



SII SDK for Windows CE Application Programmer's Guide

Rev.03

[Products]

MP-B20 Series

Seiko Instruments Inc.

SII SDK for Windows CE Application Programmer's Guide

| | |
|--------|----------------|
| Rev.01 | September 2018 |
| Rev.02 | February 2019 |
| Rev.03 | December 2019 |


Copyright © 2018-2019 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 manual describes "SII SDK for Windows CE" (hereinafter referred to as the "SDK") provided by Seiko Instruments Inc. (hereinafter referred to as "SII").

Notation in This Manual

The notation in this manual is described.

Terms

The terms used in this manual are defined as below.

| Term | Description |
|-----------------|---|
| Printer command | Instruction to control the printer, described in "MP-B20 SERIES THERMAL PRINTER TECHNICAL REFERENCE". |

Symbols

The symbols used in this manual are described below.

Caution

- ◆ Notes and limitations are described.

Reference

- Supplemental information and related matters are described.

Table of Contents

| | | |
|-------------------|---|------------|
| Chapter 1 | Overview | 1-1 |
| 1.1 | Function | 1-1 |
| 1.2 | Configuration | 1-1 |
| 1.2.1 | Configuration Diagram | 1-2 |
| 1.2.2 | Provided Functions..... | 1-2 |
| 1.2.3 | Sample Program | 1-2 |
| 1.3 | Operating Environment..... | 1-2 |
| 1.3.1 | System Environment | 1-2 |
| 1.3.2 | Target Products..... | 1-3 |
| 1.4 | Printer Settings | 1-3 |
| Chapter 2 | How to Use the Library | 2-1 |
| 2.1 | Provided Files | 2-1 |
| 2.2 | Build the Library into Microsoft Visual Studio Projects..... | 2-2 |
| 2.3 | Execution Condition on WinCE Device..... | 2-2 |
| Chapter 3 | Function of the Library | 3-1 |
| 3.1 | Data Send/Receive Processing of the Library and the Limitations | 3-1 |
| 3.2 | API Reference | 3-2 |
| 3.2.1 | Interface | 3-4 |
| 3.2.2 | Class | 3-6 |
| 3.2.3 | Enumerated Type (Enum) | 3-42 |
| 3.2.4 | Exception..... | 3-55 |
| Chapter 4 | Sample Program | 4-1 |
| 4.1 | How to Use | 4-1 |
| 4.1.1 | CAB file..... | 4-1 |
| 4.1.2 | Project Format Program | 4-1 |
| 4.2 | Screen Layout..... | 4-2 |
| 4.3 | Precaution..... | 4-3 |
| Appendix A | Character Sets (Character Code Table) | A-1 |
| A.1 | Character Code Table | A-1 |
| A.2 | International Character Set..... | A-11 |

Chapter 1

Overview

This chapter describes the product overview of the SDK.

1.1 Function

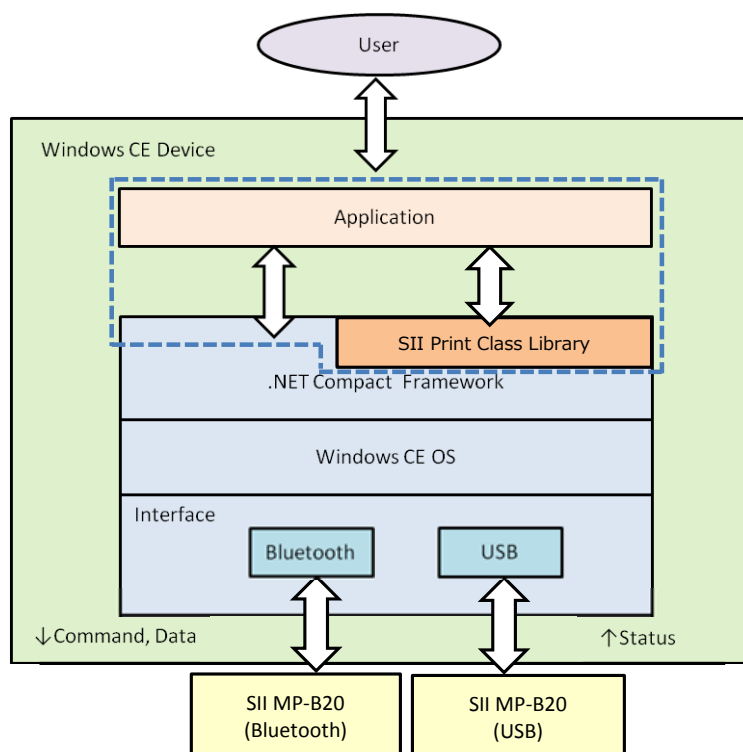
The SII print class library included in the SDK provides Windows Embedded CE (hereinafter referred to "WinCE") applications in Microsoft .NET Compact Framework (hereinafter referred to as ".NET Compact Framework") environment with the functions to use SII printer MP-B20 Series (hereinafter referred to as "printer").

Moreover, the SDK provides Microsoft Visual Studio projects as a sample program for SII print class library (hereinafter referred to as "the library").

