MP-A40 Series
SDK for Windows
Application Programmer's Guide

U00136022505

Seiko Instruments Inc.

U00136022500 August 2016
U00136022501 January 2018
U00136022502 February 2019
U00136022503 September 2021
U00136022504 October 2022
U00136022505 November 2024

Copyright© 2016-2024 by Seiko Instruments Inc.
All rights reserved.

Microsoft® and Windows® are registered trademarks of Microsoft Corporation in the U.S., Japan, and other countries.
Bluetooth® is a registered trademark of Bluetooth SIG, Inc.

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

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

SII is a trademark of Seiko Instruments Inc.

Introduction

This document describes the SDK (hereinafter referred to as "SDK") running on SII Printer Driver for Windows provided by Seiko Instruments Inc.

Target Printer Drivers

The following printer driver is supported by the SDK.

- SII Printer Driver for Windows (hereinafter referred to as "printer driver")

Terms

This section describes terms used in this document.

Definition	Description
ASB Setting command (ASB: Automatic Status Back)	Printer command "Automatic Status Back Enable / Disable". For details, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".
Printer status	Printer's status information retrievable by the SDK. This information includes the status to respond for the printer command "Automatic Status Back Enable / Disable". For details, see "6.1 Printer Status List".

Disclaimer

Seiko Instruments Inc. shall not be liable for any damages that may occur either directly or indirectly from use of this product.

Seiko Instruments Inc. shall not be liable for any damages or losses caused by or related to improper use of this product, improper handling without adherence to this document, repair or modification from third-party other than our personnel, and so forth.

Chapter 1 Overview 1-1

1.1 Introduction 1-1
1.2 Operating Conditions 1-1

Chapter 2 Installation 2-1

Chapter 3 Win32 API 3-1

3.1 Overview 3-1
3.2 Development Language 3-1
3.3 Library File 3-1
3.4 API List 3-2
3.5 API Details 3-4
MpOpen 3-4
MpClose 3-5
MpSetWriteTimeout 3-5
MpGetWriteTimeout 3-6
MpSetResponseTimeout 3-7
MpGetResponseTimeout 3-8
MpWrite 3-9
MpRead 3-10
MpWriteAndWaitResponse 3-11
MpReset 3-13
MpGetStatus 3-14
MpSetCallbackFunction 3-14
MpRegisterStyleSheet 3-15
MpDeleteStyleSheet 3-16
MpRegisterLogo 3-17
MpRegisterLogoEx 3-18
MpDeleteLogo 3-19
MpGetPrinterInformation 3-20
MpControlTransaction 3-21
MpSelectStandardMode 3-23
MpSelectPageMode 3-23
MpPrintPageModeData 3-25
MpSetPageModeVerticalPosition 3-25
MpSetStandardModeArea 3-26
MpSetStandardModeAlignment 3-27

MpSetHorizontalPosition 3-28
MpSetStandardModeBarcodeDirection 3-29
MpSetLineSpacing 3-30
MpSetCharacterRightSpace 3-30
MpSelectCharacterSet 3-31
MpSelectInternationalCharacterSet 3-32
MpSetCharacterFormatting 3-33
MpPrintText 3-35
MpPrintLogo 3-36
MpSendDataFile 3-36
MpSendDataFileEx 3-38
MpPrintBarcode 3-40
MpPrint2Dcode 3-43
MpPrintPageModeRectangle 3-45
MpFeedLine 3-47
MpFeedDotLine 3-47
MpFeedCutPosition 3-48
MpFeedMarkPosition 3-49

Chapter 4 .NET API 4-1

4.1 Overview 4-1
4.2 Development Language 4-1
4.3 Library File 4-1
4.4 API List 4-2
4.4.1 Printer Manager Class 4-2
4.4.2 Printer Exception Class 4-3
4.5 API Details (Printer Manager Class) 4-4
4.5.1 Property 4-4
IsOpened 4-4
4.5.2 Method 4-5
Open 4-5
Close 4-5
SetWriteTimeout 4-6
GetWriteTimeout 4-6
SetResponseTimeout 4-7
GetResponseTimeout 4-7
Write 4-8
Read 4-9
GetReadSize 4-9
WriteAndWaitResponse 4-10

Reset 4-11
GetStatus 4-11
StartCallbackFunction 4-12
StopCallbackFunction 4-12
RegisterStyleSheet 4-13
DeleteStyleSheet 4-13
RegisterLogo 4-14
RegisterLogoEx 4-15
DeleteLogo 4-16
GetPrinterInformation 4-16
ControlTransaction 4-17
SelectStandardMode 4-18
SelectPageMode 4-18
PrintPageModeData 4-19
SetPageModeVerticalPosition 4-20
SetStandardModeArea 4-20
SetStandardModeAlignment 4-21
SetHorizontalPosition 4-21
SetStandardModeBarcodeDirection 4-22
SetLineSpacing 4-23
SetCharacterRightSpace 4-23
SelectCharacterSet 4-24
SelectInternationalCharacterSet 4-24
SetCharacterFormatting 4-25
PrintText 4-26
PrintLogo 4-27
SendDataFile 4-27
SendDataFileEx 4-28
PrintBarcode 4-29
Print2Dcode 4-30
PrintPageModeRectangle 4-32
FeedLine 4-33
FeedDotLine 4-33
FeedCutPosition 4-34
FeedMarkPosition 4-34
4.5.3 Event 4-35
statusChanged 4-35
4.6 API Details (Printer Exception Class) 4-36
4.6.1 Method 4-36
GetErrorCode 4-36

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 Printer Information 6-3
6.3 Batch Process Control Method 6-5
6.4 Print Direction 6-5
6.5 Print Position 6-5
6.6 Character Set 6-6
6.7 International Character Set 6-6
6.8 Character Font 6-7
6.9 Character Size 6-7
6.10 Underline 6-7
6.11 Bold Print 6-8
6.12 Reverse Print 6-8
6.13 Character Rotation Print 6-9
6.14 Barcode Type 6-10
6.15 Barcode Module Width 6-10
6.16 Barcode Height 6-11
6.17 Barcode HRI Character 6-12
6.18 Barcode N:W Ratio 6-12
6.19 2-Dimensional Barcode Type 6-13
6.20 2-Dimensional Barcode Mode 6-13
6.21 2-Dimensional Barcode Module Size 6-15
6.22 2-Dimensional Barcode Module Height 6-15
6.23 Number of 2-Dimensional Barcode Columns 6-16
6.24 Number of 2-Dimensional Barcode Rows 6-16
6.25 2-Dimensional Barcode Error Correction Level 6-17

Chapter 1 Overview

1.1 Introduction

This chapter describes the overview of the SDK.

The SDK is a dynamic link library to directly control printers, provided with the printer driver for developers.

The SDK is provided with the printer driver and uses the printer driver to work.

You can use the SDK to directly control printers in an application development and design the application independent of the port type.

For details of specific usages of the SDK, see sample programs provided for each language.

1.2 Operating Conditions

It basically follows the operating environment of the printer driver, and the use conditions and limitations of the memory switch.

See "MP-A40 Series Printer Driver User's Guide" for details.

In addition, the following operating conditions must be met.

- .NET Framework Version 2.0 or later must be installed.
- The bidirectional support function must be enabled.
  (See "MP-A40 Series Printer Driver User's Guide " for details about how to set it.)
- The printer pool function must be disabled.
  (See "MP-A40 Series Printer Driver User's Guide " for details about how to set it.)
- When all of the following conditions apply, use a shared printer. (If a shared printer is not used, there is a possibility that transmitted data will be interrupted.)
  1) When using 1 printer from multiple hosts via TCP/IP connection
  2) When sending data in multiple parts

  For information on how to send data in a batch, see (Mp)ControlTransaction.

Chapter 2 Installation

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

Chapter 3 Win32 API

3.1 Overview

This chapter describes the SDK for Win32 development environment (Win32 API).

3.2 Development Language

The following development language is covered.

- Visual C++ *1

*1: Visual C++ 2008 or later is recommended

3.3 Library File

The library file has the following file name.

- SiiMpa4Api.dll

The library file is stored in the Windows system folder.

Use the library file without moving it from the folder. In this case, you do not have to set a path to the folder containing the library file.

When the library file is moved to another location, the library file could not be updated properly during version up of the printer driver.

3.4 API List

The following APIs are implemented in the SDK.

✓: Available
- : Not available

API	Function Summary	Standard Mode	Page Mode
MpOpenA[*1 *2] **MpOpenW**[*1 *2]	Begins using the SDK in the specified printer and returns the session ID.	✓	✓
MpClose	Ends using the SDK with the specified session ID.	✓	✓
MpSetWriteTimeout	Sets the timeout value of the API to transmit data.	✓	✓
MpGetWriteTimeout	Retrieves the timeout value of the API to transmit data.	✓	✓
MpSetResponseTimeout	Sets the timeout value of the API to receive data.	✓	✓
MpGetResponseTimeout	Retrieves the timeout value of the API to receive data.	✓	✓
MpWrite[*3]	Sends binary data to the printer.	✓	✓
MpRead	Retrieves the data received from the printer.	✓	✓
MpWriteAndWaitResponse[*3]	Receives response data after sending binary data.	✓	✓
MpReset[*4]	Resets the printer.	✓	✓
MpGetStatus	Retrieves the latest printer status.	✓	✓
MpSetCallbackFunction	Registers the callback function called when detecting the printer status change.	✓	✓
MpRegisterStyleSheetA[*1 *3] **MpRegisterStyleSheetW**[*1 *3]	Registers the style sheet in the printer.	✓	✓
MpDeleteStyleSheet[*3]	Deletes the style sheet from the printer.	✓	✓
MpRegisterLogoA[*1 *3] **MpRegisterLogoW**[*1 *3]	Registers the logo in the printer.	✓	✓
MpDeleteLogo[*3]	Deletes the registered logo from the printer.	✓	✓
MpGetPrinterInformation[*3]	Retrieves the specified printer information.	✓	✓
MpControlTransaction[*5]	Processes the target APIs in a batch.	✓	✓
MpSelectStandardMode[*3]	Begins the standard mode.	-	✓
MpSelectPageMode[*3]	Specifies the print area and print direction in the page mode and begins the page mode.	✓	-
MpPrintPageModeData[*3]	Prints the page mode data.	-	✓[*6]
MpSetPageModeVerticalPosition[*3]	Sets the vertical absolute position in page mode.	-	✓[*6]
MpSetStandardModeArea[*3]	Sets the print area in the standard mode.	✓[*7]	-
MpSetStandardModeAlignment[*3]	Sets the print position the in standard mode.	✓[*7]	-
MpSetHorizontalPosition[*3]	Sets the horizontal absolute position.	✓	✓

API	Function Summary	Standard Mode	Page Mode
MpSetStandardModeBarcode Direction[*3]	Sets the print direction of a barcode.	✓[*7]	-
MpSetLineSpacing[*3]	Sets the line spacing.	✓[*8]	✓[*8]
MpSetCharacterRightSpace[*3]	Sets the amount of character right space.	✓[*8]	✓[*8]
MpSelectCharacterSet	Sets the character set.	✓	✓
MpSelectInternationalCharacter Set	Sets the international character set.	✓	✓
MpSetCharacterFormatting[*3]	Sets the character formatting.	✓	✓
MpPrintTextA[*1 *3] **MpPrintTextW**[*1 *3]	Prints the character string.	✓	✓
MpPrintLogo[*3]	Prints the image registered in the printer.	✓	✓
MpSendDataFileA[*1 *3] **MpSendDataFileW**[*1 *3]	Sends the data of the specified file.	✓	✓
MpPrintBarcode[*3]	Prints the barcode.	✓	✓
MpPrint2Dcode[*3]	Prints the 2-dimensional barcode.	✓	✓
MpPrintPageModeRectangle[*3]	Prints the rectangle when the page mode is selected.	-	✓[*6]
MpFeedLine[*3]	Feeds the paper by line.	✓	✓
MpFeedDotLine[*3]	Feeds the paper by dot.	✓	✓
MpFeedCutPosition[*3]	Feeds the paper to the paper cut position.	✓	✓
MpFeedMarkPosition[*3]	Performs marked paper form feed.	✓	✓

*1: Specify arguments of strings by MBCS (MultiByte Character Set) or Unicode. Use API added the suffix 'A' for MBCS or 'W' for Unicode. Note that a suffix of 'A' or 'W' is omitted hereafter in the descriptions.
*2: This API succeeds even when the printer is not connected to the system or the printer power is turned off.
*3: When any print job exists in the spooler, or any disconnection or communication failure with the printer occurs, this API fails.
*4: When any print job exists in the spooler, any disconnection with the printer occurs, or the printer power is turned off, this API fails.
*5: When any print job exists in the spooler, or any disconnection or communication failure with the printer occurs, this API fails by executing it with "End batch print and batch process".
*6: When use this API, execute **MpSelectPageMode** beforehand to start the page mode.
*7: When use this API, execute **MpSelectStandardMode** beforehand to change to the standard mode.
*8: Independent settings are available for the standard mode and page mode respectively.

3.5 API Details

Caution

- For Bluetooth connection, when the connection is once disconnected, part of response data may not be retrieved.
- For Bluetooth connection, response data of the disconnected printer cannot be retrieved.

MpOpen

Begins using the SDK in the specified printer and returns the session ID.

INT **MpOpen**(

LPCTSTR *pszPrnName*,

LPDWORD *pdwSessionId*)

Parameters

pszPrnName

Name of the printer that uses the SDK

Specifies the printer name (friendly name).

pdwSessionId

Session ID

Specifies the variable of the session ID that identifies the printer.

Return value

Returns 0 for success.

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

Remarks

- The number that printers open per 1 process is up to 8.
- When the session ID retrieved in this API is not used, be sure to close it by **MpClose**.
- When the printer driver connects other than USB, Bluetooth or TCP/IP, this API fails.

MpClose

Ends using the SDK with the specified session ID.

INT **MpClose**(

DWORD *dwSessionId*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

Return value

Returns 0 for success.

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

Remarks

- When the session ID called by this API is used in another API, this API is not executed until the process is completed.
- The command data retained by **MpControlTransaction** is discarded.
- All settings that have been associated with the session ID specified by this API are discarded.

MpSetWriteTimeout

Sets the timeout value of the API to transmit the data.

INT **MpSetWriteTimeout**(

DWORD *dwSessionId*,

DWORD *dwTimeout*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

dwTimeout

Data transmission timeout value

Specifies the transmission wait time (in milliseconds) of the API to transmit the data. The range of timeout is from 3000 ms to 90000 ms. When the value is less than 3000 ms, it is corrected to 3000 ms and when the value exceeds 90000 ms, it is corrected to 90000 ms.

