



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

U00138400704

Seiko Instruments Inc.

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

U00138400700	August 2016
U00138400701	November 2016
U00138400702	August 2018
U00138400703	February 2019
U00138400704	December 2019


Copyright © 2016-2019 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 MP-A40 Series SDK (hereinafter referred to as "the SDK") for Windows Embedded CE (hereinafter referred to as "WinCE") provided by Seiko Instruments Inc. (hereinafter referred to as "SII").

Target Printers

The printers supported by the SDK are as follows.

Printer	Communication Interface
MP-A40 (Bluetooth Model)	Bluetooth
	USB
MP-A40 (Wireless LAN Model)	Wireless LAN
	USB

Table of Contents

Chapter 1	Product Overview	1-1
1.1	Function	1-1
1.2	Configuration	1-1
1.2.1	SII Print Class Library	1-2
1.2.2	Sample Program	1-2
Chapter 2	Product Specifications	2-1
2.1	Product Specifications	2-1
2.1.1	Operating Environment	2-1
2.1.2	Operating Conditions	2-2
2.1.3	Precaution	2-2
Chapter 3	How to Use the Library	3-1
3.1	Provided Files	3-1
3.2	Build the Library into Microsoft Visual Studio Projects	3-2
3.3	Execution Condition on WinCE Device	3-2
3.4	Others	3-2
Chapter 4	Function of the Library	4-1
4.1	Overview of the Library	4-1
4.2	Structure of the Library	4-1
4.3	Data Send/Receive Processing of the Library and the Limitations	4-1
4.4	API Reference	4-2
4.4.1	Interface	4-4
4.4.2	Class	4-6
4.4.3	Enumerated Type (Enum)	4-44
4.4.4	Exception	4-58
Chapter 5	Sample Program	5-1
5.1	Sample Program Overview	5-1
5.2	How to Use Sample Program	5-1
5.3	Outline of Sample Program UI display	5-2
5.4	Precaution	5-2
Chapter 6	Disclaimer	6-1

A.1 Character Code Table A-1

A.2 International Character Set..... A-11

Chapter 1

Product Overview

This chapter describes the product overview of the SDK.

1.1 Function

The SII print class library included in the SDK provides WinCE applications in Microsoft .NET Compact Framework (hereinafter referred to as ".NET Compact Framework") environment with the functions to use SII printer MP-A40 series (hereinafter referred to as "printer"). Moreover, the SDK provides Microsoft Visual Studio projects as a sample program for SII print class library.

1.2 Configuration

The SII print class library and the sample program included in the SDK are located in the section surrounded by dashed lines in the WinCE OS configuration diagram (Figure 1-1).

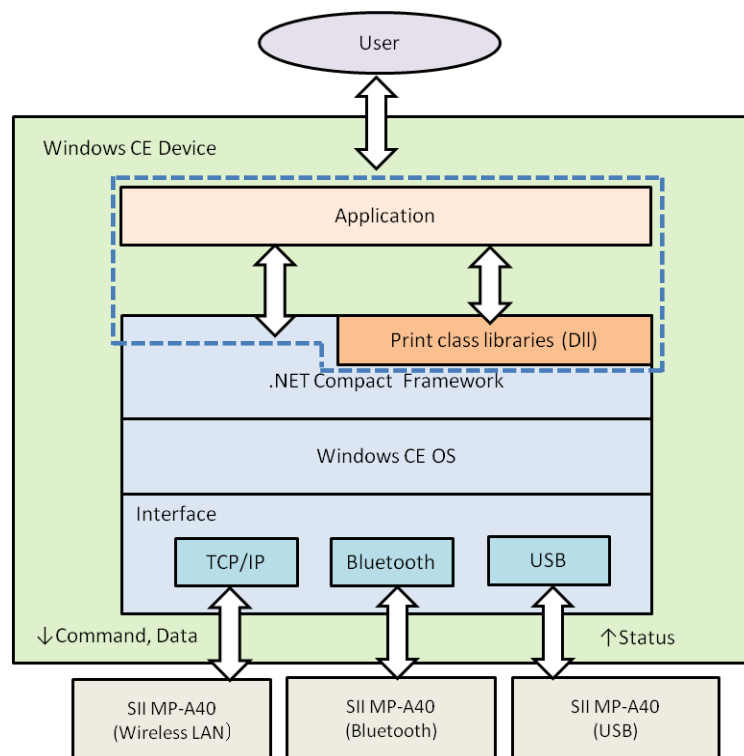


Figure 1-1

1.2.1 SII Print Class Library

By using SII Print Class Library (hereinafter referred to as "the library"), WinCE applications can easily send print data and printer commands to printer through communication port (Bluetooth, USB, or TCP/IP) on an WinCE device. Also, the applications can retrieve printer status.

The library provides the following functions.

- Connection to / disconnection from a printer
- Sending data to a printer (print data and/or printer commands)
- Barcode print and 2-dimensional barcode print
- Sending a data file to a printer (print data and/or printer commands)
- Retrieving printer status
- Retrieving various responses from a printer
- Printer search by Bluetooth or TCP/IP

1.2.2 Sample Program

SII provides this program as an WinCE application sample using the library.

Chapter 2

Product Specifications

This chapter describes the product specifications of the library.

2.1 Product Specifications

2.1.1 Operating Environment

Operating environment for the library is shown in the following table.

Item	Specifications
Target OS	Windows CE 5.0 Windows Embedded CE 6.0 Windows Embedded Compact 7 (including Windows Mobile 6.1, Windows Mobile 6.5, and Windows Embedded Handheld 6.5)
Target .NET Framework	.NET Compact Framework 3.5 ^{*1}
Supported Language	Japanese English
Communication Port	Bluetooth ^{*2} USB ^{*3} TCP/IP ^{*4}
Supported Development Language	Able to use .NET Compact Framework Microsoft C# Microsoft Visual Basic

(NOTE) *1: It is necessary to install before using this software.

*2: Bluetooth connection needs to be established by SPP (Serial Port Profile).
For the driver related to Bluetooth, use the Microsoft system standard driver.

*3: For using USB, Windows CE 5.x based OS is only supported.
And WinCE device needs to support USB host function and to be implemented
USB Printer class driver (USBPRN.dll) provided by Platform Builder on the target OS.
USB Printer class driver can be handled as an LPT port on the target OS of the library.

*4: WinCE device and printer need to be connected to the same network.
When the printer is in client mode, connect through the access point.
When the printer is in simple AP mode, connect directly.

2.1.2 Operating Conditions

The operating conditions for the library are shown in the following table.

Set the Function Settings of the printer to the values shown in the following table before using the library.
See "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCES" for details about Function Settings.

MS	Function	Value	Setting
1-1	Interface Selection (Interface)	0 / 1	0: USB 1: USB/Wireless ^{*1}
1-2	Mark Mode Selection (Mark Mode)	0 / 1	0: Enable ^{*2} 1: Disable
1-3	Command System Selection (Command System)	0	ESC/POS
1-4		0	
1-5		0	
2-2	Realtime Command Selection (Realtime Command)	0	Enable
9-1	Automatic Status Response Selection <ESC/POS> (Auto Status Back<ESC/POS>)	0	Enable
9-2	Initialized Response Selection <ESC/POS> (Init. Response<ESC/POS>)	0	Enable

(NOTE) *1: Select "USB/Wireless" when using Bluetooth or Wireless LAN interface.

*2: Select "Enable" when feeding to home position of the marked thermal paper.

2.1.3 Precaution

When using TCP/IP connection, this library occupies the printer port. Therefore, the communication port cannot be shared with the printer driver or other libraries.

Chapter 3

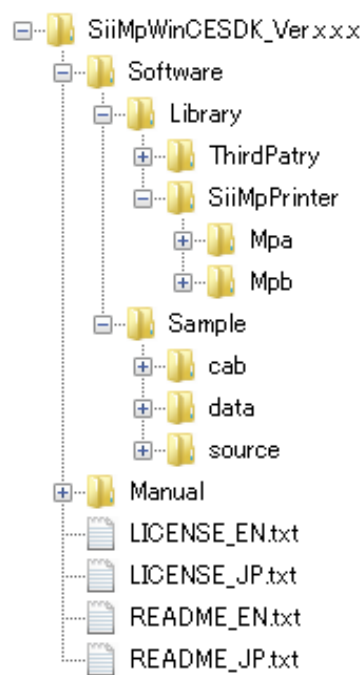
How to Use the Library

This chapter describes how to use the library.

3.1 Provided Files

The SDK provides DLL file for the library.

The folder configuration of the SDK is shown in the following.



3.2 Build the Library into Microsoft Visual Studio Projects

This section describes how to build the library into Microsoft Visual Studio projects. The explanation is based on Microsoft Visual Studio C# 2008.

- (a) Create any project in Microsoft Visual Studio.
- (b) Select [Project] – [Add Reference] in Microsoft Visual Studio.
- (c) Add the DLL file located in [CD-ROM drive]\WindowsCE\Library\SiiMpPrinter\Mpx^{*1} to [Browse] of the [Add Reference] dialog.

*1: The alphabet (a or b) indicating series is displayed for x.

3.3 Execution Condition on WinCE Device

When executing the WinCE application created by Microsoft Visual Studio on WinCE device, put the following files in the same folder and execute it.

- WinCE application (xxxxx.exe)
- DLL file under the \WindowsCE\Library\SiiMpPrinter\Mpx folder
- DLL file under the \WindowsCE\Library\ThirdParty folder

3.4 Others

The following SDKs may be needed depending on the type of program to develop. Obtain your needed program from Microsoft website.

- Windows Mobile 5 Pocket PC SDK
- Windows Mobile 6 Standard SDK
- Windows Mobile 6 Professional SDK

Chapter 4

Function of the Library

This chapter describes the APIs for each class implemented in the library.

4.1 Overview of the Library

The library provides the functions for using printers for WinCE applications.

4.2 Structure of the Library

The file format of the library is DLL.

In order to use the library in WinCE applications, build the library into Microsoft Visual Studio projects.

See "Chapter 3 How to Use the Library" for details about how to build the library into Microsoft Visual Studio projects.

4.3 Data Send/Receive Processing of the Library and the Limitations

This library secures 10 MB (10485760 bytes) of memory for the send buffer in the SDK before starting use of the printer by **Open**. Since securing memory depends on the system, an error occurs if memory cannot be secured.

This library buffers the receive data from the printer into the receive buffer in the SDK. Since the printer returns the auto status response every time the status changes, the receive data is buffered into the receive buffer sequentially. The maximum receive data to be buffered is 4096 bytes. When the printer is reconnected, the receive buffer in the SDK may collectively receive the data that was buffered in the printer.

The receive data buffered in the receive buffer can be retrieved by **Read**. The receive data retrieved by **Read** is deleted from the buffer.

When the buffered receive data exceeds 4096 bytes, the old data in buffer is discarded sequentially; therefore, execute **Read** so that the receive data size does not exceed 4096 bytes. The receive data size can be retrieved by **GetReadSize**.

For the details of each method, see "4.4 API Reference".

For the details of the various responses, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

4.4 API Reference

The name space of the library is

SII.SPS.Windows.NetCompactFramework.PrintClassLibrary.MobilePrinter.

SII.SPS.Windows.NetCompactFramework.PrintClassLibrary.MobilePrinter provides the following functions.

• Interface

Class Name	Description
StatusCallbackHandler	Interface that implements the printer status change.
DiscoveryHandler	Interface that retrieves the complete event of printer searching.

• Class

Class Name	Description
PrinterManager	Class that provides the API used for communication with the printer and for printing.
PrinterStatus	Class that stores the printer status.
PrinterDiscovery	Class that searches the printer.
DiscoveredPrinter	Class that stores the printer information found by printer searching.

• Enumerated type (Enum)

Class Name	Description
PrinterInterface	Enumerator used for specifying the communication interface of Open .
PrinterModel	Enumerator used for specifying the printer model of Open .
Dithering	Enumerator used for dithering of ResisterLogo and SendDataFile .
PrinterInformation	Enumerator used for specifying the printer information of GetPrinterInformation , GetPrinterInformationNumber and GetPrinterInformationString .
TransactionFunction	Enumerator used for specifying the batch process control method of ControlTransaction .
Direction	Enumerator used for specifying the print direction of SelectPageMode .
Alignment	Enumerator used for specifying the print position of SetStandardModeAlignment .
CharacterSet	Enumerator used for specifying the character set of SelectCharacterSet and GetCharacterSet .
InternationalCharacterSet	Enumerator used for specifying the international character set of SelectInternationalCharacterSet and GetInternationalCharacter .
CharacterType	Enumerator used for specifying the character font of SetCharacterFormatting .
CharacterScale	Enumerator used for specifying the character scale of SetCharacterFormatting .

Class Name	Description
Underline	Enumerator used for specifying the underline of SetCharacterFormatting .
Bold	Enumerator used for specifying the bold print of SetCharacterFormatting .
Reverse	Enumerator used for specifying the reverse print of SetCharacterFormatting .
Rotate	Enumerator used for specifying the character rotation print of SetCharacterFormatting , and for specifying print barcode direction of SetStandardModeBarcodeDirection .
TypeBarcode	Enumerator used for specifying the barcode type of PrintBarcode .
ModuleWidthBarcode	Enumerator used for specifying the barcode module width or narrow element of PrintBarcode .
HriPositionBarcode	Enumerator used for specifying the barcode HRI characters of PrintBarcode .
NwRatioBarcode	Enumerator used for specifying the barcode N:W ratio of PrintBarcode .
Type2Dcode	Enumerator used for specifying the 2-dimensional barcode type of Print2Dcode .
Mode2Dcode	Enumerator used for specifying the 2-dimensional barcode mode of Print2Dcode .
ModuleSize2Dcode	Enumerator used for specifying the 2-dimensional barcode module size of Print2Dcode .
ErrorCorrect2Dcode	Enumerator used for specifying the 2-dimensional barcode error correction level of Print2Dcode .
LogFileSize	Enumerator used for specifying the maximum size of the log file of SetLog .
ErrorCode	Enumerator of the error code that can be retrieved by PrinterException .

- Exception

Class Name	Description
PrinterException	Exception class that is thrown at API call.

4.4.1 Interface

(1) StatusCallbackHandler

- Summary

This interface is for retrieving the event of printer status change.
This interface provides the following function.

Public Methods

Method	Function Summary
StatusChanged	Register process on printer status change.

- Public Methods

StatusChanged	Register process on printer status change
---------------	---

Syntax public void **StatusChanged** (PrinterStatus *status*)

Parameter *status* Instance of **PrinterStatus**

Description This method is called when a callback of the printer status change is started by **StartCallbackFunction** and the printer status is changed.

This is a method of interface so it is not implemented. Implement the optional process in the class that receives a callback of the printer status change.

In *status*, assign the instance of **PrinterStatus** class including the printer status at the time of change. Printer status can be retrieved from *status* by **PrinterStatus** class method. For the **PrinterStatus** method, see "4.4.2(2) **PrinterStatus**".

(2) **DiscoveryHandler**

• **Summary**

This interface retrieves the event when searching a printer.
This interface provides the following function.

Public Methods

Method	Function Summary
DiscoveryFinished	Finalization of printer search
DiscoveryCancelled	Cancellation of printer search

• **Public Methods**

DiscoveryFinished Finalization of printer search

Syntax public void **DiscoveryFinished** ()

Description This method is called when the search by **StartDiscoveryPrinter** is finished.