1.2 Configuration

1.2.1 Configuration Diagram

The library and the sample program included in the SDK are indicated with dashed lines in the figure below.



1.2.2 Provided Functions

By using the library, WinCE applications can easily send print data and printer commands to printer through communication port (Bluetooth or USB) on an WinCE device. Also, the applications can get printer status.

The library provides the following functions.

- Connecting to / disconnecting from a printer
- Sending data to a printer (print data and/or printer commands)
- Printing barcode and 2-dimensional barcode
- Sending a data file to a printer (print data and/or printer commands)
- Getting the printer status
- Getting various responses from a printer
- Searching the printer by Bluetooth

1.2.3 Sample Program

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

1.3 Operating Environment

1.3.1 System Environment

The system environment for the library is shown in the following table.

| Item | Specifications |
|--------------------------------|--|
| 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) |
| .NET Framework | .NET Compact Framework 3.5 ^{*1} or later |
| Supported Language | Japanese English |
| Supported Development Language | Able to use .NET Compact Framework Microsoft C# Microsoft Visual Basic |

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

1.3.2 Target Products

| Printer | Communication Interface |
|---------------|-------------------------|
| MP-B20 Series | USB ^{*1} |
| | Bluetooth ^{*2} |

- *1: 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.
The USB Printer class driver can be handled as an LPT port on the target OS of the library.
- *2: Bluetooth connection needs to be established by SPP (Serial Port Profile).
For the driver related to Bluetooth, use the Microsoft system standard driver.

1.4 Printer Settings

For using the library, set the printer's memory switch to the [Value] shown in the following table.

The memory switch can be changed by the "SII Printer Driver for Windows" for MP-B20 Series.
To change the memory switch, see "5.3 Memory Switch" in "SII Printer Driver for Windows User's Guide" for MP-B20 Series.

- **For USB connection**

| MS | Item | Value |
|-----|---|-----------|
| 1-1 | Interface | 0: USB |
| 3-1 | Automatic Status Response Selection (Auto Status Back) | 0: Enable |
| 3-2 | Initialized Response Selection (Init Response) | 0: Enable |
| 3-3 | Realtime Command Selection (Realtime Command) | 0: Enable |

- **For Bluetooth connection**

| MS | Item | Value |
|-----|---|-------------|
| 1-1 | Interface | 1: Wireless |
| 3-1 | Automatic Status Response Selection (Auto Status Back) | 0: Enable |
| 3-2 | Initialized Response Selection (Init Response) | 0: Enable |
| 3-3 | Realtime Command Selection (Realtime Command) | 0: Enable |

Chapter 2

How to Use the Library

This chapter describes how to use the library.

2.1 Provided Files

The SDK provides DLL file for the library.
The folder configuration of the SDK is shown in the following.

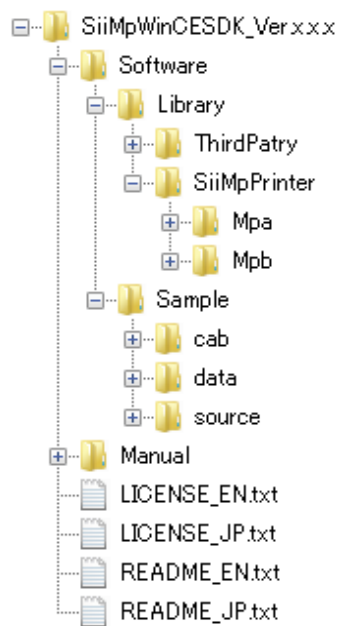


Figure 2-1

2.2 Build the Library into Microsoft Visual Studio Projects

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

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

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

By completing these procedures, functions of the library become available.

2.3 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

Reference

- 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 3

Function of the Library

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

3.1 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 before reconnecting.

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

When the buffered receive data exceeds 4096 bytes, the old data in the 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 "3.2 API Reference".

For the details of the various responses, see "MP-B20 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

3.2 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 to get the event of the printer status change. |
| DiscoveryHandler | Interface to get the event in 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 . |
| Alignment | Enumerator used for specifying the alignment of SetStandardModeAlignment . |
| CharacterSet | Enumerator used for selecting the character set of SelectCharacterSet and GetCharacterSet . |
| InternationalCharacterSet | Enumerator used for selecting 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 . |
| Underline | Enumerator used for specifying the underline of SetCharacterFormatting . |
| Bold | Enumerator used for specifying the bold print of SetCharacterFormatting . |

| Class Name | Description |
|---------------------------|---|
| Reverse | Enumerator used for specifying the reverse print of SetCharacterFormatting . |
| Rotate | Enumerator used for specifying the character rotation print of SetCharacterFormatting , and for specifying the barcode print 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 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. |

3.2.1 Interface

(1) **StatusCallbackHandler**

- **Summary**

This interface is for getting 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** class

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 "3.2.2(2) **PrinterStatus**".

(2) **DiscoveryHandler**

- **Summary**

This interface is for getting the event in printer searching.
This interface provides the following functions.