The setting is available for the following APIs.

- **MpWrite**
- **MpWriteAndWaitResponse**(Transmission process part)
- **MpRegisterStyleSheet**
- **MpRegisterLogo**
- **MpControlTransaction**("End batch print and batch process" is selected)
- **MpPrintText**
- **MpSendDataFile**
- **MpPrintBarcode**
- **MpPrint2Dcode**

Initial value

15000 ms

Return value

Returns 0 for success.

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

Remarks

- The value set by this API can be confirmed by **MpGetWriteTimeout**.

MpGetWriteTimeout

Retrieves the timeout value of the API to transmit the data.

INT **MpGetWriteTimeout**(

DWORD *dwSessionId*,

LPDWORD *pdwTimeout*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

pdwTimeout

Data transmission timeout value

Specifies the variable that retrieves the transmission wait time (in milliseconds) of the API to transmit data.

Return value

Returns 0 for success.

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

Remark

- The value retrieved by this API can be set by **MpSetWriteTimeout**.

MpSetResponseTimeout

Sets the timeout value of the API to receive the data.

INT **MpSetResponseTimeout**(
DWORD *dwSessionId*,
DWORD *dwTimeout*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

dwTimeout

Data reception timeout value

Specifies the reception wait time (in milliseconds) of the API to receive the data.
The range of timeout is from 3000 ms to 90000 ms. When the value is less than 3000 ms, it is corrected to 3000 ms and when the value is more than 90000 ms, it is corrected to 90000 ms.
The setting is available for the following APIs.

- **MpWriteAndWaitResponse**(Reception process part)
- **MpGetPrinterInformation**

Initial value

3000 ms

Return value

Returns 0 for success.

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

Remarks

- The value set by this API can be confirmed by **MpGetResponseTimeout**.

MpGetResponseTimeout

Retrieves the timeout value of the API to receive the data.

INT **MpGetResponseTimeout**(
DWORD *dwSessionId*,
LPDWORD *pdwTimeout*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

pdwTimeout

Data reception timeout value

Specifies the variable that retrieves the reception wait time (in milliseconds) of the API to receive the data.

Return value

Returns 0 for success.

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

Remarks

- The value retrieved by this API can be set by **MpSetResponseTimeout**.

MpWrite

Sends binary data to the printer.

INT **MpWrite**(

DWORD *dwSessionId*,

LPBYTE *pCmd*,

DWORD *cbCmd*,

DWORD *dwOffset*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

pCmd

Transmit data buffer

Specifies the buffer that stored the data to send to the printer.

cbCmd

Transmit data size

Specifies the size of the buffer indicated by *pCmd* in bytes.

dwOffset

Transmit data offset

Specifies the starting position of the data to send.

Return value

Returns 0 for success.

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

Remarks

- When the offset of transmission data (*dwOffset*) is larger than the size of transmission data (*cbCmd*), this API fails.
- When **MpControlTransaction** is not used, the transmission timeout period of this API is the value set by **MpGetWriteTimeout**.
- Data transmission processed by this API is not included in the jobs of the printer driver.
- This API is aborted by **MpReset**.
- For the commands or data such as image data that do not allow interrupting of other data before completion of transmission, output it in a batch. When the data is divided to output, the data from other process may interrupt. Make sure to output all data by calling this API at a time.
- Do not use the data that disables the ASB setting command in this API. Otherwise, the API that retrieves printer status does not work properly.

- When the number of data bytes written in the printer is less than the size of *cbCmd*, this API fails.
- Do not include the printer command "Hardware Reset" or "Printer Reset" in the data to send.
  Use **MpReset** when resetting the printer.

MpRead

Retrieves the data received from the printer.

INT **MpRead**(

DWORD *dwSessionId*,

LPBYTE *pData*,

DWORD *cbData*,

LPDWORD *pcbNeeded*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

pData

Receive data buffer

Specifies the buffer that stored the data to receive.

cbData

Maximum receive data size

Specifies the size of the buffer indicated by *pData* in bytes.
The maximum receive data size is 4096 bytes. The data in excess of the maximum size will be discarded in the order from the oldest.

pcbNeeded

Data size variable

Specifies the variable to store the received data size or necessary buffer size to receive the data.

Return value

Returns 0 for success.

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

Remarks

- This API retrieves the (unconverted) data received after calling of **MpOpen**.
- The received data of this API may include the response data replied by the operation of other process.

- When the printer is reconnected, the accumulated data in the printer may be received in a lump.
- To retrieve the response data size, specify 0 for *cbData*. This API fails (ERR_WORKAREA_NO_MEMORY(-260)), and the response data size is stored in the variable specified for *pcbNeeded*.

MpWriteAndWaitResponse

Receives the response data after sending binary data.

INT **MpWriteAndWaitResponse**(

DWORD *dwSessionId*,

LPBYTE *pCmd*,

DWORD *cbCmd*,

LPBYTE *pResp*,

DWORD *cbResp*,

LPDWORD *pcbNeeded*,

BOOL *bRespFlag*,

BOOL *bIncASBData*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

pCmd

Transmit data buffer

Specifies the buffer that stored the data to send.

cbCmd

Transmit data size

Specifies the size of the data to send.

pResp

Receive data buffer

Specifies the buffer that stores the data to retrieve.

cbResp

Maximum receive data size

Specifies the maximum size of the data to receive from the printer.

Specifies 0 when there is no need to retrieve the data.

The maximum receive data size is 4096 bytes. The data in excess of the maximum size will be set at 4096 bytes.

pcbNeeded

Data size variable

Specifies the variable to store the received data size or necessary buffer size to receive the data.

bRespFlag

Receive operation flag

Specifies the receive operation.

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

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

bIncASBData

Specifies whether the response for ASB setting command is included in the receive data.

TRUE: Retrieves the data including responses for the ASB setting command.

FALSE: Retrieves the data excluding responses for the ASB setting command.

Return value

Returns 0 for success.

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

Remarks

- This API retrieves the (unconverted) data received during the execution.
- The retrieved data of this API may include the response data replied by the operation of other process.
- This API is aborted by **MpReset**.
- When the number of data bytes written in the printer is less than the size of *cbCmd*, this API fails.
- The transmission timeout period of this API is the value set by **MpGetWriteTimeout**.
- The reception timeout period of this API is the value set by **MpGetResponseTimeout**.
- For the commands and data that do not allow interrupting of other data before completion of transmission, such as image data, output it in a batch. When the data is divided to output, the data from other process may interrupt. Make sure to output all data by calling this API at a time.
- Do not use the data that disables the ASB setting command in this API. Otherwise, the API that retrieves printer status does not work properly.
- Do not include the printer command "Hardware Reset" or "Printer Reset" in the binary data to send.
  Use **MpReset** when resetting the printer.
- To retrieve the response data size, specify 0 for *cbResp*. This API fails (ERR_WORKAREA_NO_MEMORY(-260)), and the response data size is stored in the variable specified for *pcbNeeded*.

MpReset

Resets the printer.

INT **MpReset**(

DWORD *dwSessionId*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

Return value

Returns 0 for success.

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

Remarks

- When this API is called, the following APIs are aborted.
  - **MpWrite**
  - **MpWriteAndWaitResponse**
  - **MpControlTransaction** ("End batch print and batch process" is selected)
  - **MpPrintText**
  - **MpSendDataFile**
- After executing this API, wait for a few seconds to send the data. If data transmission is performed right after executing this API, it may cause data skipping.
- During execution of this API, the printer status responds disconnected status.
- For Bluetooth connection, when executing this API in the state that the printer is unaccepting data, this API succeeds, but the reset is not executed until the printer is ready to print.
  And in the meantime, data transmission cannot be performed.

MpGetStatus

Retrieves the latest printer status.

INT **MpGetStatus**(

DWORD *dwSessionId*,

LPDWORD *pdwStatus*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

pdwStatus

Printer status

Specifies the variable that stores the printer status.

Return value

Returns 0 for success.

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

Remarks

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

MpSetCallbackFunction

Registers the callback function that is called when detecting the printer status change.

INT **MpSetCallbackFunction**(

DWORD *dwSessionId*,

INT (CALLBACK EXPORT **lpfnCallBackStatus*) (DWORD *dwStatus*))

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

lpfnCallBackStatus

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 cancelled.

dwStatus

Printer status

Specifies the variable in which the printer status is stored.

Return value

Returns 0 for success.

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

Remarks

- APIs in the SDK cannot be called with the same session ID from the registered callback function.
- When reconnection to the printer is detected, the printer status is the value received last at the time.
- 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 the callback function is registered with this API, the callback function is called with the current printer status.
- When this API is called in that state that the callback function is already registered, the registered function becomes invalid and new callback function is registered.
- The return value of the callback function is ignored.
- The time between receiving the printer status and calling the callback function is not guaranteed.
- For details of the printer status, see "6.1 Printer Status List".

MpRegisterStyleSheet

Registers the style sheet in the printer.

INT **MpRegisterStyleSheet**(

DWORD *dwSessionId*,

LPCTSTR *pszFilePath*,

BYTE *byRegNum*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

pszFilePath

File path of the style sheet

Specifies the file path of the style sheet to be registered in the printer.

Specify CSS file(*.css) for the file path.

byRegNum

Style sheet number

Specifies the style sheet number to be registered in the printer.

Specify the value among the number 1 to 4.

When a number other than above is specified, this API fails (ERR_PARAM(-90)).

Return value

Returns 0 for success.

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

Remarks

- The number of styles that can be registered in 1 style sheet is up to 64.
- See "Style Sheet Registration" command in "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE" for the details of creating registrable style sheets.
- The readable file size is up to 4294967295 bytes.
  However, when the memory is not secured on the system, this API fails.
- The transmission timeout period of this API is the value set by **MpGetWriteTimeout**.

MpDeleteStyleSheet

Deletes the registered style sheet from the printer.

INT **MpDeleteStyleSheet**(

DWORD *dwSessionId*,

BYTE *byRegNum*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

byRegNum

Style sheet number

Specifies the style sheet number to be deleted from the printer.

Specify the value among the number 1 to 4.

When a number other than above is specified, this API fails (ERR_PARAM(-90)).

Return value

Returns 0 for success.

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

MpRegisterLogo

Registers the logo in the printer.

INT **MpRegisterLogo**(
DWORD *dwSessionId*,
LPCTSTR *pszFilePath*,
BYTE *byRegNum*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

pszFilePath

File path of the logo image data

Specifies the file path of the logo to be registered in the printer.
Specify Windows bitmap file(*.bmp) or JPEG file(*.jpg / *.jpeg) for the file path.
Dithering is fixed to be enabled when registering the logo image data with this API.

byRegNum

Logo number

Specifies the logo number to be registered in the printer.
Specify the value among the number 0 to 99.
When a number other than above is specified, this API fails (ERR_PARAM(-90)).

Return value

Returns 0 for success.

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

Remarks

- The readable file size is up to 4294967295 bytes.
  However, when the memory is not secured on the system, this API fails.
- The registrable size of the logo image is up to 8192 dots in width and up to 2304 dots in height.
- The registration status of the logo registered by this API can be confirmed by **MpGetPrinterInformation.** Specify GET_NV_MEM_KEYCODE_LIST for *dwPrnInfo*.
- When the logo image data is registered to the same logo number, newer data is valid.
- The transmission timeout period of this API is the value set by **MpGetWriteTimeout**.

MpRegisterLogoEx

Registers the logo in the printer.

INT **MpRegisterLogoEx**(
DWORD *dwSessionId*,
LPCTSTR *pszFilePath*,
BYTE *byRegNum,*
BYTE *byOption*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

pszFilePath

File path of the logo image data

Specifies the file path of the logo to be registered in the printer.
Specify Windows bitmap file(*.bmp) or JPEG file(*.jpg / *.jpeg) for the file path.

byRegNum

Logo number

Specifies the logo number to be registered in the printer.
Specify the value among the number 0 to 99.
When a number other than above is specified, this API fails (ERR_PARAM(-90)).

byOption

Option

Specifies the option when registering the logo in the printer.
Specify the following values.

byOption	Description
0	Dithering is enabled
1	Dithering is disabled

Return value

Returns 0 for success.

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

Remarks

- The readable file size is up to 4294967295 bytes.
  However, when the memory is not secured on the system, this API fails.
- The registrable size of the logo image is up to 8192 dots in width and up to 2304 dots in height.
- The registration status of the logo registered by this API can be confirmed by **MpGetPrinterInformation.** Specify GET_NV_MEM_KEYCODE_LIST for *dwPrnInfo*.
- When the logo image data is registered to the same logo number, newer data is valid.
- The transmission timeout period of this API is the value set by **MpGetWriteTimeout**.

MpDeleteLogo

Deletes the registered logo from the printer.

INT **MpDeleteLogo**(

DWORD *dwSessionId*,

BYTE *byRegNum*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

byRegNum

Logo number

Specifies the logo number to be deleted from the printer.

Specify the value among the number 0 to 99.

When a number other than above is specified, this API fails (ERR_PARAM(-90)).

Return value

Returns 0 for success.

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

Remarks

- The registration status of the logo after executing this API can be confirmed by **MpGetPrinterInformation**. Specify GET_NV_MEM_KEYCODE_LIST for *dwPrnInfo*.

MpGetPrinterInformation

Retrieves the specified printer information.

INT **MpGetPrinterInformation**(
DWORD *dwSessionId*,
DWORD *dwPrnInfo*,
LPBYTE *pData*,
DWORD *cbData*,
LPDWORD *pdwData*,
LPDWORD *pcbNeeded*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

dwPrnInfo

Printer information

Specifies the printer information to retrieve.
When the response format of the printer information to specify is numeric, the content of *pData* is converted into a numerical value and stored in the variable specified for *pdwData*. Other than that, *pdwData* is not defined.
When an invalid value is specified, this API fails (ERR_PARAM(-90)).
See "6.2 Printer Information" for the content and response format of the printer information.

pData

Receive data buffer

Specifies the buffer to store the data to retrieve.

cbData

Receive data size

Specifies the maximum size of the data to receive from the printer.

pdwData

Receive data (numerical value)

Specifies the variable that stores numerical *pData* when the response format of *dwPrnInfo* is numeric.
Specify NULL when the response format of *dwPrnInfo* is non-numeric.

pcbNeeded

Data size variable

Specifies the variable to store the received data size or necessary buffer size to receive the data.

Return value

Returns 0 for success.

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

Remarks

- The reception timeout period of this API is the value set by **MpGetResponseTimeout**.
- When this API fails, the values of *pData* and *pcbNeeded* are not defined.
- To retrieve the response data size, specify 0 for *cbResp*. This API fails (ERR_WORKAREA_NO_MEMORY(-260)), and the data size is stored in the variable specified for *pcbNeeded*.

