transmission data.

Name		Description
	Transmission data	Describe printer commands and data to send to the printer. ・ In the transmission data, do not include the ASB setting command that disables the response. The API that gets the printer status will not work properly. ・ When specifying binary data for transmission data, describe the data in two-digit hexadecimal numbers. Separate each data with a 1-byte space. Do not include a printer command that initializes the printer other than the printer command "Initialize Printer" in the binary data. For printer initialization, see "Technical Reference". Execute **Reset** or **ResetPrinter** to perform a hardware reset. ・ When specifying a string for transmission data, describe the data in ASCII characters. To specify the string, enclose it with " ". ・ The maximum transmission data size is 10240 bytes. However, the size when transmission data is specified as a string is the size after converting the specified string to binary data. Transmission data sizes of the command definition examples shown below are as follows: ・ CmdName_1: 3 bytes ・ CmdName_2: 4 bytes ・ CmdName_3: 3 bytes ・ CmdName_4: 6 bytes
	Comment	Describe the command definition etc. ・ Add "#" at the beginning of the comment. ・ There is no limitation of characters to be written. ・ Comments are optional.

Examples of command definition

```
CmdName_1="SII"#Comments1
CmdName_2=53 49 49 0A
CmdName_3=1D 56 00#Feed the paper to cut position.
CmdName_4=1B 40 "SII" 0A#Initialize the printer, the strings "SII" and line feed.
```

Reference

- The maximum size of a command definition file is 4 GB.
- The number of registerable command definitions depends on the available memory of the system.

7.3 How to Use

How to use the command definition file is described.

1. Create a command definition file.
2. Register the command definition file in the SDK internal memory by **SendDataFile**.
3. Specifying the command name by **DirectSendRead** executes the contents of the transmission data.

Caution

- The registered command definition is discarded by **CloseMonPrinter**.
- When a new command definition file is registered after registering the command definition file, the already registered command definition is overwritten.

Chapter 8 Printing Label Function

The label files (*.sl) created using Smart Label Creator or SII Layout Editor can be printed using the SDK. It also provides the function to replace text data, image data, or barcode data using the label file and print.

Smart Label Creator is software that can create labels.
SII Layout Editor is software that can create labels with iOS or Android.

In this manual, when describing both Smart Label Creator and SII Layout Editor, they are referred to as the "app".
If an individual description is required, the name of the product will be provided.

The app can be downloaded from the following web page.

- SLP720RT/SLP721RT series download page
  https://www.sii-ps.com/slp720rt/

For SII Layout Editor, by scanning a QR code below with the smartphone, redirected to the store and the app can be installed.

- iOS

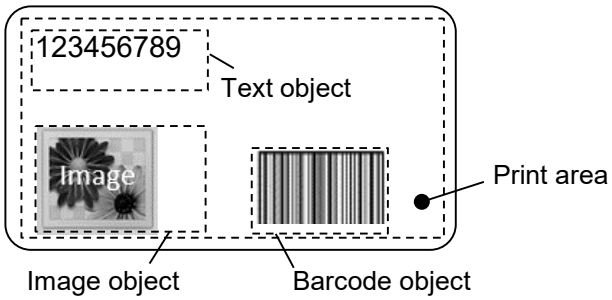
- Android

Reference

- It is possible to search "SII Layout Editor" and install in App Store or Google Play.

8.1 Structure of Label File

The label file is the file where objects are mapped within the print area for the label.



Example of label file (*.sl)

8.1.1 Types of Objects and Support in SDK

Supported objects in the SDK are shown in the following table.

Object	Description	Label File (*.sl) Source			Supported in Library
		SII Layout Editor	Smart Label Creator		
		iOS/Android	iOS/Android	Windows	
Text object	Handle text data	✓	✓	✓	✓
Image object	Handle image data	✓	✓	✓	✓
Barcode object	Handle barcode data	✓	✓	✓	✓
Drawing (rectangle) object	Handle the drawn data of a figures (rectangle)	✓	–	✓	✓
Drawing (circle/oval) object	Handle the drawn data of a figures (circle/oval)	✓	–	✓	✓
Drawing (line) object	Handle the drawn data of a figures (line)	✓	–	✓	✓
Frame object	Handle the drawn data of a decorative frame	–	–	✓	✓
Contact object	Retrieve data from the device contact book	✓	✓	–	–
DateTime object	Handle the data of the date and time	✓	✓	✓	✓
Return Address object	Handle the data of the sender	–	–	✓	✓
Group object	Grouping multiple objects	–	–	✓	✓

8.1.2 Precautions for Printing Label File Using SDK

Printing the label file using the app may differ from printing the label file using the SDK. Verify the performance with your actual device in advance.
Note the following when printing label files using the SDK.

① All object

- The drawing object mapped outside the print area is not supported.

② Text object

App

- The "Serialization" is not supported.
- If the font set using the app is not in the system, the text data is printed in the system standard font.

Windows Smart Label Creator

- The "Zip code" is not supported.
- The "Field Link" is not supported.
- It is not supported that the function to clear decorations or add different decorations or font sizes to the portions of text. The decorations and font size given to the 1st character are the decoration of the entire text.

③ Image object

Windows Smart Label Creator

- When the image source is "Link to File", the image object cannot be printed using the SDK.
- When the image source is "ClipArt", the image object cannot be printed using the SDK.