Public Methods

| Method | Function Summary |
|---------------------------|-----------------------------------|
| DiscoveryFinished | Finalization of searching printer |
| DiscoveryCancelled | Cancellation of searching printer |

- **Public Methods**

| | |
|--------------------------|--|
| DiscoveryFinished | Finalization of searching printer |
|--------------------------|--|

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 searching printer |
|---------------------------|--|

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.

3.2.2 Class

(1) PrinterManager

- **Summary**

This class provides the API used for printing on the printer and for getting the printer information. This class provides the following functions.

Public Methods

| Method | Function Summary |
|------------------------------------|---|
| PrinterManager | Constructor |
| Open | Start using printer |
| Close | Finish using printer |
| SetWriteTimeout | Set send timeout period |
| GetWriteTimeout | Get send timeout period |
| SetResponseTimeout | Set receive timeout period |
| GetResponseTimeout | Get receive timeout period |
| Write | Send binary data |
| Read | Get receive data |
| GetReadSize | Get available receive data size |
| WriteAndWaitResponse | Send and receive binary data |
| Reset | Reset printer |
| GetStatus | Get printer status |
| StartCallbackFunction | Start callback of printer status change |
| StopCallbackFunction | Finish callback of printer status change |
| RegisterStyleSheet | Register style sheet This API is not supported. |
| DeleteStyleSheet | Delete style sheet This API is not supported. |
| RegisterLogo | Register logo |
| DeleteLogo | Delete logo |
| GetPrinterInformation | Get printer information |
| GetPrinterInformationNumber | |
| GetPrinterInformationString | |
| ControlTransaction | Start/finish batch processing |
| SelectStandardMode | Select standard mode This API is not supported. |
| SelectPageMode | Select page mode This API is not supported. |
| PrintPageModeData | Print data in page mode This API is not supported. |
| SetPageModeVerticalPosition | Specify absolute vertical position in page mode This API is not supported. |
| SetStandardModeArea | Set print area width in standard mode |
| SetStandardModeAlignment | Alignment |
| SetHorizontalPosition | Specify absolute position |

| Method | Function Summary |
|--|---|
| SetStandardModeBarcodeDirection | Select print direction for barcode or 2-dimensional barcode |
| SetLineSpacing | Set line spacing |
| SetCharacterRightSpace | Set character right spacing |
| SelectCharacterSet | Select character set |
| GetCharacterSet | Get specified character set |
| SelectInternationalCharacterSet | Select international character set |
| GetInternationalCharacter | Get specified international character set |
| SetCharacterFormatting | Format character |
| PrintText | Send text data |
| PrintLogo | Print logo |
| SendDataFile | Send file data |
| PrintBarcode | Print barcode |
| Print2Dcode | Print 2-dimensional barcode |
| PrintPageModeRectangle | Draw rectangle in page mode This API is not supported. |
| FeedLine | Feed paper by line |
| FeedDotLine | Feed paper by dot |
| FeedCutPosition | Feed paper to cut position |
| FeedMarkPosition | Marked paper form feed This API is not supported. |
| SetLog | Specify log output |

Public Property

| Method | Function Summary |
|-----------------|--------------------------|
| IsOpened | Get printer using status |

- **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 "3.2.4(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 "3.2.3(1) PrinterInterface " for available settings. |
| | <i>prnModel</i> | Printer model constant See "3.2.3(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_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: s) <ul style="list-style-type: none"> 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 "3.2.4(1) PrinterException " |
|-----------|---|

| | |
|-------------|--|
| Description | This method connects to a printer specified by <i>prnModel</i> through a communication interface specified by <i>prnIf</i> . |
|-------------|--|

Monitoring the printer status is started by this method. The latest printer status can be retrieved by **GetStatus**. Changes in the printer status can be notified as events by **StatusChanged**, **StartCallbackFunction**, and **StopCallbackFunction**.

Note Do not disable the automatic status response by printer command "Enable/Disable Automatic Status Back" or the memory switch settings of the printer. In that case, the printer status cannot be monitored, and the related function cannot be operated. For the automatic status response and the memory switch settings of the printer, see "MP-B20 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

Close

Finish using printer

Finishes using the printer and monitoring the printer status.

Syntax public void **Close**()

Exception **PrinterException**
 PrinterException is thrown when an error occurs while calling the method.
 For the details, see "3.2.4(1) **PrinterException**"

Description Send data retained by **ControlTransaction** is discarded.

SetWriteTimeout

Set send timeout period

Sets the send timeout period.

Syntax public void **SetWriteTimeout**(Int32 *writeTimeout*)

Parameter *writeTimeout* Send timeout period (millisecond: ms)
 The valid range is 1000 to 90000.
 When the value is specified less than 1000, it is set to 1000 ms.
 When the value is specified more than 90000, it is set to 90000 ms.
 When the value is not specified by this method, the send timeout
 period is 10000 (initial value).

Description The set value can be retrieved by **GetWriteTimeout**.

The setting of this method is enabled in the following methods.