This is a method of interface so it is not implemented. Implement the optional process in the class that receives notification of finishing of printer search.

DiscoveryCancelled Cancellation of printer search

Syntax public void **DiscoveryCancelled** ()

Description This method is called when the search is canceled by **StartDiscoveryPrinter**.

This is a method of interface so it is not implemented. Implement the optional process in the class that receives notification of canceling of printer search.

4.4.2 Class

(1) PrinterManager

• Summary

This class provides the API used for communication with the printer and for printing.
This class provides the following functions.

Public Methods

Method	Function Summary	Standard Mode ^{*1}	Page Mode ^{*1}
PrinterManager	Constructor	✓	✓
Open	Start using printer	✓	✓
Close	Finish using printer	✓	✓
SetWriteTimeout	Set send timeout period	✓	✓
GetWriteTimeout	Retrieve send timeout period	✓	✓
SetResponseTimeout	Set receive timeout period	✓	✓
GetResponseTimeout	Retrieve receive timeout period	✓	✓
Write	Send binary data	✓	✓
Read	Retrieve receive data	✓	✓
GetReadSize	Retrieve available receive data size	✓	✓
WriteAndWaitResponse	Send and receive binary data	✓	✓
Reset	Reset printer	✓	✓
GetStatus	Retrieve printer status	✓	✓
StartCallbackFunction	Start callback of printer status change	✓	✓
StopCallbackFunction	Finish callback of printer status change	✓	✓
RegisterStyleSheet	Register style sheet to printer	✓	✓
DeleteStyleSheet	Delete style sheet in printer	✓	✓
RegisterLogo	Register logo to printer	✓	✓
DeleteLogo	Delete logo in printer	✓	✓
GetPrinterInformation	Retrieve printer information	✓	✓
GetPrinterInformationNumber		✓	✓
GetPrinterInformationString		✓	✓
ControlTransaction	Start/finish print data batch sending	✓	✓
SelectStandardMode	Start standard mode	-	✓
SelectPageMode	Start page mode	✓	-
PrintPageModeData	Print page mode data	-	✓ ^{*2}
SetPageModeVerticalPosition	Specify vertical absolute position in page mode	-	✓ ^{*2}
SetStandardModeArea	Specify print area in standard mode	✓ ^{*3}	-
SetStandardModeAlignment	Alignment	✓ ^{*3}	-

Method	Function Summary	Standard Mode ^{*1}	Page Mode ^{*1}
SetHorizontalPosition	Specify absolute position	✓	✓
SetStandardModeBarcodeDirection	Select print direction for barcode or 2-dimensional barcode	✓ ^{*3}	-
SetLineSpacing	Specify line space amount	✓ ^{*4}	✓ ^{*4}
SetCharacterRightSpace	Specify character right space amount	✓ ^{*4}	✓ ^{*4}
SelectCharacterSet	Select character set	✓	✓
GetCharacterSet	Retrieve specified character set	✓	✓
SelectInternationalCharacterSet	Select international character set	✓	✓
GetInternationalCharacter	Retrieve specified international character set	✓	✓
SetCharacterFormatting	Format character	✓	✓
PrintText	Send text data	✓	✓
PrintLogo	Print logo registered in printer	✓	✓
SendDataFile	Send file data	✓	✓
PrintBarcode	Print barcode	✓	✓
Print2Dcode	Print 2-dimensional barcode	✓	✓
PrintPageModeRectangle	Draw rectangle in page mode	-	✓ ^{*2}
FeedLine	Feed paper by line	✓	✓
FeedDotLine	Feed paper by dot	✓	✓
FeedCutPosition	Feed paper to cut position	✓	✓
FeedMarkPosition	Marked paper form feed	✓	✓
SetLog	Specify log output	✓	✓

*1 ✓: Enabled, -: Disabled

*2 When use this method, execute **SelectPageMode** beforehand to start page mode.

*3 When use this method, execute **SelectStandardMode** beforehand to start standard mode.

*4 Independent settings are available for standard mode and page mode respectively.

Public Properties

Method	Function Summary	Standard Mode ^{*1}	Page Mode ^{*1}
IsOpened	Retrieve printer using status	✓	✓

*1 ✓: Enabled, -: Disabled

• Public Methods

PrinterManager

Constructor

Constructor for **com.seikoinstruments.sdk.mobileprinter.PrinterManager** class.

Syntax **public PrinterManager()**

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "4.4.4(1) **PrinterException**"

Open

Start using printer

Starts using a printer.

Syntax **public void Open(PrinterInterface *prnIf*,**
 PrinterModel *prnModel*,
 String *address*,
 Int32 *socketKeepingTime*)

Parameter	<i>prnIf</i>	Communication interface constant See "4.4.3(1) PrinterInterface " for available settings.
	<i>prnModel</i>	Printer model constant See "4.4.3(2) PrinterModel " for available settings.
	<i>address</i>	Varies depending on <i>prnIf</i> setting. <ul style="list-style-type: none"> • When specify PRN_IF_TCP: In <i>address</i>, specify IP address of the printer to connect. Example: "192.168.0.190" • When specify PRN_IF_BT: In <i>address</i>, specify Bluetooth address of the printer to connect. Example: "00:11:22:AA:BB:CC" • When specify PRN_IF_USB: In <i>address</i>, specify in the range from LPT1: to LPT9:.
	<i>socketKeepingTime</i>	Socket keeping time (second) Varies depending on <i>prnIf</i> setting. <ul style="list-style-type: none"> • When specify PRN_IF_TCP: The valid range is 60 to 300. When the specified value is below 60, the value is set to 60, and when the specified value exceeds 300, the value is set to 300. In <i>socketKeepingTime</i>, specify the same value as the receive timeout period setting retrieved by the printer command "Send Wireless LAN Communication Setting". The initial value of the receive timeout period setting is 300. For details of the printer command, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE". • When specify PRN_IF_BT: The specified value is ignored. • When specify PRN_IF_USB: The specified value is ignored.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "4.4.4(1) **PrinterException**"

Description	<p>This method connects to a printer specified by <i>prnModel</i> through a communication interface specified by <i>prnIf</i>.</p> <p>Monitoring the printer status is started by this method. The latest printer status is retrieved by GetStatus. Changes in the printer status can be notified as events by StatusChanged, StartCallbackFunction, and StopCallbackFunction.</p>
Note	<p>Do not disable the automatic status response by printer command "Enable/Disable Automatic Status Back" or function settings. In that case, the printer status cannot be monitored, and the related function cannot be operated. For the automatic status response and the function settings, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".</p> <p>When data is sent to a printer through TCP/IP connection and the socket keeping time is passed, communication socket in the library is discarded. After that, a communication socket is created again and connection starts. Therefore, if the printer is requested to connect from the other host on the same network at the timing of the communication socket discarding, the printer establishes the communication with the other host and the reconnection may be failed.</p>

Close	Finish using printer
--------------	-----------------------------

Finishes using the printer and monitoring the printer status.

Syntax	public void Close ()
Exception	<p>PrinterException</p> <p>PrinterException is thrown when an error occurs while calling the method. For the details, see "4.4.4(1) PrinterException"</p>
Description	Send data retained by ControlTransaction is discarded.

SetWriteTimeout	Set send timeout period
------------------------	--------------------------------

Specifies the send timeout period.

Syntax	public void SetWriteTimeout (Int32 <i>writeTimeout</i>)		
Parameter	<table border="0"> <tr> <td><i>writeTimeout</i></td><td> <p>Send timeout period (millisecond)</p> <p>The valid range is 1000 to 90000.</p> <p>When the specified value is below 1000, the value is set to 1000.</p> <p>When the specified value exceeds 90000, the value is set to 90000.</p> <p>When the value is not set in this method, the send timeout period is 10000 seconds (initial value).</p> </td></tr> </table>	<i>writeTimeout</i>	<p>Send timeout period (millisecond)</p> <p>The valid range is 1000 to 90000.</p> <p>When the specified value is below 1000, the value is set to 1000.</p> <p>When the specified value exceeds 90000, the value is set to 90000.</p> <p>When the value is not set in this method, the send timeout period is 10000 seconds (initial value).</p>
<i>writeTimeout</i>	<p>Send timeout period (millisecond)</p> <p>The valid range is 1000 to 90000.</p> <p>When the specified value is below 1000, the value is set to 1000.</p> <p>When the specified value exceeds 90000, the value is set to 90000.</p> <p>When the value is not set in this method, the send timeout period is 10000 seconds (initial value).</p>		
Description	<p>The specified value can be retrieved by GetWriteTimeout.</p> <p>This method is enabled in the following methods.</p> <ul style="list-style-type: none"> • Write • WriteAndWaitResponse (Transmission process part) • RegisterLogo • RegisterStyleSheet • ControlTransaction (when TRANSACTION_PRINT is selected in <i>control</i>) • PrintText • SendDataFile • PrintBarcode • Print2Dcode 		

GetWriteTimeout

Retrieve send timeout period

Retrieves the send timeout period.

Syntax public Int32 **GetWriteTimeout**()

Return value Send timeout period (millisecond)

SetResponseTimeout

Set receive timeout period

Specifies the receive timeout period.

Syntax public void **SetResponseTimeout**(Int32 *respTimeout*)

Parameter *respTimeout* Receive timeout period (millisecond)
The valid range is 1000 to 90000.
When the specified value is below 1000, the value is set to 1000.
When the specified value exceeds 90000, the value is set to 90000.
When the value is not set in this method, the send timeout period is 10000 seconds (initial value).

Description The specified value is retrieved by **GetResponseTimeout**.

This method is enabled in the following methods.

- **WriteAndWaitResponse**
- **GetPrinterInformation**
- **GetPrinterInformationNumber**
- **GetPrinterInformationString**

GetResponseTimeout

Retrieve receive timeout period

Retrieves the receive timeout period.

Syntax public Int32 **GetResponseTimeout**()

Return value Receive timeout period (millisecond)

Write

Send Binary data

Sends the binary data.

Syntax public void **Write**(Byte[] *binary*, Int32 *offset*)

Parameter *binary* Binary data to send to a printer
A maximum of 16 KB (16384 bytes) of data size can be specified.

offset Specification of the starting position of the data to send

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "4.4.4(1) **PrinterException**".

Description When **ControlTransaction** is not used, the timeout period specified in **SetWriteTimeout** is valid.

Do not include a printer command that initializes the printer other than the printer command "Initialize Printer" in the data to send.
For printer initialization, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".
When performing a hardware reset, execute **Reset**.

This method is aborted by **Reset**.

Read	Retrieve receive data
------	-----------------------

Retrieves the receive data buffered in the buffer.

Syntax `public Byte[] Read(Int32 bufferSize)`

Parameter *bufferSize* Receive data size (byte)
The valid range is 1 to 4096.
When the specified value exceeds 4096, the value is set to 4096.
When the specified value is below 0, sends the error notice.

Return value Receive data

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "4.4.4(1) **PrinterException**".

Description When entire receive data buffered in the buffer needs to be retrieved, specify the value in *bufferSize* retrieved by **GetReadSize**.

When this method is reexecuted after retrieving receive data by this method, retrieving starts from the following receive data in the buffer.

For the procedure and the limitations, see "4.3 Data Send/Receive Processing of the Library and the Limitations".

GetReadSize	Retrieve available receive data size
-------------	--------------------------------------

Retrieves the available receive data size.

Syntax `public Int32 GetReadSize()`

Return value Available receive data size (byte)

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "4.4.4(1) **PrinterException**".

Sends the binary data and retrieves the receive data from the time when this method is executed.

```
Syntax      public Byte[] WriteAndWaitResponse(Byte[] sendBinary,  
            Int32 bufferSize,  
            Boolean respControl,  
            Boolean incAsbData)
```

Parameter	Description
<i>sendBinary</i>	Data to send A maximum data size is 16 KB (16384 bytes) can be specified.
<i>bufferSize</i>	Maximum size of the data to send The valid range is 1 to 10485760. When the specified value exceeds 10485760, the value is set to 10485760. When the specified value is below 0, sends the error notice.
<i>respControl</i>	Operation selection for receive process True : Receive some data or continue to receive data until timeout period is over False : Receive the specified sized data in <i>bufferSize</i> or continue to receive data until timeout period is over
<i>incAsbData</i>	Include the automatic status response to the receive data or not True : Include False : Not include

Return value Receive data

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
 For the details, see "4.4.4(1) **PrinterException**".

Description	This method is appropriate to send printer commands that respond with character string or capacity and other values, and to retrieve the response.
-------------	--

Next method execution after this method execution is not processed until completion of this method receiving. For retrieving the execution response or progress response, use **Read** but not this method. For the details of execution response or progress response, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

For sending, the timeout period specified in **SetWriteTimeout** is valid.

For receiving, the timeout period specified in **SetResponseTimeout** is valid.

Do not include a printer command that initializes the printer other than the printer command "Initialize Printer" in the data to send.

For printer initialization, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

When performing a hardware reset, execute **Reset**.

This method is aborted by **Reset**.

Resets the printer.

Syntax public void **Reset()**

Exception **PrinterException**

PrinterException is thrown when an error occurs while calling the method.
For the details, see "4.4.4(1) **PrinterException**".

Description When this method is executed while the following method is being processed, the process is aborted. The unsent or unreceived data of following methods are discarded.

- **Write**
- **WriteAndWaitResponse**
- **ControlTransaction** (When **TRANSACTION_PRINT** is selected in *control*)
- **PrintText**
- **SendDataFile**
- **GetPrinterInformation**
- **GetPrinterInformationNumber**
- **GetPrinterInformationString**

While this method is being executed, the printer status becomes offline.

After this method is executed, wait for a few seconds before executing data transmission method. Data transmission during reset may cause data lost.

For Bluetooth connection, when this method is executed but the printer is in the condition of no data accepting, this method succeeds, but the printer reset is not executed until the printer is ready to print. And in the meantime, data transmission cannot be performed.

This method does not support USB connection.

Retrieves the latest printer status.

Syntax public PrinterStatus **GetStatus()**

Return value Printer status is returned by **PrinterStatus** class.
According to the method in **PrinterStatus** class, the printer status can be retrieved.
For the method in **PrinterStatus** class, see "4.4.2(2) **PrinterStatus**".

Exception **PrinterException**

PrinterException is thrown when an error occurs while calling the method.
For the details, see "4.4.4(1) **PrinterException**".

Starts the callback according to the printer status change.

Syntax public void **StartCallbackFunction**(StatusCallbackHandler *handler*)

Parameter *handler* Instance of **StatusCallbackHandler**

Exception **PrinterException**

PrinterException is thrown when an error occurs while calling the method.
For the details, see "4.4.4(1) **PrinterException**".

Description The process to be executed by callback is registered in **StatusChanged**.

StopCallbackFunction

Finish callback of printer status change

Finishes the callback started by **StartCallbackFunction**.

Syntax public void **StopCallbackFunction**()

Exception **PrinterException**
 PrinterException is thrown when an error occurs while calling the method.
 For the details, see "4.4.4(1) **PrinterException**".

RegisterStyleSheet

Register style sheet to printer

Registers the style sheet to the printer.