④ Barcode object

App

- Among the barcodes supported by the app, the following barcodes are supported by the SDK.
  - CODE39
  - ITF
  - CODE128
  - UPC-A
  - EAN13
  - CODABAR
  - UPC-E
  - EAN8
  - EAN128
  - GS1 Databar Omni-directional
  - GS1 Databar Truncated
  - GS1 Databar Stacked
  - GS1 Databar Stacked Omni-directional
  - GS1 Databar Limited
  - GS1 Databar Expanded
  - GS1 Databar Expanded Stacked
  - Maxicode
  - PDF417
  - Data Matrix

    - QR Code
- The "Serialization" is not supported.
- The barcode setting shown in following is not reflected.
    - Ratio of bar width
- The barcode image created using the app and the barcode image created by the SDK may not become the same barcode image.
- If the height of the barcode object is specified to be lower than the bar height using the app, the barcode will be reduced to fit within the object in the SDK and printed.
- When the security of the PDF417 is set to "-1" using the app, it is fixed to "0" in the SDK and the object is drawn.
- If the font set using the app is not in the system, the text data is printed in the system standard font.

Windows Smart Label Creator

- The "Field Link" is not supported.
- The barcode settings shown in following are not reflected.
    - Mode of PDF417
    - Security of Data Matrix
    - Mode of QR Code
- When the mode of the Data Matrix is set to "1" (rectangle), the barcode is drawn in the middle of the object in the SDK.

⑤ Drawing object

SII Layout Editor and Windows Smart Label Creator

- When "Line Width" is too thin, dashed, long dashed, or double lines may be squished.
- The drawn position of the drawing object may differ between the app and the SDK.

⑥ DateTime object

- If the font set using the app is not in the system, the text data is printed in the system standard font.

⑦ Return Address object

App

- If the font set using the app is not in the system, the text data is printed in the system standard font.

⑧ Group object

Windows Smart Label Creator

- The drawing position of objects may differ between Smart Label Creator and the SDK.

8.2 Method for Using Label File

The printing method using the label file is described below.

8.2.1 Print Label File as It Is from Library

Print command example (Win32)

1. Specify label file

```
SelectSmartLabelFileW( i_hdl, filepath) ;
```

2. Print label

```
PrintSmartLabelImageData( i_hdl, timeout, 0 );
```

Print command example (.NET)

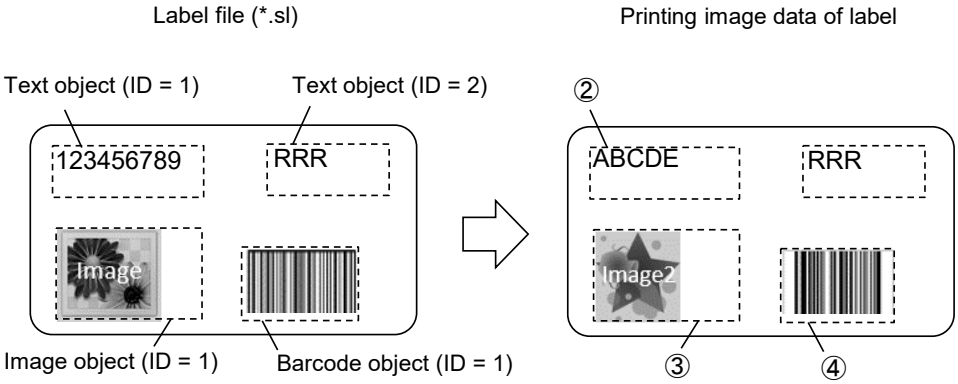
1. Specify label file

```
StatusAPI.SelectSmartLabelFile( filepath );
```

2. Print label

```
StatusAPI.PrintSmartLabelImageData( timeout, 0 );
```

8.2.2 Replace Object Data in Label File and Print



Print command example (Win32)

1. Specify label file

```
SelectSmartLabelFileW( i_hdl, filepath );
```

2. Replace text data of label (text object ID = 1)

```
ReplaceSmartLabelTextData( i_hdl, 1, textData );
```

3. Replace image data of label (image object ID = 1)

```
ReplaceSmartLabelImageData( i_hdl, 1, filepath );
```

4. Replace barcode data of label (barcode object ID = 1)

```
ReplaceSmartLabelBarcodeData( i_hdl, 1, barcodeData );
```

5. Print label

```
PrintSmartLabelImageData( i_hdl, timeout, 0 );
```

Print command example (.NET)

1. Specify label file
```
StatusAPI.SelectSmartLabelFile( filepath );
```

2. Replace text data of label (text object ID = 1)
```
StatusAPI.ReplaceSmartLabelTextData( 1, textData );
```

3. Replace image data of label (image object ID = 1)
```
StatusAPI.ReplaceSmartLabelImageData( 1, filepath );
```

4. Replace barcode data of label (barcode object ID = 1)
```
StatusAPI.ReplaceSmartLabelBarcodeData( 1, barcodeData );
```

5. Print label
```
StatusAPI.PrintSmartLabelImageData( timeout, 0 );
```

Seiko Instruments Inc.
1-8, Nakase, Mihama-ku, Chiba-shi,
Chiba 261-8507, Japan
Print System Division
Telephone:+81-43-211-1106
Facsimile:+81-43-211-8037

Seiko Instruments USA Inc.
Thermal Printer Div.
21221 S. Western Avenue, Suite 250, Torrance, CA 90501, USA
Telephone:+1-310-517-7778 Facsimile:+1-310-517-7779

Seiko Instruments GmbH
Siemensstrasse 9, D-63263 Neu-Isenburg, Germany
Telephone:+49-6102-297-0 Facsimile:+49-6102-297-222
info@seiko-instruments.de

Seiko Instruments (H.K.) Ltd.
4-5/F, Wyler Center 2,200 Tai Lin Pai Road, Kwai Chung, N.T., Kowloon, Hong Kong
Telephone:+852-2494-5160 Facsimile:+852-2424-0901

(Specifications are subject to change without notice.)