- **Write**
- **WriteAndWaitResponse** (Transmission process part)
- **RegisterLogo**
- **ControlTransaction** (when **TRANSACTION_PRINT** is selected in *control*)
- **PrintText**
- **SendDataFile**
- **PrintBarcode**
- **Print2Dcode**

GetWriteTimeout

Get send timeout period

Gets the send timeout period.

Syntax public Int32 **GetWriteTimeout**()

Return value Send timeout period (millisecond: ms)

SetResponseTimeout

Set receive timeout period

Sets the receive timeout period.

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

Parameter *respTimeout* Receive timeout period (millisecond: ms)
The valid range is 1000 to 90000.
When the value is specified less than 1000, it is set to 1000 ms.
When the value is specified more than 90000, it is set to 90000 ms.
When the value is not specified by this method, the send timeout period is 10000 (initial value).

Description The set value can be retrieved by **GetResponseTimeout**.

The setting of this method is enabled in the following methods.

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

GetResponseTimeout

Get receive timeout period

Gets the receive timeout period.

Syntax public Int32 **GetResponseTimeout**()

Return value Receive timeout period (millisecond: ms)

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 "3.2.4(1) **PrinterException**".

Description When **ControlTransaction** is not used, the timeout period set 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-B20 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

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

This method is aborted by **Reset**.

Read

Get receive data

Gets 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 value is specified more than 4096, it is set to 4096 bytes.
When the value is specified 0 or less, an error is notified.

Return value Receive data

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "3.2.4(1) **PrinterException**".

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

When this method is reexecuted after getting receive data by this method, the subsequent receive data is received from the buffer.

For processing of receive data and the limitations, see "3.1 Data Send/Receive Processing of the Library and the Limitations".

GetReadSize

Get available receive data size

Gets 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 "3.2.4(1) **PrinterException**".

WriteAndWaitResponse

Send and receive binary data

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

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

Parameter *sendBinary* Data to send
A maximum of 16 KB (16384 bytes) of data size can be specified.

bufferSize Maximum size of the data to send
The valid range is 1 to 10485760.
When the value is specified more than 10485760, it is set to 10485760 bytes.
When the value is specified 0 or less, an error is notified.

| | |
|--------------------|--|
| <i>respControl</i> | <p>Operation selection for receive process</p> <p>True : Receive some data or continue to receive data until timeout period is over.</p> <p>False: Receive data of the size specified in <i>bufferSize</i> or continue to receive data until timeout period is over.</p> |
| <i>incAsbData</i> | <p>Whether the automatic status response is included in the receive data or not</p> <p>True : Included</p> <p>False: Not included</p> |

Return value Receive data

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "3.2.4(1) **PrinterException**".

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

Next method execution after this method execution is not processed until completion of this method receiving. For getting the execution response or progress response, use **Read** but not this method. For the details of execution response or progress response, see "MP-B20 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-B20 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

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

This method is aborted by **Reset**.

| | |
|--------------|----------------------|
| Reset | Reset printer |
|--------------|----------------------|

Resets the printer.

Syntax public void **Reset**()

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "3.2.4(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.

GetStatus

Get printer status

Gets 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 "3.2.2(2) **PrinterStatus**".

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "3.2.4(1) **PrinterException**".

StartCallbackFunction

Start callback of printer status change

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 "3.2.4(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 "3.2.4(1) **PrinterException**".

RegisterStyleSheet

Register style sheet

This API is not supported.

When called, it throws **PrinterException**, and returns **ERR_UNSUPPORTED_METHOD**.

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

This API is not supported.

When called, it throws **PrinterException**, and returns **ERR_UNSUPPORTED_METHOD**.

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

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 *filePath*, Int32 *regNum*)
 (b) public void **RegisterLogo**(String *filePath*, Int32 *regNum*, Dithering *dithering*)

| | | |
|-----------|------------------|---|
| 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 "3.2.3(3) Dithering " for available settings. |

Exception **PrinterException**
 PrinterException is thrown when an error occurs while calling the method.
 For the details, see "3.2.4(1) **PrinterException**".

Description 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.

- When using Bluetooth:
Execute **Reset**, or restart the printer.
- When using USB:
Restart the printer.

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

The maximum image size that can be registered is 8192 dots in width and 2304 dots in height.

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 *prnInfo*.

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

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

Deletes the logo registered in the printer.

Syntax public void **DeleteLogo**(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 "3.2.4(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 *prnInfo*.

GetPrinterInformation

GetPrinterInformationNumber

GetPrinterInformationString

Get printer information

Gets the printer information.

Syntax (a) public Byte[] **GetPrinterInformation**(PrinterInformation *prnInfo*)

(b) public Int32 **GetPrinterInformationNumber**(PrinterInformation *prnInfo*)

(c) public String **GetPrinterInformationString**(PrinterInformation *prnInfo*)

Parameter *prnInfo* Printer information type to get
See "3.2.3(4) **PrinterInformation**" for available settings and the list of available printer information.
See "MP-B20 SERIES THERMAL PRINTER TECHNICAL REFERENCE" for the details of available printer information.