Syntax public void **RegisterStyleSheet**(String *filePath*, Int32 *regNum*)

Parameter *filePath* File path for style sheet

 regNum Style sheet number
 The valid range is 1 to 4.

Exception **PrinterException**
 PrinterException is thrown when an error occurs while calling the method.
 For the details, see "4.4.4(1) **PrinterException**".

Description A maximum of 4 sheets can be registered.

 The file extension of supported style sheet is .css.

 The maximum file size that can be registered is 1 MB (1048576 bytes).

 The maximum number of styles that can be registered in 1 style sheet is 64.

 When specify a style sheet number with which style sheet is already registered, the style sheet is overwritten.

 For the available style sheet to register, see the printer command "Register Style Sheet " in "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

 The timeout period specified in **SetWriteTimeout** is valid.

DeleteStyleSheet

Delete style sheet in printer

Deletes the style sheet registered to the printer.

Syntax public void **DeleteStyleSheet**(Int32 *regNum*)

Parameter *regNum* Style sheet number
 The valid range is 1 to 4.

Exception **PrinterException**
 PrinterException is thrown when an error occurs while calling the method.
 For the details, see "4.4.4(1) **PrinterException**".

Registers the image data to a printer as a logo.
 The method of syntax (a), dithering is fixed to be enabled.
 The method of syntax (b), dithering can be specified.

Syntax	(a) public void RegisterLogo (String <i>filePath</i> , Int32 <i>regNum</i>)	
	(b) public void RegisterLogo (String <i>filePath</i> , Int32 <i>regNum</i> , Dithering <i>dithering</i>)	
Parameter	<i>filePath</i>	File path of image data
	<i>regNum</i>	Logo number The valid range is 0 to 99.
	<i>dithering</i>	Dithering See "4.4.3(3) Dithering " for available settings.
Exception	PrinterException PrinterException is thrown when an error occurs while calling the method. For the details, see "4.4.4(1) PrinterException ".	
Description	<p>The file extension of supported image data is .bmp. Bitmap data supports monochrome (binary) image only. The image data is converted to printable format for the printer from bitmap data and is sent to the printer. Execute this method in an environment that enough memory capacity is implemented. Even when the sending of bitmap data is interrupted for a certain reason, this method continues sending the remaining data as a bitmap up to the file size specified by this method. In that case, execute the following by the interface.</p> <ul style="list-style-type: none"> • When using Bluetooth or TCP/IP: Execute Reset, or restart the printer. • When using USB: Restart the printer. <p>The maximum file size that can be registered is 1 MB (1048576 bytes).</p> <p>The maximum image sizes that can be registered are 8192 dots in width and 2304 dots in height.</p> <p>The registration status of the logo registered by this method can be confirmed by executing GetPrinterInformation or GetPrinterInformationString with specifying GET_NV_MEM_KEYCODE_LIST in <i>prnInfo</i>.</p> <p>When specify a logo number with which logo is already registered, the logo is overwritten.</p> <p>The timeout period specified in SetWriteTimeout is valid.</p>	

Deletes the logo registered to the printer.

Syntax	public void DeleteLogo (Int32 <i>regNum</i>)	
Parameter	<i>regNum</i>	Logo number The valid range is 0 to 99.

Exception	PrinterException PrinterException is thrown when an error occurs while calling the method. For the details, see "4.4.4(1) PrinterException ".
Description	The deletion status of the logo deleted by this method can be confirmed by executing GetPrinterInformation or GetPrinterInformationString with specifying GET_NV_MEM_KEYCODE_LIST in <i>prnInfo</i> .

GetPrinterInformation	
GetPrinterInformationNumber	
GetPrinterInformationString	Retrieve printer information

Retrieves the printer information.

Syntax	(a) public Byte[] GetPrinterInformation (PrinterInformation <i>prnInfo</i>) (b) public Int32 GetPrinterInformationNumber (PrinterInformation <i>prnInfo</i>) (c) public String GetPrinterInformationString (PrinterInformation <i>prnInfo</i>)
Parameter	<i>prnInfo</i> Printer information type to retrieve See "4.4.3(4) PrinterInformation " for available settings and a list of printer information to retrieve. See "MP-A40 SERIESTHERMAL PRINTER TECHNICAL REFERENCE" for details of printer information to retrieve.
Return value	Printer information
Exception	PrinterException PrinterException is thrown when an error occurs while calling the method. For the details, see "4.4.4(1) PrinterException ".
Description	Sends printer command to the printer for responding the printer information based on the specification in <i>prnInfo</i> , analyzes the response data from the printer by the response extension, and returns in numeric array, numerical value, or character string. For all PrinterInformation , the printer information can be retrieved in the numeric array by using the syntax (a).

For the following **PrinterInformation**, the printer information can be retrieved in the numerical value by using the syntax (b).

- **GET_NV_MEM_CAP**
- **GET_NV_MEM_REM_CAP**
- **GET_REM_USER_MEM_CAP**
- **GET_REM_USER_MEM_CAP_DEFRAG**
- **GET_PRN_ID_MODEL**
- **GET_PRN_ID_TYPE**
- **GET_PRN_ID_ROM_VER**
- **GET_PRN_ID_FIRM_CHECKSUM_BOOT**
- **GET_PRN_ID_FIRM_CHECKSUM_MAIN**
- **GET_PRN_ID_FIRM_CHECKSUM**
- **GET_MAINT_NUM_FEED_LINE**
- **GET_MAINT_NUM_HEAD_ACTIVE**
- **GET_MAINT_DRIVE_TIME**
- **GET_MAINT_NUM_FEED_LINE_INTEGRATION**
- **GET_MAINT_NUM_HEAD_ACTIVE_INTEGRATION**
- **GET_MAINT_DRIVE_TIME_INTEGRATION**
- **GET_HFONT_24_CHECKSUM**
- **GET_HFONT_16_CHECKSUM**
- **GET_FFONT_CHECKSUM**
- **GET_FFONT_DATA_SIZE**

For the following **PrinterInformation**, the printer information can be retrieved in the character string by using the syntax (c).

- **GET_NV_MEM_KEYCODE_LIST**
- **GET_PRN_ID_FIRM_VER_MAIN**
- **GET_PRN_ID_MFR**
- **GET_PRN_ID_MODEL_NAME**
- **GET_PRN_ID_FIRM_VER_BOOT**
- **GET_HFONT_24_ID**
- **GET_HFONT_24_INT_CHAR**
- **GET_HFONT_16_ID**
- **GET_HFONT_16_INT_CHAR**
- **GET_FFONT_LANG**
- **GET_FFONT_STANDARD**
- **GET_FFONT_COMPANY**

The timeout period specified in **SetResponseTimeout** is valid.

This method is aborted by **Reset**.

ControlTransaction

Start/finish print data batch sending

Buffers the send data in the target method for batch processing and sends the data in send buffer to the printer.

Syntax public void **ControlTransaction**(TransactionFunction *control*)

Parameter *control* Operation selection of batch processing
See "4.4.3(5) **TransactionFunction**" for available settings.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "4.4.4(1) **PrinterException**".

Description The target method for the batch processing is as follows.

- **Write**
- **SelectStandardMode**
- **SelectPageMode**
- **PrintPageModeData**
- **SetPageModeVerticalPosition**
- **SetStandardModeArea**
- **SetStandardModeAlignment**
- **SetHorizontalPosition**
- **SetStandardModeBarcodeDirection**
- **SetLineSpacing**
- **SetCharacterRightSpace**
- **SetCharacterFormatting**
- **PrintText**
- **PrintLogo**
- **SendDataFile**
- **PrintBarcode**
- **Print2Dcode**
- **PrintPageModeRectangle**
- **FeedLine**
- **FeedDotLine**
- **FeedCutPosition**
- **FeedMarkPosition**

By *control*, buffering of the send data in the target method for batch processing is started or finished.

• Start buffering

When this method with **TRANSACTION_START** in *control* is executed, the buffering of the send data in the target method for batch processing is started.

The send data of the target method for batch processing executed during buffering is buffered in the send buffer without being sent to the printer.

The maximum send data to be buffered is 10 MB (10485760 bytes).

If the buffered send data exceeds the maximum size, the method for batch processing at the time of exceeding becomes an error. If an error occurs, the send data up to the error is retained, so finish the batch processing in " • Finish buffering".

Any method other than the target method for batch processing is immediately executed.

• Finish buffering

When this method with **TRANSACTION_PRINT** in *control* is executed, the buffering of the send data in the target method for batch processing is finished, and the data in the send buffer is sent to the printer.

When this method with **TRANSACTION_CLEAR** in *control* is executed, the buffering is interrupted and the data in the send buffer is discarded.

When this method with **TRANSACTION_PRINT** or **TRANSACTION_CLEAR** in *control* is executed without starting buffering, an error occurs.

When this method with **TRANSACTION_START** in *control* is executed during buffering, the data in the send buffer is discarded but the buffering is continued.

When no data exists in the send buffer and this method with **TRANSACTION_PRINT** in *control* is executed, the data is not sent to the printer and the buffering is finished.

When this method with **TRANSACTION_PRINT** in *control* is being executed and this method with **TRANSACTION_START** in *control* is executed from another thread, the data sending by this method with **TRANSACTION_PRINT** in *control* is continued, this method with **TRANSACTION_START** in *control* is executed from another thread, and the buffering is newly started.

When the buffered send data is being sent by **TRANSACTION_PRINT**, the timeout period specified in **SetWriteTimeout** is valid.

When an error occurs during sending data by **TRANSACTION_PRINT**, the remaining data for sending is discarded.

When **Reset** is executed during sending data by **TRANSACTION_PRINT**, the remaining data for sending is discarded.

SelectStandardMode	Select standard mode
--------------------	----------------------

Starts standard mode.

Syntax public void **SelectStandardMode**()

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "4.4.4(1) **PrinterException**".

Description Right after execution of **Open**, standard mode is selected.

When page mode is already started by **SelectPageMode**, the page mode is finished and the page mode data is discarded.

SelectPageMode	Start page mode
----------------	-----------------

Starts page mode and specify the print area and direction.

Syntax public void **SelectPageMode**(Int32 *startX*,
Int32 *startY*,
Int32 *width*,
Int32 *height*,
Direction *direction*)

Parameter	<i>startX</i>	Horizontal start point (dot) The valid range is 0 to 2399.
	<i>startY</i>	Vertical start point (dot) The valid range is 0 to 2399.
	<i>width</i>	Print area width (dot) The valid range is 1 to 2400.
	<i>height</i>	Print area height (dot) The valid range is 1 to 2400.
	<i>direction</i>	Print direction (dot) See "4.4.3(6) Direction " for available settings.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "4.4.4(1) **PrinterException**".

Description	<p>When <i>startX</i> exceeds the print width, the print area setting by this method is disabled.</p> <p>When (<i>startX</i> + <i>width</i>) exceeds the print width, <i>startX</i> is set to the specified value and <i>width</i> is set to (print width – <i>startX</i>).</p> <p>When (<i>startY</i> + <i>height</i>) is 2400 or more, <i>startY</i> is set to the specified value and <i>height</i> is set to (2400 – <i>startY</i>).</p> <p>The print width can be specified in the function settings. For the print width and the function settings, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".</p> <p>Arrange character, image, barcode, and 2-dimensional barcode at any print position within the specified print area and configure the page mode data by SetHorizontalPosition and SetPageModeVerticalPosition.</p> <p>When page mode is interrupted by SelectStandardMode, the page mode data will be discarded.</p> <p>When the printer is initialized, it is back to standard mode.</p> <p>For printer initialization, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".</p>
-------------	---

PrintPageModeData	Print page mode data
-------------------	----------------------

Prints the page mode data

Syntax public void **PrintPageModeData**()

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "4.4.4(1) **PrinterException**".

Description Execute this method after starting page mode by **SelectPageMode** and configuring the page mode data within the print area.

This method is ignored in standard mode.

SetPageModeVerticalPosition	Specify vertical absolute position in page mode
-----------------------------	---

Specifies the absolute position of print data in vertical direction within the print area in page mode.

Syntax public void **SetPageModeVerticalPosition**(Int32 *verticalPosition*)

Parameter *verticalPosition* Data mapping starting position in vertical direction (dot)
The valid range is 0 to 2399.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "4.4.4(1) **PrinterException**".

Description The start point varies depending on the setting of *direction* (print direction) in **SelectPageMode**. For the print direction and the start point, see "4.4.3(6) **Direction**".

The setting of this method is disabled in standard mode.

When the specified value exceeds the print area, it is ignored.

When the printer is initialized, the setting in this method is disabled.

For printer initialization, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

Specifies the print area in standard mode.

Syntax `public void SetStandardModeArea(Int32 leftMargin, Int32 prnAreaWidth)`

Parameter *leftMargin* Left margin position (dot)
The valid range is 0 to 831.

prnAreaWidth Print area width (dot)
The valid range is 1 to 832.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "4.4.4(1) **PrinterException**".

Description The relation between left margin and print area width is shown in Figure 4-1. The print data is mapped to the shaded print area.

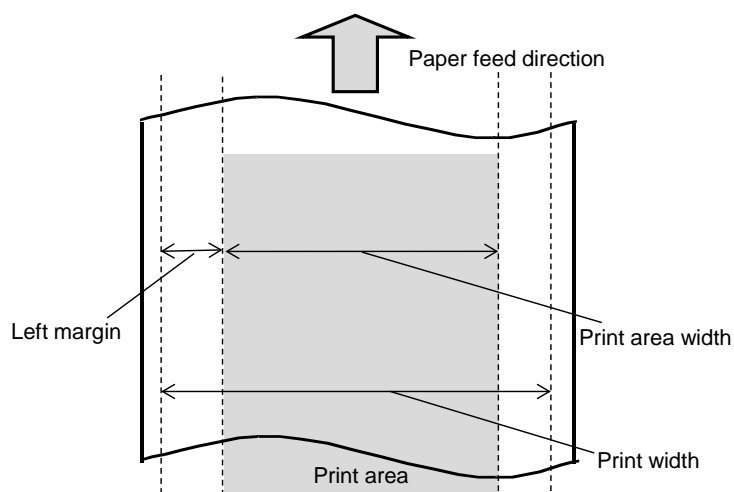


Figure 4-1

The setting of this method is disabled in page mode. When this method is executed in page mode, the setting is reflected to the print after starting standard mode.

When the print data remains in the printer, this method is not executed. Execute this method after the printing is completed.

When the specified value in *leftMargin* exceeds the print width, the value is set to the print width. When the specified value in *prnAreaWidth* exceeds the print width, the value is set to the difference between the print width and the left margin.

The print width can be specified in the function settings. For the print width and the function settings, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

When the printer is initialized, *leftMargin* is set to 0, and *prnAreaWidth* is set to the print width. For printer initialization, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

Alignment

Syntax `public void SetStandardModeAlignment(Alignment align)`

Exception	PrinterException PrinterException is thrown when an error occurs while calling the method. For the details, see "4.4.4(1) PrinterException ".
-----------	--

The setting of this method is disabled in page mode. When this method is executed in page mode, the setting is reflected to the print after starting standard mode.

When the print data remains in the printer, this method is not executed. Execute this method after the printing is completed.