MpControlTransaction

Processes the target APIs in a batch.

INT **MpControlTransaction**(

DWORD *dwSessionId*,

BYTE *byControl*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

byControl

Control method

Specifies the control method of batch process.

When an invalid value is specified, this API fails (ERR_PARAM(-90)).

See "6.3 Batch Process Control Method" for details.

Return value

Returns 0 for success.

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

Remarks

- The following APIs are applicable for batch process by this API.
  - **MpWrite**
  - **MpSelectStandardMode**
  - **MpSelectPageMode**
  - **MpPrintPageModeData**
  - **MpSetPageModeVerticalPosition**
  - **MpSetStandardModeArea**
  - **MpSetStandardModeAlignment**
  - **MpSetHorizontalPosition**
  - **MpSetStandardModeBarcodeDirection**
  - **MpSetLineSpacing**
  - **MpSetCharacterRightSpace**
  - **MpSetCharacterFormatting**
  - **MpPrintText**
  - **MpPrintLogo**
  - **MpSendDataFile**
  - **MpPrintBarcode**
  - **MpPrint2Dcode**
  - **MpPrintPageModeRectangle**
  - **MpFeedLine**
  - **MpFeedDotLine**
  - **MpFeedCutPosition**
  - **MpFeedMarkPosition**

- When this API is executed by "End batch print and batch process", the registered process is discarded after batch printing.
  When the registered process is not used and this API is executed by "End batch process", the registered process is discarded without printing.
- Even when "End batch process" is executed without executing "Start batch process", this API succeeds.
- When "End batch print and batch process" is executed without executing "Start batch process", this API fails.
- Even when "Start batch process" is executed and then "End batch process" or "End batch print and batch process" is executed in the state that no process registered, this API succeeds.
- When "Start batch process" is executed and it is executed again without executing "End batch process" or "End batch print and batch process", the registered process is discarded, and new batch process is started.
- When this API is being executed by "End batch print and batch process", this API is aborted by **MpReset**.
- When "Start batch process" is executed from another thread during executing "End batch print and batch process", the following operations are performed.
  - The execution of "End batch print and batch process" is maintained.
  - "Start batch process" from another thread begins new buffering as another "Start batch process".

- The transmission timeout period of this API during executing "End batch print and batch process" is the value set by **MpGetWriteTimeout**.

MpSelectStandardMode

Begins standard mode.

INT **MpSelectStandardMode**(
DWORD *dwSessionId*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

Return value

Returns 0 for success.

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

Remarks

- When this API is executed during page mode, the page mode data is discarded and the printer begins standard mode.

MpSelectPageMode

Specifies the print area and print direction in page mode and begins the page mode.

INT **MpSelectPageMode**(
DWORD *dwSessionId*,
WORD *wStartX*,
WORD *wStartY*,
WORD *wWidth*,
WORD *wHeight*,
BYTE *byDirection*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

wStartX

Horizontal start point

Specifies the horizontal start point in the print area of page mode in dots.

wStartY

Vertical start point

Specifies the vertical start point in the print area of page mode in dots.

wWidth

Horizontal length

Specifies the horizontal length in the print area of page mode in dots.

Specify 1 or larger number.

wHeight

Vertical length

Specifies the vertical length in the print area of page mode in dots.

Specify 1 or larger number.

byDirection

Print direction

Specifies the print direction of page mode.

When an invalid value is specified, this API fails (ERR_PARAM(-90)).

See "6.4 Print Direction" for details.

Return value

Returns 0 for success.

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

Remarks

- When the horizontal start point is over the print width, or the vertical start point is 2400 or more, this API succeeds, however the setting is dependent on the printer status.
- When the specified area exceeds the printable area, the exceeding area is specified for the printable area.

MpPrintPageModeData

Prints the page mode data.

INT **MpPrintPageModeData**(
DWORD *dwSessionId*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

Return value

Returns 0 for success.

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

MpSetPageModeVerticalPosition

Sets the vertical absolute position in page mode.

INT **MpSetPageModeVerticalPosition**(
DWORD *dwSessionId*,
WORD *wVerticalPosition*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

wVerticalPosition

Absolute mapping starting position in a vertical direction

Specifies the absolute mapping starting position in a vertical direction in page mode in dots.
When the value that exceeds the print area of page mode specified in **MpSelectPageMode** is specified, the API succeeds however the setting is ignored.

Return value

Returns 0 for success.

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

MpSetStandardModeArea

Sets the print area in standard mode.

INT **MpSetStandardModeArea**(
DWORD *dwSessionId*,
WORD *wLeftMargin*,
WORD *wPrintAreaWidth*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

wLeftMargin

Left margin

Specifies the position of the left margin in dots.
When the value that exceeds 1 line printable area is entered, the maximum value of the printable area is set as the left margin,

wPrintAreaWidth

Print area width

Specifies the width of the print area in dots.
When the value that exceeds 1 line printable area is entered, all area excluding the left margin is set as the print area width,

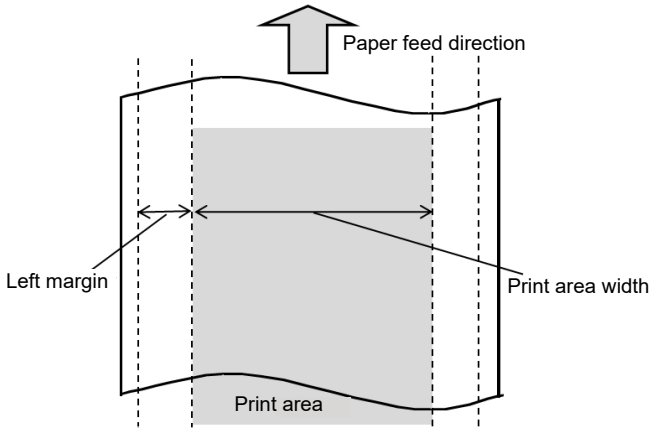
Return value

Returns 0 for success.

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

Remarks

- Execute this API before sending the print data.
- The value of the left margin is not included in the print area width.



MpSetStandardModeAlignment

Sets the print position in standard mode.

INT **MpSetStandardModeAlignment**(
DWORD *dwSessionId*,
BYTE *byAlign*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

byAlign

Print position

Specifies the print position for printing in standard mode.
When an invalid value is specified, this API fails (ERR_PARAM(-90)).
See "6.5 Print Position" for details.

Return value

Returns 0 for success.

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

Remarks

- Execute this API before sending the print data.
- The print position cannot be changed in the middle of a line.

MpSetHorizontalPosition

Sets the horizontal absolute position.

INT **MpSetHorizontalPosition**(

DWORD *dwSessionId*,

WORD *wHorizontalPosition*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

wHorizontalPosition

Horizontal print starting position

Specifies the horizontal print starting position in dots.

When the value that exceeds the print area of page mode specified in **MpSelectPageMode** is specified, the API succeeds however the setting is ignored.

Return value

Returns 0 for success.

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

Remarks

- In standard mode, this API is executed based on the left margin set in **MpSetStandardModeArea**.

MpSetStandardModeBarcodeDirection

Sets the print direction of a barcode in standard mode.

INT **MpSetStandardModeBarcodeDirection**(
DWORD *dwSessionId*,
BYTE *byRotate*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

byRotate

Print direction

Specifies the barcode print direction.
When 90 degrees right rotation or 90 degrees left rotation is specified, the printable width of a barcode is up to 300mm. When the width exceeds 300mm, the barcode printing is not performed.
When an invalid value is specified, this API fails (ERR_PARAM(-90)).
See "6.13 Character Rotation Print" for details.

Return value

Returns 0 for success.

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

Remarks

- Do not use this API for printing in the page mode. If this API is executed in the page mode, the setting is reflected to the print after starting the standard mode.

MpSetLineSpacing

Sets the line spacing.

INT **MpSetLineSpacing**(
DWORD *dwSessionId*,
BYTE *byLineSpacing*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

byLineSpacing

Line spacing

Specifies the line spacing.

Return value

Returns 0 for success.

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

Remarks

- Independent settings are available in standard mode and page mode respectively.

MpSetCharacterRightSpace

Sets the character right space amount.

INT **MpSetCharacterRightSpace**(
DWORD *dwSessionId*,
BYTE *bySpace*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

bySpace

Character right space amount

Specifies the character right space amount in dots.

Return value

Returns 0 for success.

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

Remarks

- Independent settings are available in standard mode and page mode respectively.
- This API is enabled for 1-byte characters and 2-byte characters. For 2-byte characters, the left space is set at 0.

MpSelectCharacterSet

Sets the character set.

INT **MpSelectCharacterSet**(

DWORD *dwSessionId*,

BYTE *byCharSet*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

byCharSet

Character set

Specifies the character set to use.

When an invalid value is specified, this API fails (ERR_PARAM(-90)).

See "6.6 Character Set" for details.

Initial value

It depends on the language setting of Windows OS.

For Japanese: Katakana (CODEPAGE_KATAKANA)

For other languages: Latin (CODEPAGE_1252)

Return value

Returns 0 for success.

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

Remarks

- Even when the printer is reset by **MpReset**, or the printer power is turned off, the setting of this API is retained.
- The setting of this API is used for the printer setting when calling the following APIs.
  - **MpPrintText**
  - **MpSendDataFile**(Text file specified)
- When Arabic (CODEPAGE_864) is specified for the character set by this API, the text is printed in Font A (24×12) regardless of specifying the character font in **MpSetCharacterFormatting**.

MpSelectInternationalCharacterSet

Sets the international character set.

INT **MpSelectInternationalCharacterSet**(

DWORD *dwSessionId*,

BYTE *byIntCharSet*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

byIntCharSet

International character set

Specifies the international character set to use.

When an invalid value is specified, this API fails (ERR_PARAM(-90)).

See "6.7 International Character Set" for details.

Initial value

It depends on the language setting of Windows OS.

For Japanese: Japan (INT_CHAR_SET_JAPAN)

For other languages: USA (INT_CHAR_SET_USA)

Return value

Returns 0 for success.

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

Remarks

- Even when the printer is reset by **MpReset**, or the printer power is turned off, the setting of this API is retained.
- The setting of this API is used for the printer setting when calling the following APIs.
  - **MpPrintText**
  - **MpSendDataFile**(Text file specified)

MpSetCharacterFormatting

Sets the character formatting.

INT **MpSetCharacterFormatting**(

DWORD *dwSessionId*,

BYTE *byType*,

BYTE *byVerticalScale*,

BYTE *byHorizontalScale*,

BYTE *byUnderline*,

BYTE *byBold*,

BYTE *byReverse*,

BYTE *byRotate*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

byType

Character font

Specifies the character font.

See "6.8 Character Font" for details.

byVerticalScale

Character size (in vertical direction)

Specifies the character size (in vertical direction).

See "6.9 Character Size" for details.

byHorizontalScale

Character size (in horizontal direction)

Specifies the character size (in horizontal direction).

See "6.9 Character Size" for details.

byUnderline

Underline

Specifies the underline.

See "6.10 Underline" for details.

byBold

Bold print

Specifies the bold print.

See "6.11 Bold Print" for details.

byReverse

Reverse print

Specifies the reverse print.

When the value is CHAR_REVERSE_ON (2: Specify reverse print), the operation is as follows.

- Underline and bold print are not performed.
- Character right space amount is also affected.

See "6.12 Reverse Print" for details.

byRotate

Character rotation

Specifies the character rotation.

The setting is disabled in the page mode. When the setting is made, it is reflected in the print after starting the standard mode.

When the value other than "ROTATE_NONE (1: No rotation)" is specified, execute this API before sending the print data.

When the value is "ROTATE 90_TO_RIGHT (2: Rotate 90 degrees to right)" or "ROTATE_90_TO_LEFT (4: Rotate 90 degrees to left)", the operation is as follows.

- Underline print is not performed.
- The value of *byVerticalScale* is horizontal character size, and the value of *byHorizontalScale* is vertical character size.

See "6.13 Character Rotation Print" for details.

Return value

Returns 0 for success.

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

Remarks

- When either the value of *byVerticalScale* or the value of *byHorizontalScale* is "CHAR_SCALE_CURRENT (0: Current setting)", both values operate as "CHAR_SCALE_CURRENT (0: Current setting)".
- When an invalid value is specified in *byType, byVerticalScale, byHorizontalScale, byUnderline, byBold, byReverse, byRotate*, this API fails (ERR_PARAM(-90)).

MpPrintText

Prints the character string.

INT **MpPrintText**(

DWORD *dwSessionId*,

LPCTSTR *pszText*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

pszText

Text data to transmit

Specifies the text data to transmit to the printer.

Be sure to specify NULL at the end of the character string.

The available number of bytes in the string is up to 65535 including NULL.

When NULL is not specified at the end, this API fails.

Return value

Returns 0 for success.

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

Remarks

- This API is aborted by **MpReset**.
- When calling this API, the following printer commands are transmitted to the beginning of the data to transmit.
  - International Character Select
  - Character Code Table Select
  - Kanji Code System Selection
  - Kanji Mode Cancel
- The Kanji code system differs depending on the character code table.

Windows OS	Character Code Table	Kanji Code System
Japanese	Katakana	Shift-JIS code
	Other than Katakana	JIS code
Other than Japanese	Katakana	Shift-JIS code
	Other than Katakana	JIS code

- The string encoding by this API is encoded in the Codepage (CODEPAGE_XXX) selected by **MpSelectCharacterset**.
- When **MpControlTransaction** is not used, the transmission timeout period of this API is the value set by **MpGetWriteTimeout**.

MpPrintLogo

Prints the logo registered in the printer.

INT **MpPrintLogo**(
DWORD *dwSessionId*,
BYTE *byRegNum*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

byRegNum

Logo number

Specifies the logo number to print.
Specify the value among the number 0 to 99.
When a number other than above is specified, this API fails (ERR_PARAM(-90)).
When the logo image data is not registered in the specified logo number, this API is ignored.

Return value

Returns 0 for success.

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

MpSendDataFile

Sends the data of the specified file.

INT **MpSendDataFile**(
DWORD *dwSessionId*,
LPCTSTR *pszFilePath*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

pszFilePath

File path

Specifies the file to transmit.

Specify the following files.

pszFilePath	File Descriptions
Image file*1	Windows bitmap file (*.bmp) JPEG file (*.jpg / *.jpeg)
Text file	Text document (*.txt)
Binary data file	Binary data file (*.bin, *.dat, *.prn)
HTML file	HTML document (*.htm, *.html)

*1: Dithering is fixed to be enabled when registering the image file with this API.

Return value

Returns 0 for success.