Return value Printer information

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "3.2.4(1) **PrinterException**".

Description Sends printer command to the printer for responding the printer information based on the specification in *prnInfo*, 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 batch processing

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

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

Parameter *control* Operation selection for batch processing
See "3.2.3(5) **TransactionFunction**" for available settings.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "3.2.4(1) **PrinterException**".

| | |
|-------------|---|
| Description | <p>The procedure of batch processing is as follows:</p> <ol style="list-style-type: none"> (1) Start batch processing. Specify TRANSACTION_START. If TRANSACTION_CLEAR or TRANSACTION_PRINT is specified before starting batch processing, an error will occur. (2) Execute the method. In the case of the target method for batch processing, buffering send data 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 step (3). In the case of a method other than the target method for batch processing, the send data is immediately executed without being buffered. (3) Finish batch processing. When TRANSACTION_PRINT is specified, buffering is finished, and the buffered send data is sent to the printer. Send processing of the send data is as follows: <ul style="list-style-type: none"> · The send timeout period is the value set by SetWriteTimeout. · If an error occurs, unsent send data is discarded. · If Reset is executed, sending is interrupted and unsent send data is discarded. · When TRANSACTION_START is executed from another thread, sending is continued and buffering is newly started in another thread. When TRANSACTION_START is specified, buffering is continued, but the buffered send data is discarded. When TRANSACTION_CLEAR is specified, buffering is canceled and the buffered send data is discarded. |
|-------------|---|

The target methods for batch processing are as follows:

- **Write**
- **SetStandardModeArea**
- **SetStandardModeAlignment**
- **SetHorizontalPosition**
- **SetStandardModeBarcodeDirection**
- **SetLineSpacing**
- **SetCharacterRightSpace**
- **SetCharacterFormatting**
- **PrintText**
- **PrintLogo**
- **SendDataFile**
- **PrintBarcode**
- **Print2Dcode**
- **FeedLine**
- **FeedDotLine**
- **FeedCutPosition**

Select standard mode

Syntax `public void SelectStandardMode()`

Select page mode

[illegible]

Print page mode data

Syntax `public void PrintPageModeData()`

Specify absolute vertical position in page mode

Syntax `public void SetPageModeVerticalPosition(Int32 verticalPosition)`

Set print area width in standard mode

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

| | |
|---------------------|--|
| <i>prnAreaWidth</i> | Print area width (dot) The valid range is 1 to 384. |
|---------------------|--|

3-18

Description The relation between left margin position and print area width is shown in Figure 3-1. The print data is mapped to the shaded print area.

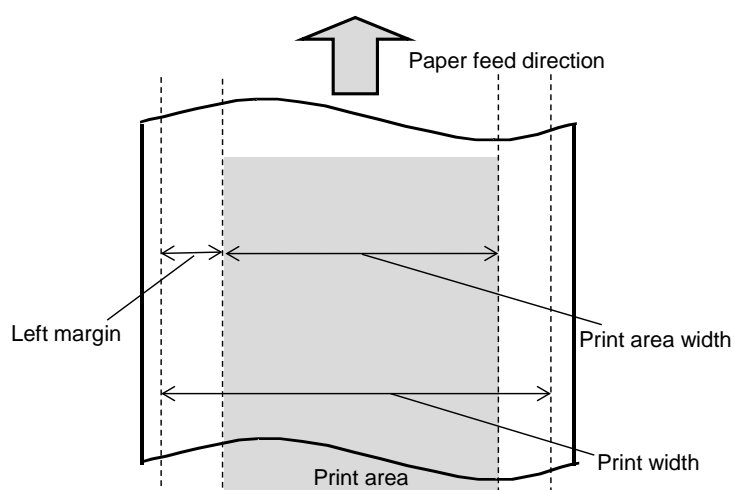


Figure 3-1

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

When the printer is initialized, *leftMargin* is set to 0, and *prnAreaWidth* is set to the print width. For printer initialization, see "MP-B20 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

| SetStandardModeAlign | Alignment |
|----------------------|-----------|
|----------------------|-----------|

Sets the alignment.

Syntax `public void SetStandardModeAlign(Alignment align)`

Parameter *align* Alignment
See "3.2.3(6) **Alignment**" for available settings.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "3.2.4(1) **PrinterException**".

Description When the print area is set in **SetStandardModeArea**, the position is aligned within the set print area.

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

When the printer is initialized, the parameter of this method is set to the initial value. For printer initialization, see "MP-B20 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

SetHorizontalPosition

Set absolute position

Specifies the absolute position in horizontal direction.

Syntax public void **SetHorizontalPosition**(Int32 *horizontalPosition*)

Parameter *horizontalPosition* Horizontal print starting position (dot)
The valid range is 0 to 383.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "3.2.4(1) **PrinterException**".

Description The left margin is based on the one set in **SetStandardModeArea**.

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

When the printer is initialized, the setting in this method is disabled.
For printer initialization, see "MP-B20 SERIES THERMAL PRINTER TECHNICAL
REFERENCE".