When the printer is initialized, the parameter of this method is set to the initial value. For printer initialization, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

Specify absolute position

Syntax `public void SetHorizontalPosition(Int32 horizontalPosition)`

Exception	PrinterException PrinterException is thrown when an error occurs while calling the method. For the details, see "4.4.4(1) PrinterException ".
-----------	--

In page mode, the start point changes depending on the specified print direction in *direction* of **SelectPageMode**. For print direction and start point, see "4.4.3(6) **Direction**".

Specification that exceeds the print area set in **SetStandardModeArea** or **SelectPageMode** is ignored.

When the printer is initialized, the setting in this method is disabled.
For printer initialization, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL
REFERENCE".

SetStandardModeBarcodeDirection

Select print direction for barcode or 2-dimensional barcode

Specifies the print direction for barcode or 2-dimensional barcode in standard mode.

Syntax public void **SetStandardModeBarcodeDirection**(Rotate *rotate*)

Parameter *rotate* Print direction
When **ROTATE_90_TO_RIGHT** or **ROTATE_90_TO_LEFT** is specified, the width of printable barcode and 2-dimensional barcode is 300 mm max. When exceeds 300 mm, barcode and 2-dimensional barcode are not printed.
See "4.4.3(15) **Rotate**" for available settings.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "4.4.4(1) **PrinterException**".

Description This method can be used only in standard mode. When this method is executed in the page mode, the setting is reflected in printing after starting standard mode.
When using this method, execute **SelectStandardMode** beforehand and select standard mode.

By executing **PrintBarcode** or **Print2Dcode**, the setting of this method is back to the initial value.

When the printer is initialized, the parameter of this method is set to the initial value.
For printer initialization, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

Note The print position and the orientation of barcode or 2-dimensional barcode may be changed by *rotate* of **SetCharacterFormatting**. For the print position and the orientation, see "4.4.3(15) **Rotate**".

SetLineSpacing

Specify line space amount

Specifies the line spacing.

Syntax public void **SetLineSpacing**(Int32 *lineSpacing*)

Parameter *lineSpacing* Line spacing (dot)
The valid range is 0 to 255.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "4.4.4(1) **PrinterException**".

Description The line spacing can be independently set in the standard mode and the page mode.

When the printer is initialized, *lineSpacing* is set to 34.
For printer initialization, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

SetCharacterRightSpace

Specify character right space amount

Specifies the character right space amount.

Syntax public void **SetCharacterRightSpace**(Int32 *space*)

Parameter *space* Right space amount (dot)
The valid range is 0 to 255.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "4.4.4(1) **PrinterException**".

Description The right space amount can be independently set in the standard mode and the page mode.

This method is enabled for 1-byte characters and 2-byte characters. For 2-byte characters, the left space is set to 0.

When the printer is initialized, *space* is set to 0.
For printer initialization, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

SelectCharacterSet

Select character set

Selects the character set.

Syntax public void **SelectCharacterSet**(CharacterSet *charSet*)

Parameter *charSet* Character set
See "4.4.3(8) **CharacterSet**" for available settings.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "4.4.4(1) **PrinterException**".

Description According to the language setting in WinCE device, the initial value of the character set is as follows.

Japanese: **CODEPAGE_KATAKANA**
Other than Japanese: **CODEPAGE_1252**

Even when **Reset** is executed and the printer is initialized after executing this method, the setting in this method is still enabled. For printer initialization, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

GetCharacterSet

Retrieve specified character set

Retrieves the specified character set.

Syntax public CharacterSet **GetCharacterSet**()

Return value Character set

Description For the available setting to retrieve, see "4.4.3(8) **CharacterSet**".

SelectInternationalCharacterSet

Select international character set

Selects the international character set.

Syntax public void **SelectInternationalCharacterSet**(InternationalCharacterSet *intCharSet*)

Parameter *intCharSet* International character set
See "4.4.3(9) **InternationalCharacterSet**" for available settings.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "4.4.4(1) **PrinterException**".

Description According to the language setting in WinCE device, the initial value of the International character set is as follows.

Japanese: **INT_CHAR_SET_JAPAN**
Other than Japanese: **INT_CHAR_SET_USA**

Even when **Reset** is executed and the printer is initialized after executing this method, the setting in this method is still enabled. For printer initialization, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

GetInternationalCharacter

Retrieve specified international character set

Retrieves the specified international character set.

Syntax public InternationalCharacterSet **GetInternationalCharacter**()

Return value International character set

Description For the available setting to retrieve, see "4.4.3(9) **InternationalCharacterSet**".

SetCharacterFormatting

Format character

Sets the character formatting.

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

Parameter *type* Character font
See "4.4.3(10) **CharacterType**" for available settings.

verticalScale Character size (in vertical direction)
See "4.4.3(11) **CharacterScale**" for available settings.

horizontalScale Character size (in horizontal direction)
See "4.4.3(11) **CharacterScale**" for available settings.

underline Underline
See "4.4.3(12) **Underline**" for available settings.

bold Bold print
See "4.4.3(13) **Bold**" for available settings.

<i>reverse</i>	Reverse print See "4.4.3(14) Reverse " for available settings.
<i>rotate</i>	Character rotation print In page mode, the setting in <i>rotate</i> is disabled. When this method is executed in page mode, the setting is reflected to the print after starting standard mode. In standard mode, when the print data remains in the printer, this method is not executed. Execute this method after the printing is completed. In standard mode, when ROTATE_90_TO_RIGHT or ROTATE_90_TO_LEFT is specified, the operation is as follows: <ul style="list-style-type: none"> • Underline print is not performed. • The scales of height and width in <i>verticalScale</i> and <i>horizontalScale</i> are reversed. Specifying <i>rotate</i> may affect the print position and the orientation of barcode or 2-dimensional barcode. See "4.4.3(15) Rotate " for available settings, and the print position and the orientation of barcode or 2-dimensional barcode.
Exception	PrinterException PrinterException is thrown when an error occurs while calling the method. For the details, see "4.4.4(1) PrinterException ".
Description	When the printer is initialized, the parameter of this method is set to the initial value. For printer initialization, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

PrintText		Send text data
Sends the text data.		
Syntax	public void PrintText (String <i>text</i>)	
Parameter	<i>text</i>	Text data to send to the printer
Exception	PrinterException PrinterException is thrown when an error occurs while calling the method. For the details, see "4.4.4(1) PrinterException ".	
Description	<p>This method sends the text data specified in <i>text</i> after encoding it to recognizable text data for the printer based on the character set specified in SelectCharacterSet.</p> <p>Printer commands "Select Character Code Table", "Select International Character Set", "Select Kanji Code System", and "Cancel Kanji Mode" are added to the data to send as the header.</p> <p>The printer commands "Select Character Code Table" and "Select International Character Set" are sent on the basis of the settings of SelectCharacterSet and SelectInternationalCharacterSet.</p> <p>For the printer command "Select Kanji Code System", the Kanji code system is selected as follows on the basis of the setting of SelectInternationalCharacterSet.</p> <p>When SelectInternationalCharacterSet is INT_CHAR_SET_JAPAN: Shift-JIS When SelectInternationalCharacterSet is other than INT_CHAR_SET_JAPAN: JIS</p> <p>For the details of printer commands, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".</p> <p>This method does not add a line feed code at the end of the text data.</p> <p>A maximum of 16 KB (16384 bytes) of data size can be specified.</p>	

When **ControlTransaction** is not used, the timeout period specified in **SetWriteTimeout** is valid.

This method is aborted by **Reset**.

PrintLogo

Print logo registered in printer

Prints the logo registered in the printer.

Syntax public void **PrintLogo**(Int32 *regNum*)

Parameter *regNum* Logo number
The valid range is 0 to 99.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "4.4.4(1) **PrinterException**".

Description Register the logo by **RegisterLogo**.

When a logo is not registered in the specified *regNum*, this method is ignored.

SendDataFile

Send file data

Sends the file data to the printer.
The method of syntax (a), dithering is fixed to be enabled.
The method of syntax (b), dithering can be specified.

Syntax (a) public void **SendDataFile**(String *filePath*)

 (b) public void **SendDataFile**(String *filePath*, Dithering *dithering*)

Parameter *filePath* Path of the file to the printer

 dithering Dithering
It is enabled when the extension of the file specified at *filePath*
is .bmp.
See "4.4.3(3) **Dithering**" for available settings.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "4.4.4(1) **PrinterException**".

Description Depending on the extension of the specified file, the processing is as follows.

- When the file extension is .bmp:
 Bitmap data supports monochrome (binary) image only.
 The image data is converted to printable format for the printer from bitmap data and is sent to the printer.
 Execute this method in an environment that enough memory capacity is implemented.
 Even when the sending of bitmap data is interrupted for a certain reason, this method continues processing the remaining data as a bitmap up to the file size specified by this method. In that case, execute the following by the interface.
- When using Bluetooth or TCP/IP:
 Execute **Reset**, or restart the printer.
- When using USB:
 Restart the printer.

- When the file extension is .txt:
Text data format supports UTF-8.
This method sends the text data after encoding it to recognizable text data for the printer based on the character set specified in **SelectCharacterSet**.
Printer commands "Select Character Code Table", "Select International Character Set", "Select Kanji Code System", and "Cancel Kanji Mode" are added to the data to send as the header.
The printer commands "Select Character Code Table" and "Select International Character Set" are sent on the basis of the settings of **SelectCharacterSet** and **SelectInternationalCharacterSet**.
For the printer command "Select Kanji Code System", the Kanji code system is selected as follows on the basis of the setting of **SelectInternationalCharacterSet**.
When **SelectInternationalCharacterSet** is INT_CHAR_SET_JAPAN: Shift-JIS
When **SelectInternationalCharacterSet** is other than INT_CHAR_SET_JAPAN: JIS
For the details of printer commands, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".
This method does not add a line feed code at the end of the text data.
- When the file extension is .bin, or .dat:
The text data is sent to printer without conversion.
- When the file extension is .htm, or .html:
The text data is sent to printer without conversion.
Be sure to specify HTML end tag at the end of file data.

A maximum of 1 MB (1048576 bytes) of file size can be specified.

When **ControlTransaction** is not used, the timeout period specified in **SetWriteTimeout** is valid.

Do not include a printer command that initializes the printer other than the printer command "Initialize Printer" in the data to send.

For printer initialization, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

When performing a hardware reset, execute **Reset**.

This method is aborted by **Reset**.

PrintBarcode

Print barcode

Prints the barcode.

Syntax	(a) public void PrintBarcode (TypeBarcode <i>type</i> , String <i>text</i> , ModuleWidthBarcode <i>moduleWidth</i> , Int32 <i>moduleHeight</i> , HriPositionBarcode <i>hri</i> , NwRatioBarcode <i>nwRatio</i>)
	(b) public void PrintBarcode (TypeBarcode <i>type</i> , Byte[] <i>binary</i> , ModuleWidthBarcode <i>moduleWidth</i> , Int32 <i>moduleHeight</i> , HriPositionBarcode <i>hri</i> , NwRatioBarcode <i>nwRatio</i>)
Parameter	<i>type</i> Barcode type See "4.4.3(16) TypeBarcode " for available settings. (a) <i>text</i> : Barcode text data (b) <i>binary</i> : Barcode binary data

The available values in *text* and *binary* are as follows.

Barcode Type	Data Size	Barcode Data to Input
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	' ' (0x20)
		'\$' (0x24)
		'%' (0x25)
		'+' (0x2B)
		'-' (0x2D)
		':' (0x2E)
		'/' (0x2F)
		'0' to '9' (0x30 to 0x39)
		'A' to 'Z' (0x41 to 0x54)
ITF	2 to 15 bytes (Even number only)	'0' to '9' (0x30 to 0x39)
CODABAR	1 to 150 bytes	'\$' (0x24)
		'+' (0x2B)
		'-' (0x2D)
		':' (0x2E)
		'/' (0x2F)
		'0' to '9' (0x30 to 0x39)
		':' (0x3A)
		'A' to 'D' (0x41 to 0x44)
CODE93	1 to 150 bytes	(0x00 to 0x7F)
CODE128	2 to 150 bytes	(0x00 to 0x7F) ^{*1}
JAN13 add-on 2 (EAN13 add-on 2)	14 to 15 bytes	'0' to '9' (0x30 to 0x39)
JAN13 add-on 5 (EAN13 add-on 5)	17 to 18 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 to 150 bytes	' ' 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 barcode data varies depending on the specified code set.
CODE A : 0x00 to 0x50
CODE B : 0x00 to 0x7F
CODE C : 2 digit numbers from 00 to 99 (0x00 to 0x63)

moduleWidth Barcode module width or fine element
See "4.4.3(17) **ModuleWidthBarcode**".

moduleHeight Barcode module height(dot)
See the following table for available settings.

Barcode Type	<i>moduleHeight</i> (dot)
UPC-A	0: Initial value (162 dots), 1 to 255
UPC-E	
JAN13 (EAN13)	
JAN8 (EAN8)	
CODE39	
ITF	
CODABAR	
CODE93	
CODE128	
JAN13 add-on 2 (EAN13 add-on 2)	
JAN13 add-on 5 (EAN13 add-on 5)	
GS1 Databar Omni-directional	0: Initial value (162 dots), 66 to 255
GS1 Databar Truncated	0: Initial value (162 dots), 26 to 255
GS1 Databar Limited	0: Initial value (162 dots), 20 to 255
GS1 Databar Expanded	0: Initial value (162 dots), 68 to 255

hri Barcode HRI character
See "4.4.3(18) **HriPositionBarcode**" for available settings.

nwRatio Barcode N:W ratio
The available setting varies depending on the setting of *type*.
For the details, see "4.4.3(19) **NwRatioBarcode**".

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "4.4.4(1) **PrinterException**".

Description In standard mode, when the print data remains in the printer, this method is not executed.
Execute this method after the printing is completed.

When the barcode width exceeds the print area, the barcode is not printed.

This method sends the text data specified in syntax (a) *text* after encoding it to recognizable text data for the printer.

The binary data specified in syntax (b) *binary* is sent to the printer as it is.

Special codes to specify in CODE128 are as follows.

Input Data	Special Code per Code Set		
	CODE A	CODE B	CODE C
'{S' (0x7B, 0x53)	SHIFT	SHIFT	SHIFT
'{A' (0x7B, 0x41)	-	CODE A	CODE A
'{B' (0x7B, 0x42)	CODE B	-	CODE B
'{C' (0x7B, 0x43)	CODE C	CODE C	-
'{1' (0x7B, 0x31)	FNC 1	FNC 1	FNC 1
'{2' (0x7B, 0x32)	FNC 2	FNC 2	FNC 2
'{3' (0x7B, 0x33)	FNC 3	FNC 3	FNC 3
'{4' (0x7B, 0x34)	FNC 4	FNC 4	FNC 4
'{' (0x7B, 0x7B)	'{'	'{'	-

Check digits for the following barcodes are automatically calculated.

- UPC-A
- UPC-E
- JAN13 (EAN13)
- JAN8 (EAN8)
- JAN13 add-on 2 (EAN13 add-on 2)
- JAN13 add-on 5 (EAN13 add-on 5)
- GS1 Databar Omni-directional
- GS1 Databar Truncated
- GS1 Databar Limited

When the data including the check digit is specified for the following barcodes, the check digit is ignored and recalculated on the printer.

- UPC-A
- UPC-E
- JAN13 (EAN)
- JAN8 (EAN)

Specifying the start code and stop code ('*') of CODE39 is not required since the codes are added automatically.