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

Remarks

- This API is aborted by **MpReset**.
- When an image file is specified, the image is put into the printer command and raster bit image is printed.
- When an image file that the image is 4095 dots or more in height is specified, this API fails.
- When a text file is specified, the character string in the file is printed.
- ANSI format and UTF-8 format files are available for the text file and HTML file.
- When a text file is specified and this API is called, the following printer commands are transmitted to the beginning of the data to transmit.
  - International Character Select
  - Character Code Table Select
  - Kanji Code System Selection
  - Kanji Mode Cancel

- The Kanji code system differs depending on the character code table.

Windows OS	Character Code Table	Kanji Code System
Japanese	Katakana	Shift-JIS code
	Other than Katakana	JIS code
Other than Japanese	Katakana	Shift-JIS code
	Other than Katakana	JIS code

- When a text file is specified and this API is called, the string encoding is encoded in the Codepage (CODEPAGE_XXX) selected by **MpSelectCharacterset**.
- When an HTML file is specified and this API is called, the following command is transmitted to the beginning of the command to transmit.
  - Tag Processing Start

- When a binary data file is specified, the data in the file is transmitted to the printer without converting.
- When **MpControlTransaction** is not used, the transmission timeout period of this API is the value set by **MpGetWriteTimeout**.
- The readable file size is up to 4294967295 bytes.
  However, when the memory is not secured on the system, this API fails.

- Do not include the printer command "Hardware Reset" or "Printer Reset" in the data to send.
  Use **MpReset** when resetting the printer.

MpSendDataFileEx

Sends the data of the specified file.

INT **MpSendDataFileEx**(

DWORD *dwSessionId*,

LPCTSTR *pszFilePath,*

BYTE *byOption*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

pszFilePath

File path

Specifies the file to transmit.

Specify the following files.

pszFilePath	File Descriptions
Image file	Windows bitmap file (*.bmp) JPEG file (*.jpg / *.jpeg)
Text file	Text document (*.txt)
Binary data file	Binary data file (*.bin, *.dat, *.prn)
HTML file	HTML document (*.htm, *.html)

byOption

Option

Specifies the option when transmitting the image file.

Specify the following values.

byOption	Description
0	Dithering is enabled
1	Dithering is disabled

Return value

Returns 0 for success.

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

Remarks

- This API is aborted by **MpReset**.
- When an image file is specified, the image is put into the printer command and raster bit image is printed.
- When an image file that the image is 4095 dots or more in height is specified, this API fails.
- When a text file is specified, the character string in the file is printed.
- ANSI format and UTF-8 format files are available for the text file and HTML file.
- When a text file is specified and this API is called, the following printer commands are transmitted to the beginning of the data to transmit.
  - International Character Select
  - Character Code Table Select
  - Kanji Code System Selection
  - Kanji Mode Cancel

- The Kanji code system differs depending on the character code table.

Windows OS	Character Code Table	Kanji Code System
Japanese	Katakana	Shift-JIS code
	Other than Katakana	JIS code
Other than Japanese	Katakana	Shift-JIS code
	Other than Katakana	JIS code

- When a text file is specified and this API is called, the string encoding is encoded in the Codepage (CODEPAGE_XXX) selected by **MpSelectCharacterset**.
- When an HTML file is specified and this API is called, the following command is transmitted to the beginning of the command to transmit.
  - Tag Processing Start

- When a binary data file is specified, the data in the file is transmitted to the printer without converting.
- When **MpControlTransaction** is not used, the transmission timeout period of this API is the value set by **MpGetWriteTimeout**.
- The readable file size is up to 4294967295 bytes.
  However, when the memory is not secured on the system, this API fails.
- Do not include the printer command "Hardware Reset" or "Printer Reset" in the data to send.
  Use **MpReset** when resetting the printer.

MpPrintBarcode

Prints the barcode.

INT **MpPrintBarcode**(
DWORD *dwSessionId*,
BYTE *byType*,
LPBYTE *pData*,
DWORD *cbData*,
BYTE *byModuleWidth*,
BYTE *byModuleHeight*,
BYTE *byHRI*,
BYTE *byNWRatio*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

byType

Barcode type

Specifies the barcode type to print.
See "6.14 Barcode Type" for details.

pData

Barcode data

Specifies the barcode data to print.
Specify the string data in multi-bytecode.

cbData

Barcode data size

Specifies the size of the barcode data that has been specified.

byModuleWidth

Module width or narrow element

Specifies the module width of the barcode.
See "6.15 Barcode Module Width" for details.

byModuleHeight

Barcode height

Specifies the module height of the barcode in dots.
The range of the input value depends on the barcode type.
See "6.16 Barcode Height" for details.

byHRI

HRI characters

Specifies the character font and position of the barcode HRI characters.

See "6.17 Barcode HRI Character" for details.

byNWRatio

N:W ratio

Specifies the N:W ratio of the barcode.

It needs to be specified for the following barcode types. For the other barcode types, specify 0.

- CODE39
- ITF
- CODABAR

See "6.18 Barcode N:W Ratio" for details.

Return value

Returns 0 for success.

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

Remarks

- In the standard mode, execute this API before sending the print data.
- When an invalid value is specified in any one of *byType*, *byModuleWidth*, *byModuleHeight*, *byHRI*, or *byNWRatio*, this API fails (ERR_PARAM(-90)).
- When the barcode exceeds the print area, it is not printed.
- The input conditions of the barcode data are as follows.
  When the value that does not match the input conditions is specified, this API fails (ERR_INVALID_DATA(-1020)).

Type	Number of Data	Available Data	
UPC-A	11 to 12 bytes	'0' to '9'	(0x30 to 0x39)
UPC-E	11 to 12 bytes	'0' to '9'	(0x30 to 0x39)
JAN13 (EAN13)	12 to 13 bytes	'0' to '9'	(0x30 to 0x39)
JAN8 (EAN8)	7 to 8 bytes	'0' to '9'	(0x30 to 0x39)
CODE39	1 to 150 bytes	'0' to '9'	(0x30 to 0x39)
		'A' to 'Z'	(0x41 to 0x54)
		' '	(0x20)
		'$'	(0x24)
		'%'	(0x25)
		'+'	(0x2B)
		'-'	(0x2D)
		'.'	(0x2E)
		'/'	(0x2F)
ITF	2 to 150 bytes (Even number only)	'0' to '9'	(0x30 to 0x39)
CODABAR	1 to 150 bytes	'0' to '9'	(0x30 to 0x39)
		'A' to 'D'	(0x41 to 0x44)
		'$'	(0x24)
		'+'	(0x2B)
		'-'	(0x2D)

Type	Number of Data	Available Data	
CODABAR	1 to 150 bytes	'.'	(0x2E)
		'/'	(0x2F)
		':'	(0x3A)
CODE128	2 bytes or more		(0x00 to 0x7F) *1
CODE93	1 byte or more		(0x00 to 0x7F)
JAN13 (EAN13) add-on 2	12 to 13 + 2 bytes	'0' to '9'	(0x30 to 0x39)
JAN13 (EAN13) add-on 5	12 to 13 + 5 bytes	'0' to '9'	(0x30 to 0x39)
GS1 Databar Omni-directional	13 bytes	'0' to '9'	(0x30 to 0x39)
GS1 Databar Truncated	13 bytes	'0' to '9'	(0x30 to 0x39)
GS1 Databar Limited	13 bytes	'0' to '9'	(0x30 to 0x39)
GS1 Databar Expanded	2 bytes or more	' ' to '"'	(0x20 to 0x22)
		'%' to '?'	(0x25 to 0x3F)
		'A' to 'Z'	(0x41 to 0x5A)
		'_'	(0x5F)
		'a' to 'z'	(0x61 to 0x7A)
		'{'	(0x7B)

*1: The available data depends on CODE selection.

- CODE A: 0x00 to 0x5F
- CODE B: 0x20 to 0x7F
- CODE C: 2-digit numbers from 00 to 99 (0x00 to 0x63)

CODE, FNC and SHIFT are selected by the following special characters.

Code	CODE A	CODE B	CODE C
{S	SHIFT	SHIFT	SHIFT
{A	-	CODE A	CODE A
{B	CODE B	-	CODE B
{C	CODE C	CODE C	-
{1	FNC1	FNC1	FNC1
{2	FNC2	FNC2	FNC2
{3	FNC3	FNC3	FNC3
{4	FNC4	FNC4	FNC4
{{	'{'	'{'	-

- The check digit of the following barcodes is automatically calculated.
  - UPC-A
  - UPC-E
  - JAN13 (EAN13)
  - JAN8 (EAN8)
  - JAN13 (EAN13) add-on 2
  - JAN13 (EAN13) add-on 5
  - GS1 Databar Omni-directional
  - GS1 Databar Truncated
  - GS1 Databar Limited
- When the data including check digit is specified for the following barcodes, the check digit data is ignored and recalculated on the printer.
  - UPC-A
  - UPC-E
  - JAN13 (EAN13)
  - JAN8 (EAN8)

- Specifying the start code and stop code ('*') of CODE39 is not required since the codes are added automatically.
- When an odd number of the data is specified for the ITF barcode data, this API fails.
- In the CODABAR barcode data, enter 'A' to 'D' as the start code and stop code.
- In the GS1 Databar Expanded barcode data, use '{' only for specifying FNC. When specifying FNC1, enter '{', '1'(0x7B, 0x31).
- When **MpControlTransaction** is not used, the transmission timeout period of this API is the value set by **MpGetWriteTimeout**.

MpPrint2Dcode

Prints the 2-dimensional barcode.

INT **MpPrint2Dcode**(
DWORD *dwSessionId*,
BYTE *byType*,
LPBYTE *pData*,
DWORD *cbData*,
BYTE *byMode*,
BYTE *byModuleSize*,
BYTE *byModuleHeight*,
BYTE *byColumn*,
BYTE *byRow*,
BYTE *byErrCorrect*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

byType

2-dimensional barcode type

Specifies the 2-dimensional barcode type to print.
See "6.19 2-Dimensional Barcode Type" for details.

pData

2-dimensional barcode data

Specifies the barcode data to print.
Specify the string data in multi-bytecode.

cbData

2-dimensional barcode data size

Specifies the barcode data size that has been specified.

byMode

2-dimensional barcode mode

Specifies the 2-dimensional barcode mode.

It needs to be specified for the following 2-dimensional barcode types. When using other 2-dimensional barcode types, specify 0.

- QR Code
- PDF417
- Data Matrix
- Maxi Code

The range of the input value depends on the 2-dimensional barcode type.

See "6.20 2-Dimensional Barcode Mode" for details.

byModuleSize

Module size of the 2-dimensional barcode

Specifies the module size of the 2-dimensional barcode.

It needs to be specified for the following 2-dimensional barcode types. When using other 2-dimensional barcode types, specify 0.

- QR Code
- PDF417
- Data Matrix
- GS1 Databar Stacked
- GS1 Databar Stacked Omni-directional
- GS1 Databar Expanded Stacked

The range of the input value depends on the 2-dimensional barcode type.

See "6.21 2-Dimensional Barcode Module Size" for details.

byModuleHeight

Module height of the 2-dimensional barcode

Specifies the module height of the 2-dimensional barcode.

It needs to be specified for the following 2-dimensional barcode types. When using other 2-dimensional barcode types, specify 0.

- PDF417
- GS1 Databar Stacked Omni-directional

The range of the input value depends on the 2-dimensional barcode type.

See "6.22 2-Dimensional Barcode Module Height" for details.

byColumn

Number of 2-dimensional barcode columns or elements in 1 line

Specifies the number of 2-dimensional barcode columns or elements in 1 line.

It needs to be specified for the following 2-dimensional barcode types. When using other 2-dimensional barcode types, specify 0.

- PDF417
- GS1 Databar Expanded Stacked

The range of the input value depends on the 2-dimensional barcode type.

See "6.23 Number of 2-Dimensional Barcode Columns" for details.

byRow

Number of 2-dimensional barcode rows

Specifies the number of 2-dimensional barcode rows.

It needs to be specified for the following 2-dimensional barcode type. When using other 2-dimensional barcode types, specify 0.

- PDF417

See "6.24 Number of 2-Dimensional Barcode Rows" for details.

byErrCorrect

2-dimensional barcode error correction level

Specifies the error correction level of 2-dimensional barcode.

It needs to be specified for the following 2-dimensional barcode type. When using other 2-dimensional barcode types, specify 0.

- QR Code
- PDF417

The range of the input value depends on the 2-dimensional barcode type.

See "6.25 2-Dimensional Barcode Error Correction Level" for details.

Return value

Returns 0 for success.

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

Remarks

- When an invalid value is specified in any one of *byType*, *byMode*, *byModuleSize*, *byModuleHeight*, *byColumn*, *byRow* and *byErrCorrect*, this API fails (ERR_PARAM(-90)).
- When the barcode exceeds the print area, it is not printed.
- When **MpControlTransaction** is not used, the transmission timeout period of this API is the value set by **MpGetWriteTimeout**.

MpPrintPageModeRectangle

Prints the rectangle when page mode is selected.

INT **MpPrintPageModeRectangle**(

DWORD *dwSessionId*,

WORD *wStartX*,

WORD *wStartY*,

WORD *wWidth*,

WORD *wHeight*,

BYTE *byThickness*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

wStartX

Horizontal start point

Specifies the horizontal start point of the rectangle in dots.

wStartY

Vertical start point

Specifies the vertical start point of the rectangle in dots.

wWidth

Horizontal length

Specifies the horizontal length of the rectangle in dots.

Specify 1 or larger number.

wHeight

Vertical length

Specifies the vertical length of the rectangle in dots.

Specify 1 or larger number.

byThickness

Line width

Specifies the line width of the rectangle in dots.

The setting range is from 2 to 40.

When the specified value is less than 2, the value is set to 2, and when the specified value exceeds 40, the value is set to 40.

Return value

Returns 0 for success.

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

Remarks

- When using this API in the standard mode, the operation is undefined.
- The print direction of the rectangle depends on the setting of **MpSelectPageMode**.
- When using this API, the printer command "Ruled Line Buffer Clear" and " Ruled Line OFF" are automatically executed.

MpFeedLine

Feeds the paper by line.

INT **MpFeedLine**(
DWORD *dwSessionId*,
BYTE *byLines*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

byLines

Paper feed length by line

Specifies the paper feed length by line.
The setting range is from 0 to 255.

Return value

Returns 0 for success.

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

MpFeedDotLine

Feeds the paper by dot.

INT **MpFeedDotLine**(
DWORD *dwSessionId*,
INT *nDotLines*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

nDotLines

Paper feed length by dot

Specifies the paper feed length by dot.
The setting range is from -48 to 8192.
When the specified value is less than -48, the value is set to -48, and when the specified value exceeds 8192, the value is set to 8192.

Return value

Returns 0 for success.