SetStandardModeBarcodeDirection

Set print direction for barcode or 2-dimensional barcode

Sets the print direction for barcode or 2-dimensional barcode.

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

Parameter *rotate* Print direction
Only **ROTATE_CURRENT** (Current setting) is valid.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "3.2.4(1) **PrinterException**".

Note The print direction of barcode or 2-dimensional barcode cannot be changed by this method.
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
"3.2.3(14) **Rotate**".

SetLineSpacing

Set line spacing

Sets 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 "3.2.4(1) **PrinterException**".

Description When the printer is initialized, *lineSpacing* is set to 34.
For printer initialization, see "MP-B20 SERIES THERMAL PRINTER TECHNICAL
REFERENCE".

SetCharacterRightSpace

Set character right spacing

Sets the character right spacing.

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 "3.2.4(1) **PrinterException**".

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

When the printer is initialized, *space* is set to 0.
For printer initialization, see "MP-B20 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

SelectCharacterSet

Select character set

Selects the character set.

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

Parameter *charSet* Character set
See "3.2.3(7) **CharacterSet**" for available settings.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "3.2.4(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-B20 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

GetCharacterSet

Get specified character set

Gets the specified character set.

Syntax public CharacterSet **GetCharacterSet**()

Return value Character set

Description For available settings, see "3.2.3(7) **CharacterSet**".

SelectInternationalCharacterSet

Select international character set

Selects the international character set.

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

Parameter *intCharSet* International character set
See "3.2.3(8) **InternationalCharacterSet**" for available settings.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "3.2.4(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-B20 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

GetInternationalCharacter

Get specified international character set

Gets the specified international character set.

Syntax public InternationalCharacterSet **GetInternationalCharacter**()

Return value International character set

Description For available settings, see "3.2.3(8) **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 "3.2.3(9) **CharacterType**" for available settings.

verticalScale Character size (in vertical direction)
See "3.2.3(10) **CharacterScale**" for available settings.

horizontalScale Character size (in horizontal direction)
See "3.2.3(10) **CharacterScale**" for available settings.

underline Underline
See "3.2.3(11) **Underline**" for available settings.

bold Bold print
See "3.2.3(12) **Bold**" for available settings.

| | |
|----------------|---|
| <i>reverse</i> | Reverse print See "3.2.3(13) Reverse " for available settings. |
| <i>rotate</i> | Character rotation print When the print data remains in the printer, this method is not executed. Execute this method after all printing is completed. 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 "3.2.3(14) 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 "3.2.4(1) PrinterException ". |
| Description | When the printer is initialized, the parameter of this method is set to the initial value. For printer initialization, see "MP-B20 SERIES THERMAL PRINTER TECHNICAL REFERENCE". |

| | |
|------------------|-----------------------|
| PrintText | Send text data |
|------------------|-----------------------|

Sends the text data.

Syntax public void **PrintText**(String *text*)

Parameter *text* Text data to send to the printer

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "3.2.4(1) **PrinterException**".

Description This method sends the text data specified in *text* 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-B20 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

This method does not add a line feed code at the end of the text data.

A maximum of 16 KB (16384 bytes) of data size can be specified.

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

This method is aborted by **Reset**.

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 "3.2.4(1) **PrinterException**".

Description Logo registration is done by **RegisterLogo**.

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

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 send to the printer

dithering Dithering
It is enabled when the extension of the file specified at *filePath* is .bmp.
See "3.2.3(3) **Dithering**" for available settings.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "3.2.4(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:
Execute **Reset**, or restart the printer.
- When using USB:
Restart the printer.

- When the file extension is .txt:
The 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-B20 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.

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-B20 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 | <pre>(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>)</pre> | | | | | | |
| Parameter | <table> <tr> <td style="vertical-align: top;"><i>type</i></td><td>Barcode type See "3.2.3(15) TypeBarcode" for available settings.</td></tr> <tr> <td style="vertical-align: top;">(a) <i>text</i></td><td>Barcode text data</td></tr> <tr> <td style="vertical-align: top;">(b) <i>binary</i></td><td>Barcode binary data</td></tr> </table> | <i>type</i> | Barcode type See "3.2.3(15) TypeBarcode " for available settings. | (a) <i>text</i> | Barcode text data | (b) <i>binary</i> | Barcode binary data |
| <i>type</i> | Barcode type See "3.2.3(15) 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 150 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 : From 0x00 to 0x50

CODE B : From 0x00 to 0x7F

CODE C : 2 digit numbers from 00 to 99 (0x00 to 0x63)

moduleWidth Barcode module width or fine element
See "3.2.3(16) **ModuleWidthBarcode**".

moduleHeight Barcode module height (dot)
The table shows the available settings when the barcode type is as follows.

| Barcode Type | <i>moduleHeight</i> |
|---------------------------------|----------------------------|
| UPC-A | 0 ^{*1} , 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) | |

*1: When specified 0, the module height is automatically set to the initial value of a barcode (162 dots).

Regarding the following barcode types, the settable range of the barcode height depends on the barcode module width.