When odd number of data is specified in ITF barcode data, the last data is discarded.

In CODABAR barcode data, input 'A' to 'D' as the start code and stop code.

In GS1 Databar Expanded barcode data, use '{' only for specifying FUNC1. To specify FUNC1, input '{1'(0x7B, 0x31).

When specified value in any one of *text*, *binary*, *moduleWidth*, *moduleHeight*, or *nwRatio* is not corresponded to the barcode type specified in *type*, an error occurs.

When **ControlTransaction** is not used, the timeout period specified in **SetWriteTimeout** is valid.

Prints the 2-dimensional barcode.

Syntax (a) public void **Print2Dcode**(Type2Dcode *type*,
String *text*,
Mode2Dcode *mode*,
ModuleSize2Dcode *moduleSize*,
Int32 *moduleHeight*,
Int32 *column*,
Int32 *row*,
ErrorCorrect2Dcode *errorCorrect*)

(b) public void **Print2Dcode**(Type2Dcode *type*,
Byte[] *binary*,
Mode2Dcode *mode*,
ModuleSize2Dcode *moduleSize*,
Int32 *moduleHeight*,
Int32 *column*,
Int32 *row*,
ErrorCorrect2Dcode *errorCorrect*)

Parameter *type* 2-dimensional barcode type
See "4.4.3(20) **Type2Dcode**" for available settings.
(a) *text* : Barcode text data
(b) *binary* : Barcode binary data

mode 2-dimensional barcode mode
The available setting varies depending on the setting of *type*.
For the details, see "4.4.3(21) **Mode2Dcode**".
When select **MODE_2DCODE_MAXI_CODE_2**, add the service class (3 digits), the country code (3 digits), and the postal code (9 digits) data in the beginning of *text* or *binary*.
When select **MODE_2DCODE_MAXI_CODE_3**, add the service class (3 digits), the country code (3 digits), and the postal code (6 digits) data in the beginning of *text* or *binary*.

moduleSize 2-dimensional barcode module size
The available setting varies depending on the setting of *type*.
For the details, see "4.4.3(22) **ModuleSize2Dcode**".

moduleHeight 2-dimensional barcode module height (dot)
moduleHeight is enabled in PDF417 and GS1 Databar Stacked Omni-directional.
When using other barcode types, specify 0.

2-dimensional Barcode Type	<i>moduleHeight</i>	
	PDF Module Height	Height of 1 Row
PDF417	2 to 127	-
GS1 Databar Stacked Omni-directional	-	33 to 255

column

Number of columns in 2-dimensional barcode

column is enabled in PDF417 and GS1 Databar Expanded Stacked.
When using other barcode types, specify 0.

2-dimensional Barcode Type	<i>column</i>	
	Number of Columns in Data Area	Number of Segments in 1 Line ^{*1}
PDF417	0 ^{*2} , 1 to 30	-
GS1 Databar Expanded Stacked	-	2 to 20

*1: Specify an even number for the number of segments in 1 line.

*2: When specified 0, the number of columns is automatically set.

row

Number of rows in 2-dimensional barcode

row is enabled in PDF417 only.

When using other barcode types, specify 0.

2-dimensional Barcode Type	<i>row</i>
PDF417	0 ^{*1} , 3 to 90

*1: When specified 0, the number of rows is automatically set.

errorCorrect

Error correction level of 2-dimensional barcode

The available setting varies depending on the setting of *type*.

For the details, see "4.4.3(23) **ErrorCorrect2DCode**".

Exception

PrinterException

PrinterException is thrown when an error occurs while calling the method.

For the details, see "4.4.4(1) **PrinterException**".

Description

In standard mode, when the print data remains in the printer, this method is not executed.
Execute this method after the printing is completed.

This method sends the text data specified in syntax (a) *text* after encoding it to recognizable text data for the printer.

In *text*, input the corresponded character data in ASCII format, according to the standard of 2-dimensional barcode type specified in *type*.

The binary data specified in syntax (b) *binary* is sent to the printer as it is.

In *binary*, input the corresponded character data in ASCII format character code according to the standard of 2-dimensional barcode type specified in *type*.

When specified value in any one of *text*, *binary*, *mode*, *moduleSize*, *moduleHeight*, *column*, or *errorCorrect* is not corresponded to the 2-dimensional barcode type specified in *type*, an error occurs.

When the 2-dimensional barcode width exceeds the print area width, this method is ignored.

When **ControlTransaction** is not used, the timeout period specified in **SetWriteTimeout** is valid.

Draws a rectangle in page mode.

Syntax	public void PrintPageModeRectangle (Int32 <i>startX</i> , Int32 <i>startY</i> , Int32 <i>width</i> , Int32 <i>height</i> , Int32 <i>thickness</i>)	
Parameter	<i>startX</i>	Horizontal start point (dot) The valid range is 0 to 2395.
	<i>startY</i>	Vertical start point (dot) The valid range is 0 to 2395.
	<i>width</i>	Width (dot) The valid range is 4 to 2400.
	<i>height</i>	Height (dot) The valid range is 4 to 2400.
	<i>thickness</i>	Line width (dot) The valid range is 2 to 40.
Exception	PrinterException PrinterException is thrown when an error occurs while calling the method. For the details, see "4.4.4(1) PrinterException ".	
Description	Do not execute this method in standard mode.	
	The start point varies depending on the print direction in page mode specified in <i>direction</i> of SelectPageMode . For the details of print direction and start point, see "4.4.3(6) Direction ".	
	When the setting in <i>startX</i> , <i>startY</i> , <i>width</i> , or <i>height</i> exceeds the print area in page mode specified in SelectPageMode , the exceeded area is not drawn.	
Note	In this method, rectangle drawing is realized using the ruled line command of the printer. Note that when this method ends, the printer commands "Set Ruled Line OFF" and "Clear Ruled Line Buffer" are sent, and the ruled line buffer of the printer is cleared. For the ruled line command, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".	

Feeds the paper by line.

Syntax	public void FeedLine (Int32 <i>lines</i>)	
Parameter	<i>lines</i>	Number of lines to feed (line) The valid range is 0 to 255.
Exception	PrinterException PrinterException is thrown when an error occurs while calling the method. For the details, see "4.4.4(1) PrinterException ".	
Description	In standard mode, feeds the paper by the specified number of lines. When data exists in the line buffer of the printer, paper feed is performed after printing 1 line.	
	In page mode, the vertical data mapping starting position is shifted by the specified number of lines.	

The paper feed length for 1 line (line spacing) can be specified in **SetLineSpacing**.

FeedDotLine

Feed paper by dot

Feeds the paper by dot.

Syntax public void **FeedDotLine**(Int32 *dotLines*)

Parameter *dotLines* Number of dots to feed (dot)
In standard mode, the valid range is -48 to 8192.
In page mode, the valid range is 0 to 8192. When the specified value is from -48 to -1, this method is ignored.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "4.4.4(1) **PrinterException**".

Description In standard mode, feeds the paper by the specified number of dots. When data exists in the line buffer of the printer, paper feed is performed after printing 1 line.
In page mode, the vertical data mapping starting point is shifted by the specified number of dots.

FeedCutPosition

Feed paper to cut position

Feeds the paper to the paper cut position.

Syntax public void **FeedCutPosition**()

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "4.4.4(1) **PrinterException**".

Description In standard mode, when the print data remains in the printer, this method is not executed.
Execute this method after the printing is completed.

FeedMarkPosition

Marked paper form feed

Performs the marked paper form feed and the form feed position correct.

Syntax public void **FeedMarkPosition**(Int32 *dotLines*)

Parameter *dotLines* Correction amount (dot)
The valid range is -48 to 255.
When correction is not needed, specify 0.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "4.4.4(1) **PrinterException**".

Description In standard mode, when the print data remains in the printer, this method is not executed.
Execute this method after the printing is completed.

This method is effective only when marked paper is selected.
See "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE" for details about how to use the marked paper.

Specifies the log output.

Syntax public void **SetLog**(Int32 *logLevel*, LogFileSize *logFileSize*)

Parameter	<i>logLevel</i>	Log output level Specify 0. When specified 0, an error log is output. Do not specify other than 0.
	<i>logFileSize</i>	Maximum size of log file See "4.4.3(24) LogFileSize " for the available setting.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "4.4.4(1) **PrinterException**".

Description Log file is saved under the folder where the WinCE application incorporating this library exists.

Log file name: PrinterManager.log.x (x range is 0 to 4)

First of all, the log file is created as PrinterManager.log.0.
When the file size exceeds the maximum size of the log file, the file name is changed to PrinterManager.log.1, and a new PrinterManager.log.0 is created.
Up to 5 log files are created.

When this method is not executed, an error log of 1 MB is output. The file name and the number of created log files are the same as above.

- **Public Property**

IsOpened

Retrieve printer using status

Retrieves the printer's open status by **Open**.

Syntax public Boolean **IsOpened**{get;}

Initial Value False

Return value True : Using printer has been started.
 False : Using printer is not started.

Description Since this property is read-only, the value cannot be set.

(2) **PrinterStatus**

• **Summary**

This class stores the printer status.

This class provides following functions.

Public Methods

Method	Function Summary
GetErrOffline	Retrieve offline error status
GetErrHardware	Retrieve hardware error status
GetErrVoltage	Retrieve Vp Voltage error status
GetErrHeadTemperature	Retrieve head temperature error status
GetErrOutOfPaper	Retrieve out-of-paper error status
GetErrMarkPaperJam	Retrieve paper jam error status while detecting marked paper.
GetErrCoverOpen	Retrieve paper cover open error status
GetErrBattery	Retrieve battery error status
GetStateFeedSwitch	Retrieve feed switch status
GetStatePaperFeed	Retrieve paper feed status
GetStateReturnWaiting	Retrieve return waiting status
GetStateFlashMemoryRewriting	Retrieve FLASH memory rewrite status
GetStateBattery	Retrieve battery voltage status

• **Public Methods**

GetErrOffline	Retrieve offline error status
----------------------	--------------------------------------

Retrieves the offline error status.

Syntax public Boolean **GetErrOffline()**

Return value True : Offline error
 False : Online

GetErrHardware	Retrieve hardware error status
-----------------------	---------------------------------------

Retrieves the hardware error status.

Syntax public Boolean **GetErrHardware()**

Return value True : Error
 False : OK

GetErrVoltage

Retrieve Vp Voltage error status

Retrieves the Vp Voltage error status.

Syntax public Boolean **GetErrVoltage()**

Return value True : Error
 False : OK

GetErrHeadTemperature

Retrieve head temperature error status

Retrieves the head temperature error status.

Syntax public Boolean **GetErrHeadTemperature()**

Return value True : Error
 False : OK

GetErrOutOfPaper

Retrieve out-of-paper error status

Retrieves the out-of-paper error status

Syntax public Boolean **GetErrOutOfPaper()**

Return value True : Error
 False : OK

GetErrMarkPaperJam

Retrieve paper jam error status while detecting marked paper.

Retrieves the paper jam error status while detecting marked paper.

Syntax public Boolean **GetErrMarkPaperJam()**

Return value True : Error
 False : OK

GetErrCoverOpen

Retrieve paper cover open error status

Retrieves the paper cover open error status.

Syntax public Boolean **GetErrCoverOpen()**

Return value True : Error
 False : OK

GetErrBattery

Retrieve battery error status

Retrieves the battery error status.

Syntax public Boolean **GetErrBattery()**

Return value True : Error
 False : OK

GetStateFeedSwitch

Retrieve feed switch status

Retrieves the feed switch status.

Syntax public Boolean **GetStateFeedSwitch()**

Return value True : ON
 False : OFF

GetStatePaperFeed

Retrieve paper feed status

Retrieves the paper feed status.

Syntax public Boolean **GetStatePaperFeed()**

Return value True : Feeding
 False : Stopped

GetStateReturnWaiting

Retrieve return waiting status

Retrieves the return waiting status.

Syntax public Boolean **GetStateReturnWaiting()**

Return value True : In return waiting status
 False : Not in return waiting status

GetStateFlashMemoryRewriting

Retrieve FLASH memory rewrite status

Retrieves the FLASH memory rewrite status.

Syntax public Boolean **GetStateFlashMemoryRewriting()**

Return value True : FLASH memory is being rewritten
 False : FLASH memory is not being rewritten

GetStateBattery

Retrieve battery voltage status

Retrieves the battery voltage status.

Syntax public Int32 **GetStateBattery()**

Return value 0 : No battery
 1 : Battery remaining capacity level 4 or 5
 (Need to charge or battery remaining capacity is 0%)
 2 : Battery remaining capacity level 3 (battery remaining capacity: approx. 10%)
 3 : Battery remaining capacity level 2 (battery remaining capacity: approx. 40%)
 4 : Battery remaining capacity level 1 (battery remaining capacity: approx. 80%)

(3) PrinterDiscovery

• Summary

This class provides printer search functions.
This class provides the following functions.

Public Methods

Method	Function Summary
PrinterDiscovery	Constructor
StartDiscoveryPrinter	Start printer search
CancelDiscoveryPrinter	Cancel printer search
GetFoundPrinter	Retrieve information list of found printer

• Public Methods

PrinterDiscovery Constructor

Constructor for **com.seikoinstruments.sdk.mobileprinter.PrinterDiscovery** class.

Syntax public **PrinterDiscovery**()

Exception **PrinterException**
 PrinterException is thrown when an error occurs while calling the method.
 For the details, see "4.4.4(1) **PrinterException**".

StartDiscoveryPrinter Start printer search

Starts the printer search.

Syntax public void **StartDiscoveryPrinter**(DiscoveryHandler *handler*,
 PrinterInterface *prnIf*,
 Int32 *retry*,
 Int32 *timeout*)

Parameter	<i>handler</i>	Instance of DiscoveryHandler
	<i>prnIf</i>	Communication interface constant See "4.4.3(1) PrinterInterface " for the available setting. PRN_IF_USB is not supported.
	<i>retry</i>	Number of retry times (time) The operation varies depending on the setting of <i>prnIf</i> . <ul style="list-style-type: none">• When PRN_IF_TCP is specified: Sends local broadcast packet the specified times in <i>retry</i>. The valid range is 1 to 5. When the specified value is below 1, the value is processed as 1. When the specified value exceeds 5, the value is processed as 5.• When PRN_IF_BT is specified: The specified value is ignored, and the search is performed 1 time only.

timeout Timeout period for 1 time search (second)
 The operation varies depending on the setting of *prnIf*.

- When **PRN_IF_TCP** is specified:
 Every time of sending local broadcast packet, waits for a response from the printer until the time specified by *timeout* elapses.
 The valid range is 3 to 60.
 When the specified value is below 3, the value is processed as 3.
 When the specified value exceeds 60, the value is processed as 60.
- When **PRN_IF_BT** is specified:
 The specified value is ignored.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
 For the details, see "4.4.4(1) **PrinterException**".

Description The search can be canceled by **CancelDiscoveryPrinter**. When search is canceled or completed, **DiscoveryFinished** of the instance specified in *handler* is executed.

Retrieve the search result by **GetFoundPrinter**.

CancelDiscoveryPrinter	Cancel printer search
-------------------------------	------------------------------

Cancels the printer search started by **StartDiscoveryPrinter**.

Syntax public void **CancelDiscoveryPrinter**()

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
 For the details, see "4.4.4(1) **PrinterException**".