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

Remarks

- When a negative value is specified for *nDotLines* of this API in the page mode, paper feed is not performed.

MpFeedCutPosition

Feeds the paper to the paper cut position.

NT **MpFeedCutPosition**(
DWORD *dwSessionId*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

Return value

Returns 0 for success.

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

Remarks

- In the standard mode, execute this API before sending the print data.

MpFeedMarkPosition

Performs form feed to the index mark.

INT **MpFeedMarkPosition**(
DWORD *dwSessionId*,
INT *nDotLines*)

Parameters

dwSessionId

Session ID

Specifies the session ID retrieved by **MpOpen**.

nDotLines

Correction amount

Specifies the correction amount in dots.
The setting range is from -48 to 255.
When the specified value is less than -48, the value is set to -48, and when the specified value exceeds 255, the value is set to 255. Specify 0 when no correction is conducted.

Return value

Returns 0 for success.

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

Remarks

- The command performs marked paper form feed and corrects the form feed position.
- In the standard mode, execute this API before sending the print data.

Chapter 4 .NET API

4.1 Overview

This chapter describes the SDK for .NET development environment (.NET API).

4.2 Development Language

The following development languages are covered.

- Visual C#
- Visual Basic

4.3 Library File

The library file has the following file name.

- SiiMpa4ClassLib.dll

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

4.4 API List

The following APIs are implemented in the SDK.

- Namespace: SII.SPS.CommunicationLibrary.MobilePrinter

4.4.1 Printer Manager Class

Category	API	Function Summary
Property	**IsOpened**	Retrieves the call status of **Open**.
Method	**Open**	Begins using the SDK in the specified printer.
Method	**Close**	Ends using the SDK.
Method	**SetWriteTimeout**	Specifies the timeout value of API to transmit data.
Method	**GetWriteTimeout**	Retrieves the timeout value of API to transmit data.
Method	**SetResponseTimeout**	Specifies the timeout value of API to receive data.
Method	**GetResponseTimeout**	Retrieves the timeout value of API to receive data.
Method	**Write**	Sends the binary data to the printer.
Method	**Read**	Retrieves the data received from the printer.
Method	**GetReadSize**	Retrieves the data size available by Read method currently.
Method	**WriteAndWaitResponse**	Receives the response data after sending the binary data.
Method	**Reset**	Resets the printer.
Method	**GetStatus**	Retrieves the latest printer status.
Method	**StartCallbackFunction**	Registers the callback function that is called when detecting the printer status change.
Method	**StopCallbackFunction**	Ends the callback of when the printer status change has been detected.
Method	**RegisterStyleSheet**	Registers the style sheet in the printer.
Method	**DeleteStyleSheet**	Deletes the style sheet from the printer.
Method	**RegisterLogo**	Registers the logo in the printer.
Method	**DeleteLogo**	Deletes the registered logo from the printer.
Method	**GetPrinterInformation**	Retrieves the specified printer information.
Method	**ControlTransaction**	Processes target APIs in a batch.
Method	**SelectStandardMode**	Begins the standard mode.

Category	API	Function Summary
Method	**SelectPageMode**	Specifies the print area and print direction in the page mode and begins the page mode.
Method	**PrintPageModeData**	Prints the page mode data.
Method	**SetPageModeVerticalPosition**	Specifies vertical absolute position in the page mode.
Method	**SetStandardModeArea**	Specifies the print area in the standard mode.
Method	**SetStandardModeAlignment**	Specifies the print position in the standard mode.
Method	**SetHorizontalPosition**	Specifies the absolute position.
Method	**SetStandardModeBarcodeDirection**	Specifies the print direction of barcode.
Method	**SetLineSpacing**	Specifies the line spacing.
Method	**SetCharacterRightSpace**	Specifies the character right space amount.
Method	**SelectCharacterSet**	Sets the character set.
Method	**SelectInternationalCharacterSet**	Sets the international character set.
Method	**SetCharacterFormatting**	Specifies the character formatting.
Method	**PrintText**	Prints the character string.
Method	**PrintLogo**	Prints the image registered in the printer.
Method	**SendDataFile**	Sends the data of the specified file.
Method	**PrintBarcode**	Prints the barcode.
Method	**Print2Dcode**	Prints the 2-dimensional barcode.
Method	**PrintPageModeRectangle**	Prints the rectangle when selecting the page mode.
Method	**FeedLine**	Feeds the paper by line.
Method	**FeedDotLine**	Feeds the paper by dot.
Method	**FeedCutPosition**	Feeds the paper to paper cut position.
Method	**FeedMarkPosition**	Performs marked paper form feed.
Event	**statusChanged**	The event that notifies the responded printer status.

4.4.2 **Printer Exception Class**

Category	API	Function Summary
Method	**GetErrorCode**	Retrieves the error code.

4.5 API Details (Printer Manager Class)

4.5.1 Property

IsOpened

Retrieves the call status of **Open**.

bool **IsOpened** { get; }

Initial value

FALSE

Remarks

Retrieves the open status of the SDK.

- TRUE : **Open** has been already called.
- FALSE : **Open** is not called.

4.5.2 **Method**

Open

Begins using the SDK to the specified printer.

void **Open**(

string *prnName*)

Parameters

prnName

Name of the printer that uses the SDK

Specifies the printer name (friendly name) to output.

Return value

None

Remarks

- Only 1 instance can be opened per process.
- When the SDK is not used, be sure to call **Close**.
- When the printer driver connects other than USB, Bluetooth or TCP/IP, this API fails.
- When an error occurs during calling this API, **PrinterException** is thrown.

Close

Ends using the SDK.

void **Close**()

Parameters

None

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpClose** in "Chapter 3 Win32 API" for details.

SetWriteTimeout

Specifies the timeout value of the API to transmit data.

void **SetWriteTimeout**(

uint *timeout*)

Parameters

timeout

Data transmission timeout value

Specifies the waiting time of the API to transmit data (in milliseconds).

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpSetWriteTimeout** in "Chapter 3 Win32 API" for details.

GetWriteTimeout

Retrieves the timeout value of the API to transmit data.

uint **GetWriteTimeout**()

Parameters

None

Return value

Data transmission timeout value

Retrieves the waiting time of the API to transmit data (in milliseconds).

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpGetWriteTimeout** in "Chapter 3 Win32 API" for details.

SetResponseTimeout

Specifies the timeout value of the API to receive data.

void **SetResponseTimeout**(

uint *timeout*)

Parameters

timeout

Data reception timeout value

Specifies the waiting time of the API to receive data (in milliseconds).

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpSetResponseTimeout** in "Chapter 3 Win32 API" for details.

GetResponseTimeout

Retrieves the timeout value of the API to receive data.

uint **GetResponseTimeout**()

Parameters

None

Return value

Data reception timeout value

Retrieves the waiting time of the API to receive data (in milliseconds).

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpGetResponseTimeout** in "Chapter 3 Win32 API" for details.

Write

Sends the binary data to the printer.

void **Write**(

byte[] *cmd*,

uint *offset*)

Parameters

cmd

Transmit data

Specifies the data to send to the printer.

offset

Offset of data to send

Specifies the starting position of the data to send.

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpWrite** in "Chapter 3 Win32 API" for details.

Read

Retrieves the data received from the printer.

byte[] **Read**()

Parameters

None

Return value

Receive data

Retrieves the data received from the printer.

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpRead** in "Chapter 3 Win32 API" for details.

GetReadSize

Retrieve the data size available by **Read** method.

uint **GetReadSize**()

Parameters

None

Return value

Available data size to retrieve

Retrieves the receive data size available by **Read** method.

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpRead** in "Chapter 3 Win32 API" for details.

WriteAndWaitResponse

Receives the response data after sending binary data.

byte[] **WriteAndWaitResponse**(

byte[] *cmd*,

uint *respSize*,

bool *respFlag*,

bool *incASBData*)

Parameters

cmd

Transmit data

Specifies the data to send to the printer.

respSize

Maximum data size to receive

Specifies the maximum value of the data size to receive.

respFlag

Receive operation flag

Specifies the receive operation.

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

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

incASBData

Specifies whether the responses for the ASB setting command is included in the receive data.

Specifies the data to receive.

TRUE : Retrieves the data including responses for the ASB setting command.

FALSE : Retrieves the data excluding responses for the ASB Setting command.

Return value

Receive data

Retrieves the data received from the printer.

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpWriteAndWaitResponse** in "Chapter 3 Win32 API" for details.

Reset

Resets the printer.

void **Reset**()

Parameters

None

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpReset** in "Chapter 3 Win32 API" for details.

GetStatus

Retrieves the latest printer status.

PrinterStatus **GetStatus**()

Parameters

None

Return value

Printer status

Retrieves the latest printer status.

Remarks

- See "6.1 Printer Status List" for details of "PrinterStatus".
- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpGetStatus** in "Chapter 3 Win32 API" for details.

StartCallbackFunction

Registers the callback function that is called when detecting the printer status change.

void **StartCallbackFunction**()

Parameters

None

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpSetCallbackFunction** in "Chapter 3 Win32 API" for details.

StopCallbackFunction

Ends the callback of when the printer status change has been detected.

void **StopCallbackFunction**()

Parameters

None

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpSetCallbackFunction** in "Chapter 3 Win32 API" for details.

RegisterStyleSheet

Registers the style sheet in the printer.

void **RegisterStyleSheet**(

string *filePath*,

byte *regNum*)

Parameters

filePath

File path of the style sheet

Specifies the file path of the style sheet to be registered in the printer.

regNum

Style sheet number

Specifies the style sheet number to be registered in the printer.

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpRegisterStyleSheet** in "Chapter 3 Win32 API" for details.

DeleteStyleSheet

Deletes the registered style sheet from the printer.

void **DeleteStyleSheet**(

byte *regNum*)

Parameters

regNum

Style sheet number

Specifies the style sheet number to be deleted from the printer.

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpDeleteStyleSheet** in "Chapter 3 Win32 API" for details.

RegisterLogo

Registers the logo in the printer.

void **RegisterLogo**(

string *filePath*,

byte *regNum*)

Parameters

filePath

File path of the logo image data

Specifies the file path of the logo data to be registered in the printer.

Dithering is fixed to be enabled when registering the logo image data with this API.

regNum

Logo number

Specifies the logo number to be registered in the printer.

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpRegisterLogo** in "Chapter 3 Win32 API" for details.

RegisterLogoEx

Registers the logo in the printer.

void **RegisterLogoEx**(

string *filePath,*

byte *regNum,*

byte *option*)

Parameters

filePath

File path of the logo image data

Specifies the file path of the logo to be registered in the printer.

regNum

Logo number

Specifies the logo number to be registered in the printer.

option

Option

Specifies the option when registering the logo in the printer.

Specify the following values.

option	Description
0	Dithering is enabled
1	Dithering is disabled

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpRegisterLogoEx** in "Chapter 3 Win32 API" for details.

DeleteLogo

Deletes the registered logo from the printer.

void **DeleteLogo**(
byte *regNum*)

Parameters

regNum

Logo number

Specifies the logo number to be deleted from the printer.

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpDeleteLogo** in "Chapter 3 Win32 API" for details.

GetPrinterInformation

Retrieves the specified printer information.

byte[] **GetPrinterInformation**(
PrinterInformation *prnInfo*,
ref uint *data*)

Parameters

prnInfo

Printer information

Specifies the type of the printer information to retrieve.
Specifies the value defined in "PrinterInformation".

data

Receive data (numerical value)

When the response format of *prnInfo* is numerical value, the value is also stored in *data* by uint in addition to the byte array data.

Return value

Receive data (binary data)

Retrieves the received font information.

Remarks

- See "6.2 Printer Information" for details of "PrinterInformation".
- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpGetPrinterInformation** in "Chapter 3 Win32 API" for details.

ControlTransaction

Processes the target APIs in a batch.

void **ControlTransaction**(

TransactionFunction *control*)

Parameters

control

Control method

Specifies the control method for batch process.

Specifies the value defined in "TransactionFunction".

Return value

None

Remarks

- See "6.3 Batch Process Control Method" for details of "TransactionFunction".
- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpControlTransaction** in "Chapter 3 Win32 API" for details.

SelectStandardMode

Begins the standard mode.

void **SelectStandardMode**()

Parameters

None

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpSelectStandardMode** in "Chapter 3 Win32 API" for details.

SelectPageMode

Specifies the print area and print direction in the page mode and begins the page mode.

void **SelectPageMode**(

ushort *startX*,

ushort *startY*,

ushort *width*,

ushort *height*,

Direction *direction*)

Parameters

startX

Horizontal start point

Specifies the horizontal start point in the print area of the page mode in dots.

startY

Vertical start point

Specifies the vertical start point in the print area of the page mode in dots.

width

Horizontal length

Specifies the horizontal length in the print area of the page mode in dots.

height

Vertical length

Specifies the vertical length in the print area of the page mode in dots.

direction

Print direction

Specifies the print direction in the page mode.

Specifies the value defined in "Direction".

Return value

None

Remarks

- See "6.4 Print Direction" for details of "Direction".
- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpSelectPageMode** in "Chapter 3 Win32 API" for details.

PrintPageModeData

Prints the page mode data.

void **PrintPageModeData**()

Parameters

None

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpPrintPageModeData** in "Chapter 3 Win32 API" for details.

SetPageModeVerticalPosition

Sets the vertical absolute position in the page mode.

void **SetPageModeVerticalPosition**(

ushort *verticalPosition*)

Parameters

verticalPosition

Absolute mapping starting position in a vertical direction

Specifies the absolute mapping starting position in a vertical direction in dots in the page mode.

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpSetPageModeVerticalPosition** in "Chapter 3 Win32 API" for details.

SetStandardModeArea

Sets the print area in the standard mode.

void **SetStandardModeArea**(

ushort *leftMargin*,

ushort *printAreaWidth*)

Parameters

leftMargin

Left margin position

Specifies the left margin position in dots.

printAreaWidth

Print area width

Specifies the print area width in dots.

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpSetStandardModeArea** in "Chapter 3 Win32 API" for details.

SetStandardModeAlignment

Sets the print position in the standard mode.

void **SetStandardModeAlignment**(

Alignment *align*)

Parameters

align

Print position

Specifies the print position when printing in the standard mode.

Specifies the value defined in "Alignment".

Return value

None

Remarks

- See "6.5 Print Position" for details of "Alignment".
- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpSetStandardModeAlignment** in "Chapter 3 Win32 API" for details.

SetHorizontalPosition

Sets the absolute position.

void **SetHorizontalPosition**(

ushort *horizontalPosition*)

Parameters

horizontalPosition

Horizontal print starting position

Specifies the horizontal print starting position in dots.

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpSetHorizontalPosition** in "Chapter 3 Win32 API" for details.