When the *moduleHeight* value is specified less than the minimum value, the value is automatically changed to the minimum value.

| Barcode Type | <i>moduleHeight</i> |
|------------------------------|------------------------------|
| GS1 Databar Omni-directional | |
| MODULE_WIDTH_BARCODE_2 | 0 ^{*1} , 66 to 255 |
| MODULE_WIDTH_BARCODE_3 | 0 ^{*1} , 99 to 255 |
| MODULE_WIDTH_BARCODE_4 | 0 ^{*1} , 132 to 255 |
| MODULE_WIDTH_BARCODE_5 | 0 ^{*2} , 165 to 255 |
| MODULE_WIDTH_BARCODE_6 | 0 ^{*3} , 198 to 255 |
| GS1 Databar Truncated | |
| MODULE_WIDTH_BARCODE_2 | 0 ^{*1} , 26 to 255 |
| MODULE_WIDTH_BARCODE_3 | 0 ^{*1} , 39 to 255 |
| MODULE_WIDTH_BARCODE_4 | 0 ^{*1} , 52 to 255 |
| MODULE_WIDTH_BARCODE_5 | 0 ^{*1} , 65 to 255 |
| MODULE_WIDTH_BARCODE_6 | 0 ^{*1} , 78 to 255 |
| GS1 Databar Limited | |
| MODULE_WIDTH_BARCODE_2 | 0 ^{*1} , 20 to 255 |
| MODULE_WIDTH_BARCODE_3 | 0 ^{*1} , 30 to 255 |
| MODULE_WIDTH_BARCODE_4 | 0 ^{*1} , 40 to 255 |
| MODULE_WIDTH_BARCODE_5 | 0 ^{*1} , 50 to 255 |
| MODULE_WIDTH_BARCODE_6 | 0 ^{*1} , 60 to 255 |

| Barcode Type | <i>moduleHeight</i> |
|-------------------------------|------------------------------|
| GS1 Databar Expanded | |
| MODULE_WIDTH_BARCODE_2 | 0 ^{*1} , 68 to 255 |
| MODULE_WIDTH_BARCODE_3 | 0 ^{*1} , 102 to 255 |
| MODULE_WIDTH_BARCODE_4 | 0 ^{*1} , 136 to 255 |
| MODULE_WIDTH_BARCODE_5 | 0 ^{*4} , 170 to 255 |
| MODULE_WIDTH_BARCODE_6 | 0 ^{*5} , 204 to 255 |

*1: When specified 0, the module height is automatically set to the initial value of a barcode (162 dots).

*2: When specified 0, the module height is automatically set to 165 dots.

*3: When specified 0, the module height is automatically set to 198 dots.

*4: When specified 0, the module height is automatically set to 170 dots.

*5: When specified 0, the module height is automatically set to 204 dots.

hri Barcode HRI character
See "3.2.3(17) **HriPositionBarcode**" for available settings.

nwRatio Barcode N:W ratio
The available setting varies depending on the setting of *type*.
For the details, see "3.2.3(18) **NwRatioBarcode**".

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "3.2.4(1) **PrinterException**".

Description When the print data remains in the printer, this method is not executed.
Execute this method after all 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 an 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 the set 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.

Print2Dcode

Print 2-dimensional barcode

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 | <i>type</i> | 2-dimensional barcode type See "3.2.3(19) Type2Dcode " for available settings. |
| | (a) <i>text</i> | Barcode text data |
| | (b) <i>binary</i> | Barcode binary data |
| | <i>mode</i> | 2-dimensional barcode mode The available settings vary depending on the setting of <i>type</i> . For the details, see "3.2.3(20) 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 <i>text</i> or <i>binary</i> . 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 <i>text</i> or <i>binary</i> . |
| | <i>moduleSize</i> | 2-dimensional barcode module size The available settings vary depending on the setting of <i>type</i> . For the details, see "3.2.3(21) ModuleSize2DCode ". |
| | <i>moduleHeight</i> | 2-dimensional barcode module height (dot) <i>moduleHeight</i> 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 |

| | |
|---------------|--|
| <i>column</i> | Number of columns in 2-dimensional barcode <i>column</i> 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.

| | |
|------------|--|
| <i>row</i> | Number of rows in 2-dimensional barcode <i>row</i> 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 settings vary depending on the setting of *type*.
For the details, see "3.2.3(22) **ErrorCorrect2DCode**".

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "3.2.4(1) **PrinterException**".

Description When the print data remains in the printer, this method is not executed.
Execute this method after all 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 the set 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.

PrintPageModeRectangle

Draw rectangle in page mode

This API is not supported.

When called, it throws **PrinterException**, and returns **ERR_UNSUPPORTED_METHOD**.

Syntax public void **PrintPageModeRectangle**(Int32 *startX*,
Int32 *startY*,
Int32 *width*,
Int32 *height*,
Int32 *thickness*)

FeedLine

Feed paper by line

Feeds the paper by line.

Syntax public void **FeedLine**(Int32 *lines*)

Parameter *lines* 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 "3.2.4(1) **PrinterException**".

Description The paper is fed by the specified number of lines.
When data exists in the line buffer of the printer, paper feed is performed after printing 1 line.