Description When the search is canceled, **DiscoveryFinished** of the instance specified in *listener* of **StartDiscoveryPrinter** is executed.

The search result until cancellation can be retrieved by **GetFoundPrinter**.

GetFoundPrinter	Retrieve information list of found printer
------------------------	---

Retrieves all the printer information found by **StartDiscoveryPrinter**.

Syntax public ArrayList<DiscoveredPrinter> **GetFoundPrinter**()

Return value Returns the found printer information as ArrayList of **DiscoveredPrinter** class. By method of **DiscoveredPrinter** class, the printer model name, Bluetooth address, IP address, MAC address and serial number can be retrieved.
 For the method of **DiscoveredPrinter** class, see "4.4.2(4) **DiscoveredPrinter**".

(4) **DiscoveredPrinter**

• **Summary**

This class stores the printer information found by **StartDiscoveryPrinter**. From the found printer information, the printer model name, Bluetooth address, IP address, MAC address and serial number can be retrieved.

This class provides the following functions.

Public Methods

Method	Function Summary
GetPrinterModel	Retrieve printer model name
GetBluetoothAddress	Retrieve Bluetooth address
GetIpAddress	Retrieve IP address
GetMacAddress	Retrieve MAC address
GetSerialNumber	Retrieve serial number

• **Public Methods**

GetPrinterModel Retrieve printer model name

Retrieves the character string of printer model name.

Syntax public String **GetPrinterModel()**

Return value Printer model name

GetBluetoothAddress Retrieve Bluetooth address

Retrieves the character string of Bluetooth address.

Syntax public String **GetBluetoothAddress()**

Return value Bluetooth address
Retrieves the character string of "00:11:22:AA:BB:CC" format for the Bluetooth address.

GetIpAddress Retrieve IP address

Retrieves the character string of IP address.

Syntax public String **GetIpAddress()**

Return value IP address

GetMacAddress Retrieve MAC address

Retrieves the character string of MAC address.

Syntax public String **GetMacAddress()**

Return value MAC address

Retrieves the character string of serial number.

Syntax public String **GetSerialNumber()**

Return value Serial number

4.4.3 Enumerated Type (Enum)

(1) **PrinterInterface**

Enumerator used for specifying the communication interface of **Open**.

Name	Description
PRN_IF_TCP	TCP/IP
PRN_IF_BT	Bluetooth
PRN_IF_USB	USB

(2) **PrinterModel**

Enumerator used for specifying the printer model of **Open**.

Name	Description
PRN_MODEL_MP_A40	MP-A40

(3) **Dithering**

Enumerator used for dithering of **ResisterLogo** and **SendDataFile**.

Name	Description
DITHERING_DISABLE	Dithering is disabled
DITHERING_ERRORDIFFUSION	Dithering is enabled

(4) **PrinterInformation**

Enumerator used for specifying the printer information of **GetPrinterInformation**, **GetPrinterInformationNumber** and **GetPrinterInformationString**.

See "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE" for details of printer information to retrieve.

Name	Description (Printer Information)
GET_NV_MEM_CAP	NV graphics memory capacity
GET_NV_MEM_REM_CAP	NV graphics memory remaining capacity
GET_NV_MEM_KEYCODE_LIST	NV graphics key code list
GET_REM_USER_MEM_CAP_DEFRAG	Remaining user area after defragment
GET_REM_USER_MEM_CAP	Remaining user area
GET_FUNC_SET_RESP	Function setting response
GET_PRN_ID_MODEL	Printer ID send (Model ID)
GET_PRN_ID_TYPE	Printer ID send (Type ID)
GET_PRN_ID_ROM_VER	Printer ID send (ROM version ID)
GET_PRN_ID_FIRM_VER_MAIN	Printer ID send (Firmware version (main))
GET_PRN_ID_MFR	Printer ID send (Manufacturer)
GET_PRN_ID_MODEL_NAME	Printer ID send (Model name)

Name	Description (Printer Information)
GET_PRN_ID_FIRM_VER_BOOT	Printer ID send (Firmware version (boot))
GET_PRN_ID_FIRM_CHECKSUM_BOOT	Printer ID send (Firmware checksum (boot))
GET_PRN_ID_FIRM_CHECKSUM_MAIN	Printer ID send (Firmware checksum (main))
GET_PRN_ID_FIRM_CHECKSUM	Printer ID send (Firmware checksum (main+boot))
GET_MAINT_NUM_FEED_LINE	Maintenance counter (Paper feed line count (in 100 dot-lines))
GET_MAINT_NUM_HEAD_ACTIVE	Maintenance counter (Number of thermal head activation times (in 100 dot-lines))
GET_MAINT_DRIVE_TIME	Maintenance counter (Drive time of printer mechanism (in minutes))
GET_MAINT_NUM_FEED_LINE_INTEGRATION	Maintenance counter (Paper feed line count (in 100 dot-lines) (integrated value))
GET_MAINT_NUM_HEAD_ACTIVE_INTEGRATION	Maintenance counter (Number of thermal head activation times (in 100 dot-lines) (integration))
GET_MAINT_DRIVE_TIME_INTEGRATION	Maintenance counter (Drive time of printer mechanism (in minutes) (integration))
GET_HFONT_24_CHECKSUM	1-byte font ID send (24-dot font, checksum)
GET_HFONT_24_ID	1-byte font ID send (24-dot font, ID)
GET_HFONT_24_INT_CHAR	1-byte font ID send (24-dot font, registered international character)
GET_HFONT_16_CHECKSUM	1-byte font ID send (16-dot font, checksum)
GET_HFONT_16_ID	1-byte font ID send (16-dot font, ID)
GET_HFONT_16_INT_CHAR	1-byte font ID send (16-dot font, registered international character)
GET_FFONT_LANG	2-byte font ID send (Language)
GET_FFONT_STANDARD	2-byte font ID send (Standard)
GET_FFONT_COMPANY	2-byte font ID send (Company name)
GET_FFONT_CHECKSUM	2-byte font ID send (Checksum)
GET_FFONT_DATA_SIZE	2-byte font ID send (Data size)

(5) **TransactionFunction**

Enumerator used for specifying the batch process control method of **ControlTransaction**.

Name	Description
TRANSACTION_CLEAR	Cancel batch processing
TRANSACTION_START	Start batch processing
TRANSACTION_PRINT	Finish batch print and batch processing

(6) **Direction**

Enumerator used for specifying the print direction of **SelectPageMode**.

When the start point is "top left" or "bottom right", the printer maps the print data in a vertical direction to the paper feed direction.

When the start point is "top right" or "bottom left", the printer maps the print data in the paper feed direction.

Name	Description ("": Initial Value)
DIRECTION_LEFT_TO_RIGHT	Print direction: left to right, Start point: top left (Figure 4-2 A)
DIRECTION_BOTTOM_TO_TOP	Print direction: bottom to top, Start point: bottom left (Figure 4-2 B)
DIRECTION_RIGHT_TO_LEFT	Print direction: right to left, Start point: bottom right (Figure 4-2 C)
DIRECTION_TOP_TO_BOTTOM	Print direction: top to bottom, Start point: top right (Figure 4-2 D)

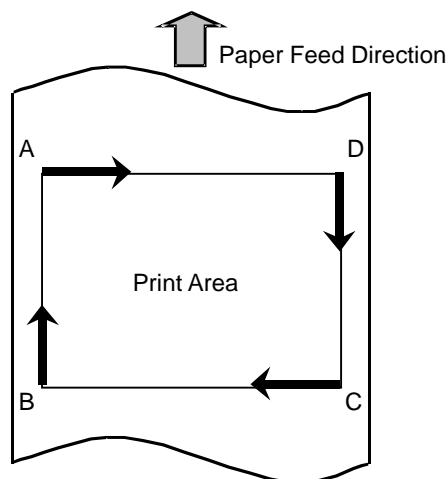


Figure 4-2

(7) **Alignment**

Enumerator used for specifying the print position of **SetStandardModeAlignment**.

Name	Description ("": Initial Value)
ALIGN_LEFT	Left aligned
ALIGN_CENTER	Centered
ALIGN_RIGHT	Right aligned

(8) **CharacterSet**

Enumerator used for specifying the character set of **SelectCharacterSet** and **GetCharacterSet**.
The initial value varies depending on the language setting of WinCE device.

Japanese: **CODEPAGE_KATAKANA**

Other language: **CODEPAGE_1252**

Name	Description ("": Initial Value)
CODEPAGE_437	Code page 437 (USA, Standard Europe)
CODEPAGE_KATAKANA	Katakana character set
CODEPAGE_850	Code page 850 (Multilingual)
CODEPAGE_860	Code page 860 (Portuguese)
CODEPAGE_863	Code page 863 (Canadian-French)
CODEPAGE_865	Code page 865 (Nordic)
CODEPAGE_857	Code page 857 (Turkish)
CODEPAGE_737	Code page 737 (Greek)
CODEPAGE_1252	Code page 1252 (Latin)
CODEPAGE_866	Code page 866 (Russian)
CODEPAGE_852	Code page 852 (Eastern Europe)
CODEPAGE_858	Code page 858 (Euro)
CODEPAGE_855	Code page 855 (Cyrillic)
CODEPAGE_864	Code page 864 (Arabic)
CODEPAGE_1250	Code page 1250 (Central European)
CODEPAGE_1251	Code page 1251 (Cyrillic)
CODEPAGE_1253	Code page 1253 (Greek)
CODEPAGE_1254	Code page 1254 (Turkish)

(9) **InternationalCharacterSet**

Enumerator used for specifying the international character set of **SelectInternationalCharacterSet** and **GetInternationalCharacter**.

The initial value varies depending on the language setting of WinCE device.

Japanese: **INT_CHAR_SET_JAPAN**

Other language: **INT_CHAR_SET_USA**

Name	Description ("": Initial Value)
INT_CHAR_SET_USA	USA
INT_CHAR_SET_FRANCE	France
INT_CHAR_SET_GERMANY	Germany
INT_CHAR_SET_UNITED_KINGDOM	United Kingdom
INT_CHAR_SET_DENMARK_1	Denmark
INT_CHAR_SET_SWEDEN	Sweden
INT_CHAR_SET_ITALY	Italy
INT_CHAR_SET_SPAIN_1	Spain I
INT_CHAR_SET_JAPAN	Japan
INT_CHAR_SET_NORWAY	Norway
INT_CHAR_SET_DENMARK_2	Denmark II
INT_CHAR_SET_SPAIN_2	Spain II
INT_CHAR_SET_LATIN_AMERICA	Latin America
INT_CHAR_SET_ARABIA	Arabia

(10) **CharacterType**

Enumerator used for specifying the character font of **SetCharacterFormatting**.

Name	Description ("": Initial Value)
CHAR_TYPE_FONT_CURRENT ^{*1}	Current setting (No sending of printer command for setting)
CHAR_TYPE_FONT_A	Font A (24×12), Kanji font A (24×24)
CHAR_TYPE_FONT_B ^{*1}	Font B (16×8), Kanji font B (16×16)

*1: When **CODEPAGE_864** is selected in **SelectCharacterSet**, the text is printed in FontA (24×12) regardless of specifying the character font in **SetCharacterFormatting**.

(11) CharacterScale

Enumerator used for specifying the character scale of **SetCharacterFormatting**.

Name	Description ("": Initial Value)
CHAR_SCALE_CURRENT	Current setting (No sending of printer command for setting)
CHAR_SCALE_X1	× 1 (Standard)
CHAR_SCALE_X2	× 2 (double)
CHAR_SCALE_X3	× 3 (triple)
CHAR_SCALE_X4	× 4 (quadruple)
CHAR_SCALE_X5	× 5 (quintuple)
CHAR_SCALE_X6	× 6 (sextuple)
CHAR_SCALE_X7	× 7 (septuple)
CHAR_SCALE_X8	× 8 (octuple)

(12) Underline

Enumerator used for specifying the underline of **SetCharacterFormatting**.

Name	Description ("": Initial Value)
CHAR_UNDERLINE_CURRENT	Current setting (No sending of printer command for setting)
CHAR_UNDERLINE_NONE	No underline
CHAR_UNDERLINE_1DOT	Specify 1-dot width underline
CHAR_UNDERLINE_2DOT	Specify 2-dot width underline

(13) Bold

Enumerator used for specifying the bold print of **SetCharacterFormatting**.

Name	Description ("": Initial Value)
CHAR_BOLD_CURRENT	Current setting (No sending of printer command for setting)
CHAR_BOLD_OFF	No bold print
CHAR_BOLD_ON	Specify bold print

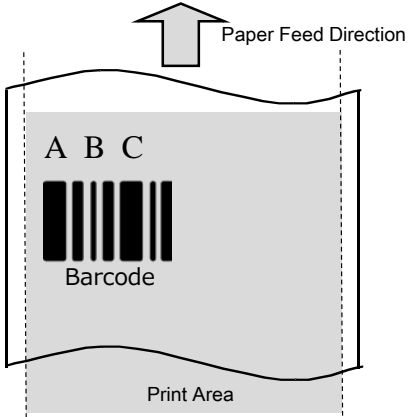
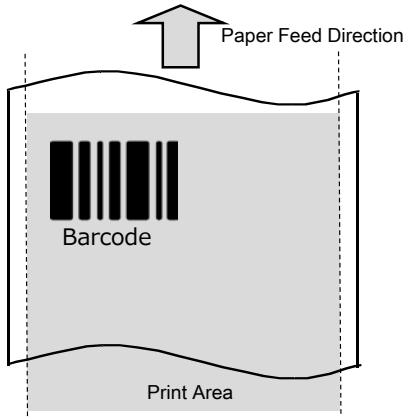
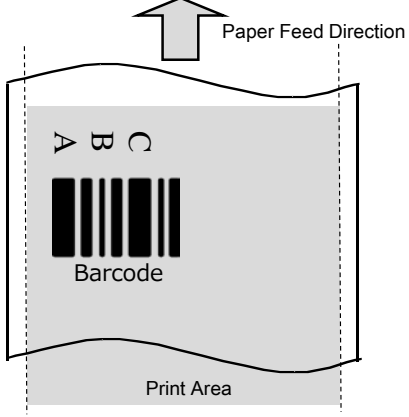
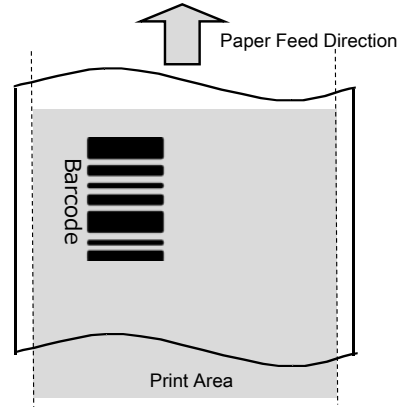
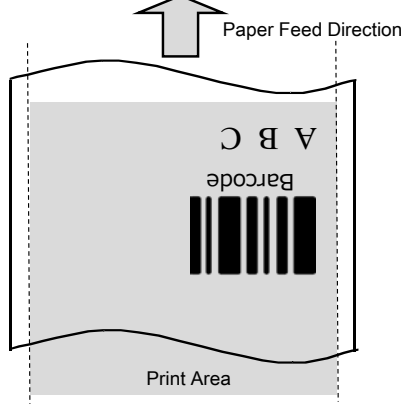
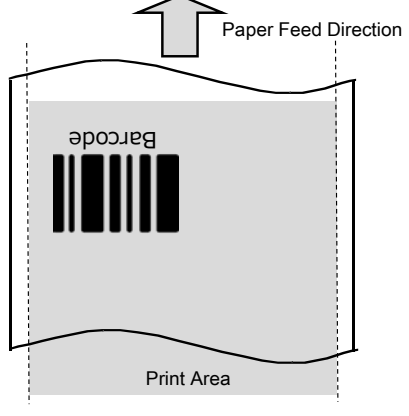
(14) Reverse

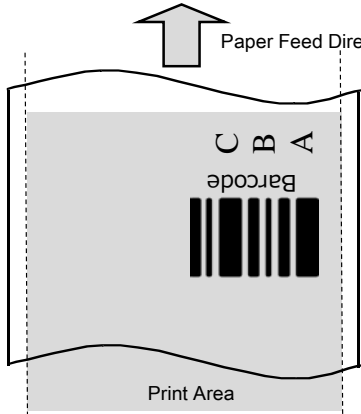
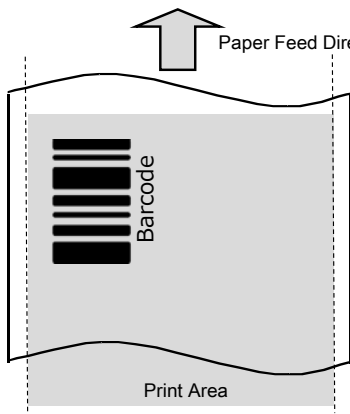
Enumerator used for specifying the reverse print of **SetCharacterFormatting**.