SetStandardModeBarcodeDirection

Sets the print direction of the barcode.

void **SetStandardModeBarcodeDirection**(

Rotate *rotate*)

Parameters

rotate

Print direction

Specifies the print direction of the barcode.

Specifies the value defined in "Rotate".

Return value

None

Remarks

- See "6.13 Character Rotation Print" for details of "Rotate".
- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpSetStandardModeBarcodeDirection** in "Chapter 3 Win32 API" for details.

SetLineSpacing

Sets the line spacing.

void **SetLineSpacing**(

byte *lineSpacing*)

Parameters

lineSpacing

Line spacing

Specifies the line spacing in dots.

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpSetLineSpacing** in "Chapter 3 Win32 API" for details.

SetCharacterRightSpace

Sets the character right space amount.

void **SetCharacterRightSpace**(

byte *space*)

Parameters

space

Character right space amount

Specifies the character right space amount in dots.

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpSetCharacterRightSpace** in "Chapter 3 Win32 API" for details.

SelectCharacterSet

Sets the character set.

void **SelectCharacterSet**(

CharacterSet *charSet*)

Parameters

charSet

Character set

Specifies the character set.

Specifies the value defined in "CharacterSet".

Return value

None

Remarks

- See "6.6 Character Set" for details of "CharacterSet".
- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpSelectCharacterSet** in "Chapter 3 Win32 API" for details.

SelectInternationalCharacterSet

Sets the international character set.

void **SelectInternationalCharacterSet**(

InternationalCharacterSet *intCharSet*)

Parameters

intCharSet

International character set

Specifies the international character set to use.

Specifies the value defined in "InternationalCharacterSet".

Return value

None

Remarks

- See "6.7 International Character Set" for details of "InternationalCharacterSet".
- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpSelectInternationalCharacterSet** in "Chapter 3 Win32 API" for details.

SetCharacterFormatting

Sets the character formatting.

void **SetCharacterFormatting**(
CharacterType *type*,
CharacterScale *verticalScale*,
CharacterScale *horizontalScale*,
Underline *underline*,
Bold *bold*,
Reverse *reverse*,
Rotate *rotate*)

Parameters

type

Character font

Specifies the character font.
Specifies the value defined in "CharacterType".

verticalScale

Character size (vertical direction)

Specifies the character size (vertical direction).
Specifies the value defined in "CharacterScale".

horizontalScale

Character size (horizontal direction)

Specifies the character size (horizontal direction).
Specifies the value defined in "CharacterScale".

underline

Underline

Specifies the underline.
Specifies the value defined in "Underline".

bold

Bold print

Specifies the bold print.
Specifies the value defined in "Bold".

reverse

Reverse print

Specifies the reverse print.

Specifies the value defined in "Reverse".

rotate

Character rotation

Specifies the character rotation.

Specifies the value defined in "Rotate".

Return value

None

Remarks

- See "6.8 Character Font" for details of "CharacterType".
- See "6.9 Character Size" for details of "CharacterScale".
- See "6.10 Underline" for details of "Underline".
- See "6.11 Bold Print" for details of "Bold".
- See "6.12 Reverse Print" for details of "Reverse".
- See "6.13 Character Rotation Print" for details of "Rotate".
- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpSetCharacterFormatting** in "Chapter 3 Win32 API" for details.

PrintText

Prints the character string.

void **PrintText**(

string *text*)

Parameters

text

Text data to transmit

Specifies the text data to transmit to the printer.

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpPrintText** in "Chapter 3 Win32 API" for details.

PrintLogo

Prints the image registered in the printer.

void **PrintLogo**(

byte *regNum*)

Parameters

regNum

Logo number

Specifies the registered logo number in the printer.

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpPrintLogo** in "Chapter 3 Win32 API" for details.

SendDataFile

Transmits data of the specified file.

void **SendDataFile**(

string *filePath*)

Parameters

filePath

File path

Specifies the file that includes data to transmit.

Dithering is fixed to be enabled when registering the image file with this API.

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpSendDataFile** in "Chapter 3 Win32 API" for details.

SendDataFileEx

Sends the data of the specified file.

void **SendDataFileEx**(

string *filePath,*

byte *option*)

Parameters

filePath

File path

Specifies the file to transmit.

option

Option

Specifies the option when transmitting the image file.

Specify the following values.

option	Description
0	Dithering is enabled
1	Dithering is disabled

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpSendDataFileEx** in "Chapter 3 Win32 API" for details.

PrintBarcode

Prints the barcode.

void **PrintBarcode**(
TypeBarcode *type*,
byte[] *data*,
ModuleWidthBarcode *moduleWidth*,
byte *moduleHeight*,
HriPositionBarcode *hri*,
NwRatioBarcode *nwRatio*)

Parameters

type

Barcode type

Specifies the barcode type to print.
Specifies the value defined in "TypeBarcode".
See "6.14 Barcode Type" for details.

data

Barcode data

Specifies the barcode data to print.

moduleWidth

Module width or narrow element

Specifies the barcode module width.
Specifies the value defined in "ModuleWidthBarcode".
See "6.15 Barcode Module Width" for details.

moduleHeight

Barcode height

Specifies the barcode module height in dots.
See "6.16 Barcode Height" for details.

hri

HRI character

Specifies the font and position of the barcode HRI character.
Specifies the value defined in "HriPositionBarcode".
See "6.17 Barcode HRI Character" for details.

nwRatio

N:W ratio

Specifies the barcode N:W ratio.

Specifies the value defined in "NwRatioBarcode".

See "6.18 Barcode N:W Ratio" for details.

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpPrintBarcode** in "Chapter 3 Win32 API" for details.

Print2Dcode

Prints the 2-dimensional barcode.

void **Print2Dcode**(

Type2Dcode *type*,

byte[] *data*,

Mode2Dcode *mode*,

ModuleSize2Dcode *moduleSize*,

byte *moduleHeight*,

byte *column*,

byte *row*,

ErrorCorrect2Dcode *errCorrect*)

Parameters

type

2-dimensional barcode type

Specifies the 2-dimensional barcode type to print.

Specifies the value defined in "Type2Dcode".

See "6.19 2-Dimensional Barcode Type" for details.

data

2-dimensional barcode data

Specifies the 2-dimensional barcode data to print.

mode

2-dimensional barcode mode

Specifies the 2-dimensional barcode mode.

Specifies the value defined in "Mode2Dcode".

See "6.20 2-Dimensional Barcode Mode" for details.

moduleSize

2-dimensional barcode module size

Specifies the2-dimensional barcode module size.

Specifies the value defined in "ModuleSize2Dcode".

See "6.21 2-Dimensional Barcode Module Size" for details.

moduleHeight

2-dimensional barcode module height

Specifies the 2-dimensional barcode module height.

See "6.22 2-Dimensional Barcode Module Height" for details.

column

Number of 2-dimensional barcode columns or elements in 1 line

Specifies the number of 2-dimensional barcode columns or elements in 1 line.

See "6.23 Number of 2-Dimensional Barcode Columns" for details.

row

Number of 2-dimensional barcode rows

Specifies the number of 2-dimensional barcode rows.

See "6.24 Number of 2-Dimensional Barcode Rows" for details.

errCorrect

Error correction level

Specifies the error correction level of 2-dimensional barcode.

Specifies the value defined in "ErrorCorrect2Dcode".

See "6.25 2-Dimensional Barcode Error Correction Level" for details.

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpPrint2Dcode** in "Chapter 3 Win32 API" for details.

PrintPageModeRectangle

Prints the rectangle when the page mode is selected.

void **PrintPageModeRectangle**(

ushort *startX*,

ushort *startY*,

ushort *width*,

ushort *height*,

byte *thickness*)

Parameters

startX

Horizontal start point

Specifies the horizontal start point of the rectangle in dots.

startY

Vertical start point

Specifies the vertical start point of the rectangle in dots.

width

Horizontal length

Specifies the horizontal length of the rectangle in dots.

height

Vertical length

Specifies the vertical length of the rectangle in dots.

thickness

Line width

Specifies the line width of the rectangle in dots.

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpPrintPageModeRectangle** in "Chapter 3 Win32 API" for details.

FeedLine

Feeds the specified length of paper by line.

void **FeedLine**(

byte *lines*)

Parameters

lines

Paper feed length by line

Specifies the paper feed length by line.

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpFeedLine** in "Chapter 3 Win32 API" for details.

FeedDotLine

Feeds the specified length of paper by dot.

void **FeedDotLine**(

int *dotLines*)

Parameters

dotLines

Paper feed length by dot

Specifies the paper feed length by dot.

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpFeedDotLine** in "Chapter 3 Win32 API" for details.

FeedCutPosition

Feeds the paper to the paper cut position.

void **FeedCutPosition**()

Parameters

None

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpFeedCutPosition** in "Chapter 3 Win32 API" for details.

FeedMarkPosition

Performs form feed to the index mark.

void **FeedMarkPosition**(

int *dotLines*)

Parameters

dotLines

Correction amount

Specifies the correction amount in dots.

Return value

None

Remarks

- When an error occurs during calling this API, **PrinterException** is thrown.
- See **MpFeedMarkPosition** in "Chapter 3 Win32 API" for details.

4.5.3 **Event**

statusChanged

The event that notifies the responded printer status.

event StatusCallbackHandler **statusChanged**

delegate void **StatusCallbackHandler**(PrinterStatus *status*)

Parameters

status

Printer status

Specifies the variable to store the printer status.

Remarks

- See "6.1 Printer Status List" for details of "PrinterStatus".
- See **MpSetCallbackFunction** in "Chapter 3 Win32 API" for details.

4.6 API Details (Printer Exception Class)

4.6.1 Method

GetErrorCode

Retrieves the error code.

ErrorCode **GetErrorCode**()

Parameters

None

Return value

Error code

Retrieves the error code for thrown exception.

Remarks

- See "Chapter 5 Error Code List" for details.

Chapter 5 Error Code List

5.1 Error Code List

Major error codes are as follows:

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_OPENED	-20	Specified printer has already been opened.
ERR_NO_PRN	-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 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_NOT_SII	-120	Printer driver is not supported.
ERR_DISK_FULL	-170	Printer is busy state.
ERR_ENTRY_OVER	-190	Maximum processing capacity is exceeded.
ERR_EXIST	-210	Existing module is called.
ERR_NOT_FOUND	-220	File cannot be found. Or, it may not be registered.
ERR_WORKAREA_NO_MEMORY	-260	Specified memory size is insufficient.
ERR_WORKAREA_FAILED	-280	Memory cannot be reserved.
ERR_EXEC_FUNC	-310	Function is not available because it is being used by other thread or process.
ERR_SPL_NOT_EXIST	-350	Spooler service or SII Printer Service Software has not been started.
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 is canceled.
ERR_PRN_HAS_JOBS_QUEUED	-1060	Printer has queued job.
ERR_UNKNOWN_PORT	-1070	Port is not supported.
ERR_INVALID_PRN_STATE	-1080	Printer status is abnormal.

Macro Definition (Constant)	Value	Description
ERR_BAD_ENVIRONMENT	-1090	Printer driver installation may be abnormal.
ERR_PORT_NOT_OPENED	-1120	Port is not opened.
ERR_DATA_SIZE_TOOBIG	-1130	Maximum data size is exceeded.
ERR_ENCODE_FAILED	-1140	Error has occurred in encoding text data.
ERR_FILE_USED	-1150	Specified file is being used by other process.
ERR_FILE_INVALID	-1160	Specified file is invalid. (File extension or file construction is invalid)
ERR_OVER_STYLE_NUM	-1170	Number of styles registered in specified file exceeds specified value (64).
ERR_TRANSACTION_NOT_STARTED	-1180	Batch process is not started.
ERR_IMAGE_DATA_CONVERT_FAILED	-1190	Failed to convert image data.
ERR_FUNCTION_NOT_CALLED	-1200	API could not be executed.

Chapter 6 Argument Information

6.1 Printer Status List

The followings are responses for the printer status.

Printer Status		Response Value (Win32 API)	PrinterStatus Struct Member (.NET API)
Voltage error	True	*pdwStatus & 0x00000001≠0	Err_Voltage = TRUE
	False	*pdwStatus & 0x00000001 = 0	Err_Voltage = FALSE
Hardware error / voltage initialization error	True	*pdwStatus & 0x00000002≠0	Err_Hardware = TRUE
	False	*pdwStatus & 0x00000002 = 0	Err_Hardware = FALSE
Head temperature error	True	*pdwStatus & 0x00000004≠0	Err_HeadTemperature = TRUE
	False	*pdwStatus & 0x00000004 = 0	Err_HeadTemperature = FALSE
Out-of-paper error	True	*pdwStatus & 0x00000010≠0	Err_OutOfPaper = TRUE
	False	*pdwStatus & 0x00000010 = 0	Err_OutOfPaper = FALSE
Marked paper jam error	True	*pdwStatus & 0x00000040≠0	Err_MarkPaperJam = TRUE
	False	*pdwStatus & 0x00000040 = 0	Err_MarkPaperJam = FALSE
Paper cover open error	True	*pdwStatus & 0x00000080≠0	Err_CoverOpen = TRUE
	False	*pdwStatus & 0x00000080 = 0	Err_CoverOpen = FALSE
FEED switch state	ON	*pdwStatus & 0x00000100≠0	State_FeedSwitch = TRUE
	OFF	*pdwStatus & 0x00000100 = 0	State_FeedSwitch = FALSE
Paper feed state	Feeding	*pdwStatus & 0x00000400≠0	State_PaperFeed = TRUE
	Stop feeding	*pdwStatus & 0x00000400 = 0	State_PaperFeed = FALSE
Recovery waiting state	True	*pdwStatus & 0x00000800≠0	State_ReturnWaiting = TRUE
	False	*pdwStatus & 0x00000800 = 0	State_ReturnWaiting = FALSE
FLASH memory rewriting	True	*pdwStatus & 0x00010000≠0	State_FlashMemoryRewriting = TRUE
	False	*pdwStatus & 0x00010000 = 0	State_FlashMemoryRewriting = FALSE

Printer Status		Response Value (Win32 API)	PrinterStatus Struct Member (.NET API)
Battery remaining capacity	No battery	*pdwStatus & 0x00700000 = 0x00000000	State_Battery = 0
	Level 4 or 5	*pdwStatus & 0x00700000 = 0x00100000	State_Battery = 1
	Level 3	*pdwStatus & 0x00700000 = 0x00200000	State_Battery = 2
	Level 2	*pdwStatus & 0x00700000 = 0x00300000	State_Battery = 3
	Level 1	*pdwStatus & 0x00700000 = 0x00400000	State_Battery = 4
Battery error	True	*pdwStatus & 0x00800000≠0	Err_Battery = TRUE
	False	*pdwStatus & 0x00800000 = 0	Err_Battery = FALSE
No connection / No response *1 *2	Off line	*pdwStatus = 0	State_Err_Offline = TRUE
	On line	*pdwStatus≠0	State_Err_Offline = FALSE

*1: Extended status to the responses for the ASB Setting command.
When the status is "No connection / No response", all bits will be 0 for Win32 API, the response will be FALSE or 0 for .NET API except for State_Battery.
*2: For wireless communication, detection of "No connection / No response" may take some time.