Name	Description ("": Initial Value)
CHAR_REVERSE_CURRENT	Current setting (No sending of printer command for setting)
CHAR_REVERSE_OFF	No reverse print
CHAR_REVERSE_ON	Specify reverse print

(15) Rotate

Enumerator used for specifying the print direction of **SetStandardModeBarcodeDirection**, and for specifying the character rotation print of **SetCharacterFormatting**. The print position and the direction of characters, barcode, and 2-dimensional barcode are shown as follows.

Name	Description (" ": Initial Value)	
	SetCharacterFormatting	SetStandardModeBarcodeDirection
ROTATE_CURRENT	Current setting (No sending of printer command for setting)	
ROTATE_NONE	No rotation 	No rotation 
ROTATE_90_TO_RIGHT		
ROTATE_180		

Name	Description (" " : Initial Value)	
	SetCharacterFormatting	SetStandardModeBarcodeDirection
ROTATE_90_TO_LEFT		

(16) TypeBarcode

Enumerator used for specifying the barcode type of **PrintBarcode**.

Name	Description	
TYPE_BARCODE_UPC_A	UPC-A	Multilevel barcode
TYPE_BARCODE_UPC_E	UPC-E	Multilevel barcode
TYPE_BARCODE_JAN13	JAN13(EAN13)	Multilevel barcode
TYPE_BARCODE_JAN8	JAN8(EAN8)	Multilevel barcode
TYPE_BARCODE_CODE39	CODE39	Binary level barcode
TYPE_BARCODE_ITF	ITF	Binary level barcode
TYPE_BARCODE_CODABAR	CODABAR	Binary level barcode
TYPE_BARCODE_CODE128	CODE128	Multilevel barcode
TYPE_BARCODE_CODE93	CODE93	Multilevel barcode
TYPE_BARCODE_JAN13_ADDON2	JAN13 add-on 2	Multilevel barcode
TYPE_BARCODE_JAN13_ADDON5	JAN13 add-on 5	Multilevel barcode
TYPE_BARCODE_GS1_OMNI_DIRECTIONAL	GS1 Databar Omni-directional	Binary level barcode
TYPE_BARCODE_GS1_TRUNCATED	GS1 Databar Truncated	Binary level barcode
TYPE_BARCODE_GS1_LIMITED	GS1 Databar Limited	Binary level barcode
TYPE_BARCODE_GS1_EXPANDED	GS1 Databar Expanded	Binary level barcode

(17) **ModuleWidthBarcode**

Enumerator used for specifying the module width in multilevel barcode or narrow element in binary level barcode of the barcode of **PrintBarcode**.

Name	Description
MODULE_WIDTH_BARCODE_2	2 dots
MODULE_WIDTH_BARCODE_3	3 dots
MODULE_WIDTH_BARCODE_4	4 dots
MODULE_WIDTH_BARCODE_5	5 dots
MODULE_WIDTH_BARCODE_6	6 dots

(18) **HriPositionBarcode**

Enumerator used for specifying the barcode HRI characters of **PrintBarcode**.

Name	Description
HRI_BARCODE_NONE	No HRI character
HRI_BARCODE_TOP_FONT_A	Above barcode (font A)
HRI_BARCODE_BOTTOM_FONT_A	Below barcode (font A)
HRI_BARCODE_FONT_A	Above and below barcode (font A)
HRI_BARCODE_TOP_FONT_B	Above barcode (font B)
HRI_BARCODE_BOTTOM_FONT_B	Below barcode (font B)
HRI_BARCODE_FONT_B	Above and below barcode (font B)

(19) **NwRatioBarcode**

Enumerator used for specifying the N:W ratio of the binary level barcode of **PrintBarcode**.

Name	Description
NWRATIO_BARCODE_1TO2	1:2
NWRATIO_BARCODE_1TO2_5	1:2.5
NWRATIO_BARCODE_1TO3	1:3

(20) **Type2Dcode**

Enumerator used for specifying the 2-dimensional barcode type of **Print2Dcode**.

Name	Description
TYPE_2DCODE_QR_CODE	QR Code
TYPE_2DCODE_PDF417	PDF417
TYPE_2DCODE_DATA_MATRIX	Data Matrix
TYPE_2DCODE_MAXI_CODE	MaxiCode
TYPE_2DCODE_GS1_STACKED	GS1 Databar Stacked
TYPE_2DCODE_GS1_OMNI_DIRECTIONAL	GS1 Databar Stacked Omni-directional
TYPE_2DCODE_GS1_EXPANDED_STACKED	GS1 Databar Expanded Stacked

(21) **Mode2Dcode**

Enumerator used for specifying the 2-dimensional barcode mode of **Print2Dcode**.

Name	Description	
MODE_2DCODE_QR_CODE_MODEL1	QR Code	Model1
MODE_2DCODE_QR_CODE_MODEL2	QR Code	Model2
MODE_2DCODE_PDF417_STANDARD	PDF417	Normal Mode
MODE_2DCODE_PDF417_COMPACT	PDF417	Simple Mode
MODE_2DCODE_DATA_MATRIX_AUTO	Data Matrix	Module numbers: Automatic
MODE_2DCODE_DATA_MATRIX_10_10	Data Matrix	Module numbers: 10×10
MODE_2DCODE_DATA_MATRIX_12_12	Data Matrix	Module numbers: 12×12
MODE_2DCODE_DATA_MATRIX_14_14	Data Matrix	Module numbers: 14×14
MODE_2DCODE_DATA_MATRIX_16_16	Data Matrix	Module numbers: 16×16
MODE_2DCODE_DATA_MATRIX_18_18	Data Matrix	Module numbers: 18×18
MODE_2DCODE_DATA_MATRIX_20_20	Data Matrix	Module numbers: 20×20
MODE_2DCODE_DATA_MATRIX_22_22	Data Matrix	Module numbers: 22×22
MODE_2DCODE_DATA_MATRIX_24_24	Data Matrix	Module numbers: 24×24
MODE_2DCODE_DATA_MATRIX_26_26	Data Matrix	Module numbers: 26×26
MODE_2DCODE_DATA_MATRIX_32_32	Data Matrix	Module numbers: 32×32
MODE_2DCODE_DATA_MATRIX_36_36	Data Matrix	Module numbers: 36×36
MODE_2DCODE_DATA_MATRIX_40_40	Data Matrix	Module numbers: 40×40
MODE_2DCODE_DATA_MATRIX_44_44	Data Matrix	Module numbers: 44×44
MODE_2DCODE_DATA_MATRIX_48_48	Data Matrix	Module numbers: 48×48
MODE_2DCODE_DATA_MATRIX_52_52	Data Matrix	Module numbers: 52×52
MODE_2DCODE_DATA_MATRIX_64_64	Data Matrix	Module numbers: 64×64
MODE_2DCODE_DATA_MATRIX_72_72	Data Matrix	Module numbers: 72×72
MODE_2DCODE_DATA_MATRIX_80_80	Data Matrix	Module numbers: 80×80
MODE_2DCODE_DATA_MATRIX_88_88	Data Matrix	Module numbers: 88×88
MODE_2DCODE_DATA_MATRIX_96_96	Data Matrix	Module numbers: 96×96
MODE_2DCODE_DATA_MATRIX_104_104	Data Matrix	Module numbers: 104×104
MODE_2DCODE_DATA_MATRIX_120_120	Data Matrix	Module numbers: 120×120
MODE_2DCODE_DATA_MATRIX_132_132	Data Matrix	Module numbers: 132×132
MODE_2DCODE_DATA_MATRIX_144_144	Data Matrix	Module numbers: 144×144
MODE_2DCODE_DATA_MATRIX_8_18	Data Matrix	Module numbers: 8×18
MODE_2DCODE_DATA_MATRIX_8_32	Data Matrix	Module numbers: 8×32
MODE_2DCODE_DATA_MATRIX_12_26	Data Matrix	Module numbers: 12×26
MODE_2DCODE_DATA_MATRIX_12_36	Data Matrix	Module numbers: 12×36
MODE_2DCODE_DATA_MATRIX_16_36	Data Matrix	Module numbers: 16×36
MODE_2DCODE_DATA_MATRIX_16_48	Data Matrix	Module numbers: 16×48
MODE_2DCODE_MAXI_CODE_2	MaxiCode	Mode2
MODE_2DCODE_MAXI_CODE_3	MaxiCode	Mode3
MODE_2DCODE_MAXI_CODE_4	MaxiCode	Mode4
MODE_2DCODE_MAXI_CODE_5	MaxiCode	Mode5

Name	Description	
MODE_2DCODE_NONE	GS1 Databar Stacked GS1 Databar Stacked Omni-directional GS1 Databar Expanded Stacked	No settings

(22) **ModuleSize2DCode**

Enumerator used for specifying the 2-dimensional barcode module size of **Print2Dcode**.

Name	Description	
MODULE_SIZE_2DCODE_DEFAULT	QR Code	Default (6 dots)
	PDF417	Default (4 dots)
	Data Matrix	Default (6 dots)
	MaxiCode	No settings
	GS1 Databar Stacked	Default (6 dots)
	GS1 Databar Stacked Omni-directional	Default (6 dots)
	GS1 Databar Expanded Stacked	Default (6 dots)
MODULE_SIZE_2DCODE_2	QR Code	2 dots
	PDF417	
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_3	QR Code	3 dots
	PDF417	
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_4	QR Code	4 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_5	QR Code	5 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	

Name	Description	
MODULE_SIZE_2DCODE_6	QR Code	6 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_7	QR Code	7 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_8	QR Code	8 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_9	QR Code	9 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_10	QR Code	10 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_11	QR Code	11 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_12	QR Code	12 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_13	QR Code	13 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	

Name	Description	
MODULE_SIZE_2DCODE_14	QR Code	14 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_15	QR Code	15 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_16	QR Code	16 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	

(23) **ErrorCorrect2DCode**

Enumerator used for specifying the 2-dimensional barcode error correction level of **Print2Dcode**.

Name	Description	
ERR_CORRECTION_2DCODE_QR_CODE_L	QR Code	L
ERR_CORRECTION_2DCODE_QR_CODE_M	QR Code	M
ERR_CORRECTION_2DCODE_QR_CODE_Q	QR Code	Q
ERR_CORRECTION_2DCODE_QR_CODE_H	QR Code	H
ERR_CORRECTION_2DCODE_PDF417_0	PDF417	0
ERR_CORRECTION_2DCODE_PDF417_1	PDF417	1
ERR_CORRECTION_2DCODE_PDF417_2	PDF417	2
ERR_CORRECTION_2DCODE_PDF417_3	PDF417	3
ERR_CORRECTION_2DCODE_PDF417_4	PDF417	4
ERR_CORRECTION_2DCODE_PDF417_5	PDF417	5
ERR_CORRECTION_2DCODE_PDF417_6	PDF417	6
ERR_CORRECTION_2DCODE_PDF417_7	PDF417	7
ERR_CORRECTION_2DCODE_PDF417_8	PDF417	8
ERR_CORRECTION_2DCODE_NONE	Data Matrix MaxiCode GS1 Databar Stacked GS1 Databar Stacked Omni-directional GS1 Databar Expanded Stacked	No settings

(24) **LogFileSize**

Enumerator used for specifying the maximum size of the log file of **SetLog**.

Name	Description ("": Initial Value)
LOG_FILE_SIZE_1MB	1 MB (1048576 bytes)
LOG_FILE_SIZE_5MB	5 MB (5242880 bytes)
LOG_FILE_SIZE_10MB	10 MB (10485760 bytes)
LOG_FILE_SIZE_50MB	50 MB (52428800 bytes)

(25) **ErrorCode**

Enumerator indicating the error content retrieved by **GetErrorCode**.

Name	Description
ERR_PARAM	Parameter is incorrect.
ERR_OPENED	Specified printer has already been opened.
ERR_NOT_OPENED	Specified printer is not opened.
ERR_TIMEOUT	Timeout or busy state occurs.
ERR_OFFLINE	Printer is disconnected or offline.
ERR_CLOSE_FAIL	Failed to disconnect printer.
ERR_NOT_MONITORING	Monitoring of connecting status is not performed.
ERR_INIT_FAILED	Failed to initialize.
ERR_DATA_SIZE_ZERO	0 byte size data is specified.
ERR_OVER_MAX_DATA_SIZE	Maximum data size is exceeded.
ERR_INVALID_DATA	Invalid data is specified.
ERR_INVALID_STATE	Access to PrinterManager object was specified for callback.
ERR_ACCESS	Printer cannot be accessed.
ERR_CANCELED	Function has been canceled.
ERR_WRITE_FAULT	Data cannot be sent to printer.
ERR_WORKAREA_NO_MEMORY	Specified memory size is insufficient.
ERR_FILE_INVALID	Specified file is invalid.
ERR_ENCODE_FAILED	Error has occurred in encoding text data.
ERR_NOT_FOUND	Specified file cannot be found.
ERR_TRANSACTION_STOPPED	Batch processing is not started.
ERR_PRINTER_STATUS_ERROR	Printer status is abnormal.

4.4.4 Exception

(1) PrinterException

- **Summary**

This class indicates an exceptional situation has occurred.
This class provides the following function.

Public Methods

Method	Function Summary
GetErrorCode	Retrieve error code

- **Public Methods**

GetErrorCode	Retrieve error codes
---------------------	----------------------

Retrieves error code for thrown exception.

Syntax public Errorcode **GetErrorCode()**

Return value Error code

Description See "4.4.3(25) **ErrorCode**" for details on the error.

Chapter 5

Sample Program

This chapter describes the sample program provided by the SDK.

5.1 Sample Program Overview

The SDK provides a CAB file and a project format program of Microsoft Visual Studio C# 2008 as the sample program.

5.2 How to Use Sample Program

(1) How to use CAB file

- (a) Copy the CAB file located under [CD-ROM Drive]:\WindowsCE\Sample\cab folder to an appropriate folder on the WinCE device.
- (b) Execute the CAB file from File Explorer.
- (c) Execute sample programs from [Program] on WinCE device.

(2) How to use project format program

Ensure that the environment for developing WinCE application is prepared. See "Chapter 3 How to Use the Library" for details about required development environment.

- (a) Copy [CD-ROM Drive]:\WindowsCE\Sample\source folder to local computer.
- (b) Execute the SLN file in the copied source folder from File Explorer.

5.3 Outline of Sample Program UI display

This section describes the outline of UI display of sample program installed in "5.2(1) How to use CAB file".



Item	Description
Connection Type	Selects the connection type to a printer.
Printer Model	Selects the printer model.
Address	Specifies the printer address. For manual input: Input the Bluetooth address of the printer. For automatic input: By tapping [List], searching a printer is started. By selecting a printer from the displayed list, the address of the selected printer is automatically input.
Various Functions	These are buttons that can execute various functions. By scrolling the screen, you can find the methods and properties not displayed in the screen. See "Chapter 4 Function of the Library" for details of each method and property.

5.4 Precaution

The sample program is subject to change without notice.

No guarantee of proper operation and support are provided for the sample program.

Chapter 6

Disclaimer

We closely monitor the development of this software in order to avoid problems. However, we are not responsible for any damages arising out of the use of this software.

Appendix A

Character Sets (Character Code Table)

A.1 Character Code Table

The codepages when **INT_CHAR_SET_USA** is set for the international character set are shown below. Print results of the specific character codes vary depending on the setting of the international character set. See "A.2 International Character Set" for the specific character codes.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	¢	£	¥	℔	ƒ
A0	á	í	ó	ú	ñ	Ñ	ª	º	¿	¬	½	¼	¿	«	»	
B0	☐	☐	☐		†	‡	§	¶	§	¶	§	¶	§	¶	§	¶
C0	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ
D0	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ
E0	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	Π
F0	≡	±	≥	≤	∫	∫	÷	≈	°	•	•	√	n	2	■	

Figure A-1 CODEPAGE_437 (USA, Standard Europe)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80																
90																
A0	。	「	」	、	・	ヲ	ア	イ	ウ	エ	オ	ヤ	ユ	ヨ	ッ	
B0	ー	ア	イ	ウ	エ	オ	カ	キ	ク	ケ	コ	サ	シ	ス	セ	ソ
C0	タ	チ	ツ	テ	ト	ナ	ニ	ヌ	ネ	ノ	ハ	ヒ	フ	ヘ	ホ	マ
D0	ミ	ム	メ	モ	ヤ	ユ	ヨ	ラ	リ	ル	レ	ロ	ワ	ン	ゝ	。
E0																
F0																

Figure A-2 CODEPAGE_KATAKANA

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Ç	ü	é	â	ä	à	â	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	×	ƒ
A0	á	í	ó	ú	ñ	Ñ	ä	ö	¿	®	¬	½	¼	¡	«	»
B0	☐	☐	☐			Á	Â	À	©	¶	¶	¶	¶	¢	¥	₱
C0	⊥	⊥	⊥	⊥	⊥	ã	Ã	ℓ	ℓ	ℓ	ℓ	ℓ	ℓ	=	ℓ	α
D0	ð	Ð	Ê	Ë	È	Í	Î	Ï	⌋	⌋	■	■	■	■	■	■
E0	Ó	β	Ô	Ò	Õ	μ	þ	þ	Ú	Û	Ü	ý	Ý	-	-	-
F0	-	±	=	¾	¶	§	÷	,	°	…	.	¹	³	²	■	■

Figure A-3 CODEPAGE_850 (Multilingual)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Ç	ü	é	â	ã	à	Á	ç	ê	Ê	è	Í	Ô	ì	Ã	Â
90	É	À	È	ô	õ	ò	Ú	ù	Ï	Ö	Ü	¢	£	Ù	Þ	Ó
A0	á	í	ó	ú	ñ	Ñ	ä	ö	ï	ò	¬	½	¼	¡	«	»
B0	☐	☐	☐													
C0	L	L	T	T	T	T	T	T	T	T	T	T	T	T	T	T
D0	L	T	T	L	L	F	T	T	T	T	T	T	T	T	T	T
E0	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	Π
F0	≡	±	≥	≤		J	÷	≈	°	•	•	√	n	2	■	

Figure A-4 CODEPAGE_860 (Portuguese)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Ç	ü	é	â	Â	à	¶	ç	ê	ë	è	ï	î	≡	À	§
90	É	È	Ê	ô	Ë	Ï	Ô	Ù	¤	Ô	Ü	¢	£	Ù	û	f
A0		'	ó	ú	..	³	-	î	¬	¬	½	¼	¾	«	»	
B0	☐	☐	☐													
C0	L	L	T	T	T	T	T	T	T	T	T	T	T	T	T	T
D0	L	T	T	L	L	F	T	T	T	T	T	T	T	T	T	T
E0	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	Π
F0	≡	±	≥	≤		J	÷	≈	°	•	•	√	n	2	■	

Figure A-5 CODEPAGE_863 (Canadian-French)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ü	ÿ	Ö	Ü	ø	£	Ø	Pt	f
A0	á	í	ó	ú	ñ	Ñ	ä	ö	¿	¬	½	¼	¡	«	»	
B0	☐	☐	☐													
C0	L	L	T	T	T	T	T	T	T	T	T	T	T	T	T	T
D0	L	T	T	L	L	F	π	π	π	π	π	π	π	π	π	π
E0	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	Π
F0	≡	±	≥	≤		J	÷	≈	°	•	•	√	n	2	■	

Figure A-6 CODEPAGE_865 (Nordic)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ü	ÿ	Ö	Ü	ø	£	Ø	Ş	ş
A0	á	í	ó	ú	ñ	Ñ	Ğ	ğ	¿	®	¬	½	¼	¡	«	»
B0	☐	☐	☐			Á	Â	À	©							
C0	L	L	T	T	T	ã	Ã	L	π	π	π	π	π	π	π	π
D0	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	Π
E0	ó	β	ô	ò	õ	Ö	μ	×	ú	û	ü	ì	ÿ	-	-	-
F0	-	±	¾	¶	§	÷	,	°	°	°	°	1	3	2	■	

Figure A-7 CODEPAGE_857 (Turkish)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	A	B	Γ	Δ	E	Z	H	Θ	I	K	Λ	M	N	Ξ	O	Π
90	P	Σ	T	Υ	Φ	X	Ψ	Ω	α	β	γ	δ	ε	ζ	η	θ
A0	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	ς	τ	υ	φ	χ	ψ
B0	⌘	⌘	⌘		†	‡	§	¶	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
C0	L	⌘	T	†	†	†	†	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
D0	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
E0	ω	ά	έ	ή	ϊ	ί	ό	ύ	ϋ	ώ	À	É	Η	Ι	Ο	Υ
F0	Ω	±	≥	≤	İ	ÿ	÷	≈	°	•	•	√	n	2	■	

Figure A-8 CODEPAGE_737 (Greek)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	€	‚	ƒ	„	…	†	‡	^	‰	Š	‹	Œ		Ž		
90	‘	’	“	”	•	-	-	~	™	š	›	œ		ž	ÿ	
A0	ı	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯	
B0	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Figure A-9 CODEPAGE_1252 (Latin)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
90	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
A0	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
B0	␣	␣	␣													
C0	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣
D0	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣
E0	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
F0	Ё	ё	Є	є	İ	ı	Ÿ	ÿ	°	•	•	√	№	α	■	

Figure A-10 CODEPAGE_866 (Russian)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Ç	ü	é	â	ä	û	ç	ł	ë	ő	ö	î	ž	Ä	Ć	
90	É	Í	í	ô	ö	Ĺ	ĺ	Š	š	Ö	Ü	ř	ť	Ł	×	č
A0	á	í	ó	ú	À	à	Ž	ž	Ę	ę	¬	ž	Č	š	«	»
B0	␣	␣	␣			Á	Â	Ě	Š				Ž	ž		
C0	␣	␣	␣	␣	␣	Ä	ä	␣	␣	␣	␣	␣	␣	␣	␣	␣
D0	đ	Đ	Ď	Ě	ď	Ň	í	î	ě	␣	␣	␣	␣	␣	␣	␣
E0	ó	ß	ô	ń	ň	š	š	ř	ú	ř	Ů	ý	Ý	ť	´	
F0	-	"	˘	˘	˘	Š	÷	˘	˘	˘	Ů	Ř	ř	■		

Figure A-11 CODEPAGE_852 (Eastern Europe)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Ç	ü	é	â	ä	à	â	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ü	ö	Ü	ø	£	Ø	×	f	
A0	á	í	ó	ú	ñ	Ñ	ä	ö	¿	®	¬	½	¼	¡	«	»
B0	☐	☐	☐			Á	Â	À	©	¶		¶	¶	¢	¥	₱
C0	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
D0	ð	Ð	Ê	Ë	È	€	Í	Î	Ï	⌋	⌋	■	■	■	■	■
E0	Ó	β	Ô	Ò	Õ	μ	þ	þ	Ú	Û	Ü	ý	Ý	-	'	
F0	-	±	=	¾	¶	§	÷	,	°	..	.	1	3	2	■	

Figure A-12 CODEPAGE_858 (Euro)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	ђ	Ђ	ѓ	Ѓ	ё	Ё	є	Є	ѕ	Ѕ	і	І	ї	Ї	ј	Ј
90	љ	Љ	њ	Њ	ћ	Ћ	ќ	Ќ	џ	Џ	џ	џ	џ	џ	џ	џ
A0	а	А	б	Б	в	В	г	Г	д	Д	е	Е	ф	Ф	г	Г
B0	☐	☐	☐			x	X	и	И	¶		¶	¶	й	Й	₱
C0	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
D0	л	Л	м	М	н	Н	о	О	п	П	■	■	■	■	■	■
E0	я	Я	р	Р	с	С	т	Т	у	У	ж	Ж	в	В	ь	№
F0	-	ы	Ы	э	Э	ш	Ш	э	Э	щ	Щ	ч	Ч	§	■	

Figure A-13 CODEPAGE_855 (Cyrillic)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	°	•	•	√	■	-		+	+	+	+	+	+	+	+	+
90	β	∞	φ	±	½	¼	≈	«	»	لأ	لأ					لا
A0	-	£	£	£	£	£	£	ل	ب	ث	ج	ح	خ	ح	خ	خ
B0	•	١	٢	٣	٤	٥	٦	٧	٨	٩	ف	س	ش	ص	ش	؟
C0	¢	ء	آ	أ	ؤ	ع	ئ	ب	ة	ث	ج	ح	خ	ح	خ	خ
D0	ذ	ر	ز	س	ش	ص	ض	ط	ظ	ع	غ	ف	ق	ك	خ	ع
E0	-	ف	ق	ك	م	ل	ه	و	ي	ض	ع	غ	ف	ق	ك	م
F0	-	"	"	ن	ه	ه	ي	ي	ق	ق	ل	ل	ل	ل	ل	■

Figure A-14 CODEPAGE_864 (Arabic)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	€	;	;	...	†	†	™	š	<	š	š	ž	ž	ž	ž	ž
90	€	;	;	...	•	-	-	™	š	>	š	š	ž	ž	ž	ž
A0	˘	˘	Ł	Ł	Ł	Ł	Ł	Ł	Ł	Ł	Ł	Ł	Ł	Ł	Ł	Ł
B0	°	±	±	±	±	±	±	±	±	±	±	±	±	±	±	±
C0	Á	Á	Á	Á	Á	Á	Á	Á	Á	Á	Á	Á	Á	Á	Á	Á
D0	Đ	Đ	Đ	Đ	Đ	Đ	Đ	Đ	Đ	Đ	Đ	Đ	Đ	Đ	Đ	Đ
E0	á	á	á	á	á	á	á	á	á	á	á	á	á	á	á	á
F0	đ	đ	đ	đ	đ	đ	đ	đ	đ	đ	đ	đ	đ	đ	đ	đ

Figure A-15 CODEPAGE_1250 (Central European)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	ђ	ѓ	џ	ѓ	„	…	†	‡	€	‰	љ	ќ	ћ	џ		
90	ђ	ѓ	џ	ѓ	„	…	†	‡	€	‰	љ	ќ	ћ	џ		
A0	Ў	ў	Ј	ѡ	Г	І	Ѕ	Ё	Є	«	¬	–	®	İ		
B0	°	±	І	і	г	μ	¶	•	ё	№	є	»	ј	ѕ	ї	
C0	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
D0	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
E0	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
F0	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я

Figure A-16 CODEPAGE_1251 (Cyrillic)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	€	‘	’	“	”	…	†	‡	‰	‹						
90	€	‘	’	“	”	…	†	‡	‰	›						
A0	“	À	£	¤	¥	¦	§	¨	©	ª	«	¬	–	®	–	
B0	°	±	²	³	´	μ	¶	·	¸	¹	º	»	¼	½	¾	
C0	ı	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
D0	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	İ	ÿ	ά	έ	ή	ί	
E0	ΐ	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
F0	π	ρ	ς	σ	τ	υ	φ	χ	ψ	ω	ï	ÿ	ό	ύ	ώ	

Figure A-17 CODEPAGE_1253 (Greek)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	€	‘	‚	“	”	…	†	‡	^	‰	Š	<	£			
90		‚	‚	“	”	•	-	-	~	™	š	>	œ		ÿ	
A0	ı	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	®	¯		
B0	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ğ	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	İ	Ş	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ğ	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ı	ş	ÿ

Figure A-18 CODEPAGE_1254 (Turkish)

A.2 International Character Set

Print results of the specific character codes vary depending on the setting of the international character set. The following table shows the specific character codes and their print results.

	23	24	40	5B	5C	5D	5E	60	7B	7C	7D	7E
COUNTRY_USA	#	\$	@	[\]	^	`	{		}	~
COUNTRY_FRANCE	#	\$	à	°	ç	§	^	`	é	ù	è	..
COUNTRY_GERMANY	#	\$	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß
COUNTRY_ENGLAND	£	\$	@	[\]	^	`	{		}	~
COUNTRY_DENMARK_1	#	\$	@	Æ	Ø	Å	^	`	æ	ø	å	~
COUNTRY_SWEDEN	#	α	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
COUNTRY_ITALY	#	\$	@	°	\	é	^	ù	à	ò	è	ì
COUNTRY_SPAIN	£	\$	@	ì	Ñ	¿	^	`	..	ñ	}	~
COUNTRY_JAPAN	#	\$	@	[¥]	^	`	{		}	~
COUNTRY_NORWAY	#	α	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü
COUNTRY_DENMARK_2	#	\$	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü
COUNTRY_SPAIN_2	#	\$	á	ì	Ñ	¿	é	`	í	ñ	ó	ú
COUNTRY_LATIN_AMERICA	#	\$	á	ì	Ñ	¿	é	ü	í	ñ	ó	ú
COUNTRY_ARABIA	#	\$	@	[\]	^	`	{		}	~

Figure A-19 International Character Set

SII



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

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

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

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

(Specifications are subject to change without notice.)