6.2 Printer Information

The followings are the printer information and response format available to specify.

Printer Information (PrinterInformation)		Description	Response Format
Member	Value		
GET_NV_MEM_CAP	0	NV graphics memory capacity	Numerical value
GET_NV_MEM_REM_CAP	1	NV graphics memory remaining capacity	Numerical value
GET_NV_MEM_KEYCODE_LIST	2	Defined NV graphics key code list	Binary
GET_REM_USER_MEM_CAP_DEFRAG	3	Remaining user area after defragment	Numerical value
GET_REM_USER_MEM_CAP	4	Remaining user area response	Binary
GET_FUNC_SET_RESP	10	Function setting response	Binary
GET_PRN_ID_MODEL	50	Printer ID (Model ID)	Binary
GET_PRN_ID_TYPE	51	Printer ID (Type ID)	Binary
GET_PRN_ID_ROM_VER	52	Printer ID (ROM version ID)	Binary
GET_PRN_ID_FIRM_VER_MAIN	53	Printer ID (Firmware version (main))	Binary
GET_PRN_ID_MFR	54	Printer ID (Manufacturer)	Binary
GET_PRN_ID_MODEL_NAME	55	Printer ID (Model name)	Binary
GET_PRN_ID_FIRM_VER_BOOT	56	Printer ID (Firmware version (boot))	Binary
GET_PRN_ID_FIRM_CHECKSUM_BOOT	57	Printer ID (Firmware checksum (boot))	Numerical value
GET_PRN_ID_FIRM_CHECKSUM_MAIN	58	Printer ID (Firmware checksum (main))	Numerical value
GET_PRN_ID_FIRM_CHECKSUM	59	Printer ID (Firmware checksum (main + boot))	Numerical value
GET_MAINT_NUM_FEED_LINE	100	Maintenance counter (Paper feed line count (in 100 dot-lines))	Numerical value
GET_MAINT_NUM_HEAD_ACTIVE	101	Maintenance counter (Number of thermal head activation times (in 100 dot-lines))	Numerical value
GET_MAINT_DRIVE_TIME	102	Maintenance counter (Drive time of printer mechanism (minute))	Numerical value

Printer Information (PrinterInformation)		Description	Response Format
Member	Value		
GET_MAINT_NUM_FEED_LINE_INTEGRATION	103	Maintenance counter (Paper feed line count (in 100 dot-lines) (integrated value))	Numerical value
GET_MAINT_NUM_HEAD_ACTIVE_INTEGRATION	104	Maintenance counter (Number of thermal head activation times (in 100 dot-lines) (integration))	Numerical value
GET_MAINT_DRIVE_TIME_INTEGRATION	105	Maintenance counter (Drive time of printer mechanism (minute) (integration))	Numerical value
GET_HFONT_24_CHECKSUM	301	1-byte font ID send (24 dots font, checksum)	Numerical value
GET_HFONT_24_ID	302	1-byte font ID send (24 dots font, ID)	Binary
GET_HFONT_24_INT_CHAR	303	1-byte font ID send (24 dots font, registered international character)	Binary
GET_HFONT_16_CHECKSUM	304	1-byte font ID send (16 dots font, checksum)	Numerical value
GET_HFONT_16_ID	305	1-byte font ID send (16 dots font, ID)	Binary
GET_HFONT_16_INT_CHAR	306	1-byte font ID send (16 dots font, registered international character)	Binary
GET_FFONT_LANG	351	2-byte font ID send (Language)	Binary
GET_FFONT_STANDARD	352	2-byte font ID send (Standard)	Binary
GET_FFONT_COMPANY	353	2-byte font ID send (Company name)	Binary
GET_FFONT_CHECKSUM	354	2-byte font ID send (Checksum)	Numerical value
GET_FFONT_DATA_SIZE	355	2-byte font ID send (Data size)	Numerical value
GET_FFONT_ROM_ID	356	2-byte font ID send (ROM ID)	Binary

6.3 Batch Process Control Method

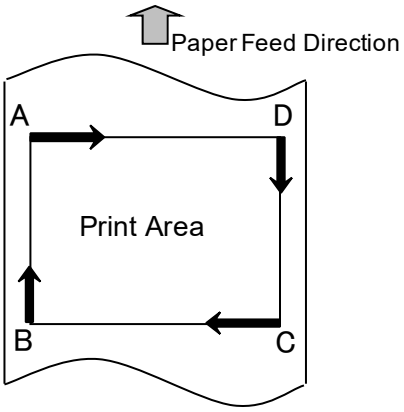
The following is the batch process control method available to specify.

Batch Process Control Method (TransactionFunction)		Description
Member	Value	
TRANSACTION_CLEAR	0	End batch process
TRANSACTION_START	1	Begin batch process
TRANSACTION_PRINT	2	End batch print and batch process

6.4 Print Direction

The followings are the print directions available to specify.

Print Direction (Direction)		Description	
Member	Value	Print Direction	Start Point
DIRECTION_LEFT_TO_RIGHT	0	Left to right	Top left (A in the figure below)
DIRECTION_BOTTOM_TO_TOP	1	Bottom to top	Bottom left (B in the figure below)
DIRECTION_RIGHT_TO_LEFT	2	Right to left	Bottom right (C in the figure below)
DIRECTION_TOP_TO_BOTTOM	3	Top to bottom	Top right (D in the figure below)



6.5 Print Position

The following is the print position available to specify.

Print Position (Alignment)		Description
Member	Value	
ALIGN_LEFT	0	Left aligned
ALIGN_CENTER	1	Centered
ALIGN_RIGHT	2	Right aligned

6.6 Character Set

The following is the character set available to specify.

Character Set (CharacterSet)		Description
Member	Value	
CODEPAGE_437	0	USA, Standard Europe (Code Page 437)
CODEPAGE_KATAKANA	1	Katakana
CODEPAGE_850	2	Multilingual (Code Page 850)
CODEPAGE_860	3	Portuguese (Code Page 860)
CODEPAGE_863	4	Canadian-French (Code Page 863)
CODEPAGE_865	5	Nordic (Code Page 865)
CODEPAGE_857	13	Turkish (Code Page 857)
CODEPAGE_737	14	Greek (Code Page 737)
CODEPAGE_1252	16	Latin (Code Page 1252)
CODEPAGE_866	17	Russian (Code Page 866)
CODEPAGE_852	18	Eastern Europe (Code Page 852)
CODEPAGE_858	19	Euro (Code Page 858)
CODEPAGE_855	34	Cyrillic (Code Page 855)
CODEPAGE_864	37	Arabic (Code Page 864)
CODEPAGE_1250	45	Central European (Code Page 1250)
CODEPAGE_1251	46	Cyrillic (Code Page 1251)
CODEPAGE_1253	47	Greek (Code Page 1253)
CODEPAGE_1254	48	Turkish (Code Page 1254)
CODEPAGE_USERPAGE	255	User Page

6.7 International Character Set

The following is the international character set available to specify.

International Character Set (InternationalCharacterSet)		Description
Member	Value	
INT_CHAR_SET_USA	0	USA
INT_CHAR_SET_FRANCE	1	France
INT_CHAR_SET_GERMANY	2	Germany
INT_CHAR_SET_UNITED_KINGDOM	3	United Kingdom
INT_CHAR_SET_DENMARK_1	4	Denmark I
INT_CHAR_SET_SWEDEN	5	Sweden
INT_CHAR_SET_ITALY	6	Italy
INT_CHAR_SET_SPAIN_1	7	Spain I
INT_CHAR_SET_JAPAN	8	Japan
INT_CHAR_SET_NORWAY	9	Norway
INT_CHAR_SET_DENMARK_2	10	Denmark II
INT_CHAR_SET_SPAIN_2	11	Spain II
INT_CHAR_SET_LATIN_AMERICA	12	Latin America

International Character Set (InternationalCharacterSet)		Description
Member	Value	
INT_CHAR_SET_ARABIA	17	Arabia

6.8 Character Font

The following is the character font available to specify.

Character Font (CharacterType)		Description
Member	Value	
CHAR_TYPE_FONT_CURRENT *1	0	Current setting
CHAR_TYPE_FONT_A	1	Font A (24×12), Kanji font A (24×24)
CHAR_TYPE_FONT_B	2	Font A (16×8), Kanji font B (16×16)

*1: When this value is set, the printer operates by the previous setting or default of the command without sending command.

6.9 Character Size

The following is the character size available to specify.

Character Size (CharacterScale)		Description
Member	Value	
CHAR_SCALE_CURRENT *1	0	Current setting
CHAR_SCALE_x1	1	× 1 (Standard)
CHAR_SCALE_x2	2	× 2 (double)
CHAR_SCALE_x3	3	× 3 (triple)
CHAR_SCALE_x4	4	× 4 (quadruple)
CHAR_SCALE_x5	5	× 5 (quintuple)
CHAR_SCALE_x6	6	× 6 (sextuple)
CHAR_SCALE_x7	7	× 7 (septuple)
CHAR_SCALE_x8	8	× 8 (octuple)

*1: When this value is set, the printer operates by the previous setting or default of the command without sending command.

6.10 Underline

The following is the underline available to specify.

Underline (Underline)		Description
Member	Value	
CHAR_UNDERLINE_CURRENT *1	0	Current setting
CHAR_UNDERLINE_NONE	1	No underline
CHAR_UNDERLINE_1DOT	2	Specify 1 dot width underline
CHAR_UNDERLINE_2DOT	3	Specify 2 dots width underline

*1: When this value is set, the printer operates by the previous setting or default of the command without sending command.

6.11 Bold Print

The following is the bold print available to specify.

Bold Print (Bold)		Description
Member	Value	
CHAR_BOLD_CURRENT *1	0	Current setting
CHAR_BOLD_OFF	1	No bold print
CHAR_BOLD_ON	2	Specify bold print

*1: When this value is set, the printer operates by the previous setting or default of the command without sending command.

6.12 Reverse Print

The following is the reverse print available to specify.

Reverse Print (Reverse)		Description
Member	Value	
CHAR_REVERSE_CURRENT *1	0	Current setting
CHAR_REVERSE_OFF	1	No reverse print
CHAR_REVERSE_ON	2	Specify reverse print

*1: When this value is set, the printer operates by the previous setting or default of the command without sending command.

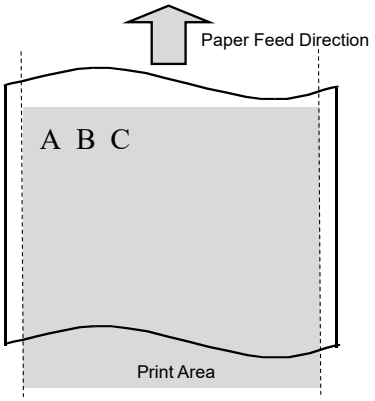
6.13 Character Rotation Print

The following is the character rotation print available to specify.

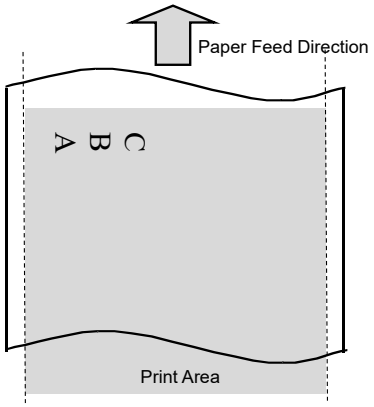
Character Rotation Print (Rotate)		Description
Member	Value	
ROTATE_CURRENT *1	0	Current setting
ROTATE_NONE	1	No rotation
ROTATE_90_TO_RIGHT	2	Rotate 90 degrees to right
ROTATE_180	3	Rotate 180 degrees
ROTATE_90_TO_LEFT	4	Rotate 90 degrees to left

*1: When this value is set, the printer operates by the previous setting or default command without sending command.

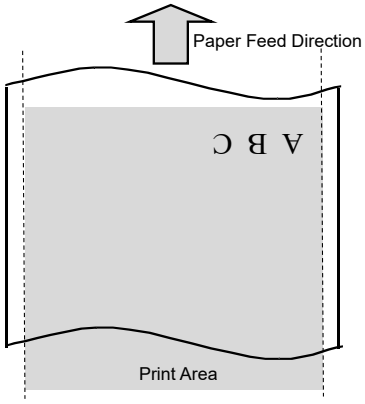
The following figures show the operation of **MpSetCharacterFormatting** or **SetCharacterFormatting**.



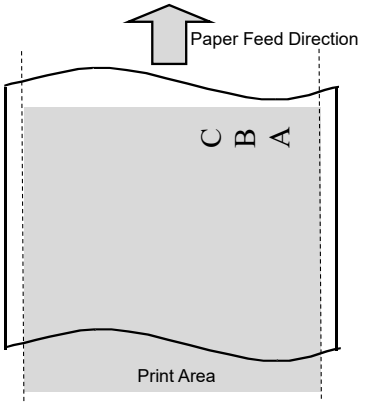
No rotation



Rotate 90 degrees to right



Rotate 180 degrees



Rotate 90 degrees to left

6.14 Barcode Type

The following is the barcode type available to specify.

Barcode Type (TypeBarcode)		Description	Category
Member	Value		
TYPE_BARCODE_UPC_A	0	UPC-A	Multilevel barcode
TYPE_BARCODE_UPC_E	1	UPC-E	Multilevel barcode
TYPE_BARCODE_JAN13	2	JAN13 (EAN13)	Multilevel barcode
TYPE_BARCODE_JAN8	3	JAN8 (EAN8)	Multilevel barcode
TYPE_BARCODE_CODE39	4	CODE39	Binary level barcode
TYPE_BARCODE_ITF	5	ITF	Binary level barcode
TYPE_BARCODE_CODABAR	6	CODABAR	Binary level barcode
TYPE_BARCODE_CODE128	7	CODE128	Multilevel barcode
TYPE_BARCODE_CODE93	8	CODE93	Multilevel barcode
TYPE_BARCODE_JAN13_ADDON2	9	JAN13 (EAN13) add-on 2	Multilevel barcode
TYPE_BARCODE_JAN13_ADDON5	10	JAN13 (EAN13) add-on 5	Multilevel barcode
TYPE_BARCODE_GS1_OMNI_DIRCTIONAL	11	GS1 Databar Omni-directional	Binary level barcode
TYPE_BARCODE_GS1_TRUNCATED	12	GS1 Databar Truncated	Binary level barcode
TYPE_BARCODE_GS1_LIMITED	13	GS1 Databar Limited	Binary level barcode
TYPE_BARCODE_GS1_EXPANDED	14	GS1 Databar Expanded	Binary level barcode

6.15 Barcode Module Width

The following is the barcode module width available to specify.

Barcode Module Width (ModuleWidthBarcode)		Description	
Member	Value	Multilevel Barcode Module Width (mm)	Binary Level Barcode Narrow Element (mm)
MODULE_WIDTH_BARCODE_2	2	0.250	0.250 (2 dots)
MODULE_WIDTH_BARCODE_3	3	0.375	0.375 (3 dots)
MODULE_WIDTH_BARCODE_4	4	0.500	0.500 (4 dots)
MODULE_WIDTH_BARCODE_5	5	0.625	0.625 (5 dots)
MODULE_WIDTH_BARCODE_6	6	0.750	0.750 (6 dots)

6.16 Barcode Height

The following is the barcode height available to specify.

Barcode Type	Default	Range of Barcode Height (dot)
UPC-A	0*1	1 to 255
UPC-E		
JAN13 (EAN13)		
JAN8 (EAN8)		
CODE39		
ITF		
CODABAR		
CODE128		
CODE93		
JAN13 (EAN13) add-on 2		
JAN13 (EAN13) add-on 5		

*1: When 0 is specified, the barcode height is set to 162 dots.

Regarding the following barcode types, the range of barcode height differs depending on the barcode module width.
When the [Height] value is specified less than the minimum value, the value is automatically changed to the minimum value.

Barcode Type / Module Width	Default	Range of Barcode Height (dot)
GS1 Databar Omni-directional		
MODULE_WIDTH_BARCODE_2	0※1	66 to 255
MODULE_WIDTH_BARCODE_3		99 to 255
MODULE_WIDTH_BARCODE_4		132 to 255
MODULE_WIDTH_BARCODE_5		165 to 255
MODULE_WIDTH_BARCODE_6		198 to 255
GS1 Databar Truncated		
MODULE_WIDTH_BARCODE_2	0※1	26 to 255
MODULE_WIDTH_BARCODE_3		39 to 255
MODULE_WIDTH_BARCODE_4		52 to 255
MODULE_WIDTH_BARCODE_5		65 to 255
MODULE_WIDTH_BARCODE_6		78 to 255
GS1 Databar Limited		
MODULE_WIDTH_BARCODE_2	0※1	20 to 255
MODULE_WIDTH_BARCODE_3		30 to 255
MODULE_WIDTH_BARCODE_4		40 to 255
MODULE_WIDTH_BARCODE_5		50 to 255
MODULE_WIDTH_BARCODE_6		60 to 255

Barcode Type / Module Width	Default	Range of Barcode Height (dot)
GS1 Databar Expanded		
MODULE_WIDTH_BARCODE_2	0※1	68 to 255
MODULE_WIDTH_BARCODE_3		102 to 255
MODULE_WIDTH_BARCODE_4		136 to 255
MODULE_WIDTH_BARCODE_5		170 to 255
MODULE_WIDTH_BARCODE_6		204 to 255

*1: When 0 is specified, the barcode height is set to 162 dots.

6.17 Barcode HRI Character

The following is the barcode HRI character available to specify.

HRI character (HriPositionBarcode)		Description
Member	Value	
HRI_BARCODE_NONE	0	No HRI character
HRI_BARCODE_TOP_FONT_A	1	Above barcode (font A)
HRI_BARCODE_BOTTOM_FONT_A	2	Below barcode (font A)
HRI_BARCODE_FONT_A	3	Above and below barcode (font A)
HRI_BARCODE_TOP_FONT_B	4	Above barcode (font B)
HRI_BARCODE_BOTTOM_FONT_B	5	Below barcode (font B)
HRI_BARCODE_FONT_B	6	Above and below barcode (font B)

6.18 Barcode N:W Ratio

The following is the barcode N:W ratio available to specify.

N:W Ratio (NwRatioBarcode)		Description
Member	Value	
NWRATIO_BARCODE_NONE *2	0	-
NWRATIO_BARCODE_1TO2 *1	0	1:2
NWRATIO_BARCODE_1TO2_5 *1	1	1:2.5
NWRATIO_BARCODE_1TO3 *1	2	1:3

*1: Specify when the following barcode types are used.

- CODE39
- ITF
- CODABAR

*2: Specify when using the barcode other than those stated in *1.

6.19 2-Dimensional Barcode Type

The following is the 2-dimensional barcode type available to specify.

2-Dimensional Barcode Type (Type2Dcode)		Description
Member	Value	
TYPE_2DCODE_QR_CODE	0	QR Code
TYPE_2DCODE_PDF417	1	PDF417
TYPE_2DCODE_DATA_MATRIX	2	Data Matrix
TYPE_2DCODE_MAXI_CODE	3	Maxi Code
TYPE_2DCODE_GS1_STACKED	4	GS1 Databar Stacked
TYPE_2DCODE_GS1_OMNI_DIRECTIONAL	5	GS1 Databar Stacked Omni-directional
TYPE_2DCODE_GS1_EXPANDED_STACKED	6	GS1 Databar Expanded Stacked

6.20 2-Dimensional Barcode Mode

The following is the 2-dimensional barcode mode available to specify.

2-Dimensional Barcode Mode (Mode2Dcode)*1		Description for 2-Dimensional Barcode Type			
		QR Code	PDF417	Data Matrix	Maxi Code
Member	Value	Model	Model	Module numbers H × W	Mode
MODE_2DCODE_NONE	0	-	-	-	Mode5
MODE_2DCODE_PDF417_STANDARD		-	Normal mode	-	-
MODE_2DCODE_DATA_MATRIX_AUTO		-	-	Automatic	-
MODE_2DCODE_QR_CODE_1	1	Model 1	-	-	-
MODE_2DCODE_PDF417_COMPACT		-	Simple mode	-	-
MODE_2DCODE_DATA_MATRIX_10_10		-	-	10 × 10	-
MODE_2DCODE_QR_CODE_2	2	Model 2	-	-	-
MODE_2DCODE_DATA_MATRIX_12_12		-	-	12 × 12	-
MODE_2DCODE_MAXI_CODE_2		-	-	-	Mode2
MODE_2DCODE_DATA_MATRIX_14_14	3	-	-	14 × 14	-
MODE_2DCODE_MAXI_CODE_3		-	-	-	Mode3
MODE_2DCODE_DATA_MATRIX_16_16	4	-	-	16 × 16	-
MODE_2DCODE_MAXI_CODE_4		-	-	-	Mode4
MODE_2DCODE_DATA_MATRIX_18_18	5	-	-	18 × 18	-
MODE_2DCODE_MAXI_CODE_5		-	-	-	Mode5
MODE_2DCODE_DATA_MATRIX_20_20	6	-	-	20 × 20	-
MODE_2DCODE_DATA_MATRIX_22_22	7	-	-	22 × 22	-
MODE_2DCODE_DATA_MATRIX_24_24	8	-	-	24 × 24	-
MODE_2DCODE_DATA_MATRIX_26_26	9	-	-	26 × 26	-
MODE_2DCODE_DATA_MATRIX_32_32	10	-	-	32 × 32	-

2-Dimensional Barcode Mode (Mode2Dcode)*1		Description for 2-Dimensional Barcode Type			
		QR Code	PDF417	Data Matrix	Maxi Code
Member	Value	Model	Model	Module numbers H × W	Mode
MODE_2DCODE_DATA_MATRIX_36_36	11	-	-	36 × 36	-
MODE_2DCODE_DATA_MATRIX_40_40	12	-	-	40 × 40	-
MODE_2DCODE_DATA_MATRIX_44_44	13	-	-	44 × 44	-
MODE_2DCODE_DATA_MATRIX_48_48	14	-	-	48 × 48	-
MODE_2DCODE_DATA_MATRIX_52_52	15	-	-	52 × 52	-
MODE_2DCODE_DATA_MATRIX_64_64	16	-	-	64 × 64	-
MODE_2DCODE_DATA_MATRIX_72_72	17	-	-	72 × 72	-
MODE_2DCODE_DATA_MATRIX_80_80	18	-	-	80 × 80	-
MODE_2DCODE_DATA_MATRIX_88_88	19	-	-	88 × 88	-
MODE_2DCODE_DATA_MATRIX_96_96	20	-	-	96 × 96	-
MODE_2DCODE_ DATA_MATRIX_104_104	21	-	-	104 × 104	-
MODE_2DCODE_ DATA_MATRIX_120_120	22	-	-	120 × 120	-
MODE_2DCODE_ DATA_MATRIX_132_132	23	-	-	132 × 132	-
MODE_2DCODE_ DATA_MATRIX_144_144	24	-	-	144 × 144	-
MODE_2DCODE_DATA_MATRIX_8_18	25	-	-	8 × 18	-
MODE_2DCODE_DATA_MATRIX_8_32	26	-	-	8 × 32	-
MODE_2DCODE_DATA_MATRIX_12_26	27	-	-	12 × 26	-
MODE_2DCODE_DATA_MATRIX_12_36	28	-	-	12 × 36	-
MODE_2DCODE_DATA_MATRIX_16_36	29	-	-	16 × 36	-
MODE_2DCODE_DATA_MATRIX_16_48	30	-	-	16 × 48	-

*1: When using the 2-dimensional barcode type that is not listed in the table, specify 0 (MODE_2DCODE_NONE).

6.21 2-Dimensional Barcode Module Size

The following is the 2-dimensional barcode module size available to specify.

2-Dimensional Barcode Module Size (ModuleSize2Dcode)*1		Description for 2-Dimensional Barcode Type			
		QR Code	PDF417	Data Matrix	All GS1 Databar
Member	Value	Module Size	Nominal Narrow Element	Module Size	Module Size
MODULE_SIZE_2DCODE_NONE	0	-	-	-	-
MODULE_SIZE_2DCODE_DEFAULT		Default (6 dots)	Default (4 dots)	Default (6 dots)	Default (6 dots)
MODULE_SIZE_2DCODE_2	2	2 dots	2 dots	2 dots	2 dots
MODULE_SIZE_2DCODE_3	3	3 dots	3 dots	3 dots	3 dots
MODULE_SIZE_2DCODE_4	4	4 dots	4 dots	4 dots	4 dots
MODULE_SIZE_2DCODE_5	5	5 dots	5 dots	5 dots	5 dots
MODULE_SIZE_2DCODE_6	6	6 dots	6 dots	6 dots	6 dots
MODULE_SIZE_2DCODE_7	7	7 dots	7 dots	7 dots	7 dots
MODULE_SIZE_2DCODE_8	8	8 dots	8 dots	8 dots	8 dots
MODULE_SIZE_2DCODE_9	9	9 dots	-	9 dots	9 dots
MODULE_SIZE_2DCODE_10	10	10 dots	-	10 dots	10 dots
MODULE_SIZE_2DCODE_11	11	11 dots	-	11 dots	11 dots
MODULE_SIZE_2DCODE_12	12	12 dots	-	12 dots	12 dots
MODULE_SIZE_2DCODE_13	13	13 dots	-	13 dots	13 dots
MODULE_SIZE_2DCODE_14	14	14 dots	-	14 dots	14 dots
MODULE_SIZE_2DCODE_15	15	15 dots	-	15 dots	15 dots
MODULE_SIZE_2DCODE_16	16	16 dots	-	16 dots	16 dots

*1: When using the 2-dimensional barcode type that is not listed in the table, specify 0 (MODULE_SIZE_2DCODE _NONE).

6.22 2-Dimensional Barcode Module Height

The following is the 2-dimensional barcode module height available to specify.

2-Dimensional Barcode Type*1	Description for 2-Dimensional Barcode Type		
	PDF module height		Height of 1 module
	Default	Range (dot)	Range (dot)
PDF417	0*2	2 to 127	-
GS1 Databar Stacked Omni-directional	-		33 to 255

*1: When using the 2-dimensional barcode type that is not listed in the table, specify 0.
*2: When 0 is specified, the PDF module height is set to 162 dots (default value).

6.23 Number of 2-Dimensional Barcode Columns

The following is the number of 2-dimensional barcode columns available to specify.

2-Dimensional Barcode Type[*1]	Description for 2-Dimensional Barcode Type	
	Number of columns in data area	Number of elements in 1 line [*2]
PDF417	0[*3], 1 to 30	-
GS1 Databar Expanded Stacked	-	2 to 20

*1: When using the 2-dimensional barcode type that is not listed in the table, specify 0.
*2: Specify an even number for the number of elements in 1 line.
*3: When 0 is specified, the number of columns is set automatically.

6.24 Number of 2-Dimensional Barcode Rows

The following is the number of 2-dimensional barcode rows available to specify.

2-Dimensional Barcode Type[*1]	Description for 2-Dimensional Barcode Type
	Number of Rows
PDF417	0[*2], 3 to 90

*1: When using the 2-dimensional barcode type that is not listed in the table, specify 0.
*2: When 0 is specified, the number of rows is set automatically.

6.25 2-Dimensional Barcode Error Correction Level

The following is the 2-dimensional barcode error correction level available to specify.

2-Dimensional barcode error correction level (ErrorCorrect2Dcode)*1		Description for 2-Dimensional Barcode Type	
		QR Code	PDF417
Member	Value	Error Correction Level	Error Correction Level
ERR_CORRECTION_2DCODE_NONE	0	-	-
ERR_CORRECTION_2DCODE_ QR_CODE_L		L	-
ERR_CORRECTION_2DCODE_ PDF417_0		-	0
ERR_CORRECTION_2DCODE_ QR_CODE_M	1	M	-
ERR_CORRECTION_2DCODE_ PDF417_1		-	1
ERR_CORRECTION_2DCODE_ QR_CODE_Q	2	Q	-
ERR_CORRECTION_2DCODE_ PDF417_2		-	2
ERR_CORRECTION_2DCODE_ QR_CODE_H	3	H	-
ERR_CORRECTION_2DCODE_ PDF417_3		-	3
ERR_CORRECTION_2DCODE_ PDF417_4	4	-	4
ERR_CORRECTION_2DCODE_ PDF417_5	5	-	5
ERR_CORRECTION_2DCODE_ PDF417_6	6	-	6
ERR_CORRECTION_2DCODE_ PDF417_7	7	-	7
ERR_CORRECTION_2DCODE_ PDF417_8	8	-	8

*1: When using the 2-dimensional barcode type that is not listed in the table, specify 0 (ERR_CORRECTION_2DCODE_NONE).