The paper feed length for 1 line (line spacing) can be set 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)
The valid range is 0 to 8192.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "3.2.4(1) **PrinterException**".

Description The paper is fed by the specified number of dots.
When data exists in the line buffer of the printer, paper feed is performed after printing 1 line.

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 "3.2.4(1) **PrinterException**".

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

FeedMarkPosition

Marked paper form feed

This API is not supported.
When called, it throws **PrinterException**, and returns **ERR_UNSUPPORTED_METHOD**.

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

SetLog

Set log output

Sets the log output.

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

Parameter *logLevel* Log output level
Specify 0. When specified 0, an error log is output.
Do not specify other than 0.

logFileSize Maximum size of log file
See "3.2.3(23) **LogFileSize**" for available settings.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling the method.
For the details, see "3.2.4(1) **PrinterException**".

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

Log file name: PrinterManager.log.x (The range of x 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

Get printer using status

Gets 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 | Get offline error status |
| GetErrHardware | Get hardware error status |
| GetErrVoltage | Get Vp voltage error status |
| GetErrHeadTemperature | Get head temperature error status |
| GetErrOutOfPaper | Get out-of-paper error status |
| GetErrBattery | Get battery error status |
| GetStateFeedSwitch | Get feed switch status |
| GetStatePaperFeed | Get paper feed status |
| GetStateReturnWaiting | Get return waiting status |
| GetStateFlashMemoryRewriting | Get FLASH memory rewrite status |
| GetStateBattery | Get battery voltage status |

- **Public Methods**

| | |
|----------------------|---------------------------------|
| GetErrOffline | Get offline error status |
|----------------------|---------------------------------|

Gets the offline error status.

Syntax public Boolean **GetErrOffline()**

Return value True : Offline
 False : Online

| | |
|-----------------------|----------------------------------|
| GetErrHardware | Get hardware error status |
|-----------------------|----------------------------------|

Gets the hardware error status.

Syntax public Boolean **GetErrHardware()**

Return value True : Error
 False : No error

GetErrVoltage

Get Vp voltage error status

Gets the Vp voltage error status.

Syntax public Boolean **GetErrVoltage()**

Return value True : Error
 False : No error

GetErrHeadTemperature

Get head temperature error status

Gets the head temperature error status.

Syntax public Boolean **GetErrHeadTemperature()**

Return value True : Error
 False : No error

GetErrOutOfPaper

Get out-of-paper error status

Gets the out-of-paper error status.

Syntax public Boolean **GetErrOutOfPaper()**

Return value True : Error
 False : No error

GetErrBattery

Get battery error status

Gets the battery error status.

Syntax public Boolean **GetErrBattery()**

Return value True : Error
 False : No error

GetStateFeedSwitch

Get feed switch status

Gets the feed switch status.

Syntax public Boolean **GetStateFeedSwitch()**

Return value True : ON
 False : OFF

GetStatePaperFeed

Get paper feed status

Gets the paper feed status.

Syntax public Boolean **GetStatePaperFeed()**

Return value True : Feeding
 False : Stopped

GetStateReturnWaiting

Get return waiting status

Gets the return waiting status.

Syntax public Boolean **GetStateReturnWaiting()**

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

GetStateFlashMemoryRewriting

Get FLASH memory rewrite status

Gets 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

Get battery voltage status

Gets the battery voltage status.

Syntax public Int32 **GetStateBattery()**

| | | | |
|--------------|---|--------------------------------------|---|
| Return value | 0 | : No battery | |
| | 1 | : Battery remaining capacity: Low | (battery remaining capacity: approx. 10%) |
| | 3 | : Battery remaining capacity: Middle | (battery remaining capacity: approx. 40%) |
| | 7 | : Battery remaining capacity: Full | (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 | Get 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 "3.2.4(1) **PrinterException**".

StartDiscoveryPrinter Start printer search

Starts the printer search.

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

Parameter *handler* Instance of **DiscoveryHandler**

prnIf Communication interface constant
 See "3.2.3(1) **PrinterInterface**" for available settings.
 PRN_IF_USB is not supported.

retry Number of retry times (time)
 • When **PRN_IF_BT** is specified:
 The specified value is ignored, and the search is performed
 only once.

timeout Timeout period for 1 time search (second: s)
 • 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 "3.2.4(1) **PrinterException**".

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

Get 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 "3.2.4(1) **PrinterException**".

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

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

GetFoundPrinter

Get information list of found printer

Gets 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 the method of **DiscoveredPrinter** class, the printer model name, Bluetooth address and serial number can be retrieved.
 For the methods of **DiscoveredPrinter** class, see "3.2.2(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 and serial number can be retrieved. This class provides the following functions.

Public Methods

| Method | Function Summary |
|----------------------------|---|
| GetPrinterModel | Get printer model name |
| GetBluetoothAddress | Get Bluetooth address |
| GetIpAddress | Get IP address This API is not supported. |
| GetMacAddress | Get MAC address This API is not supported. |
| GetSerialNumber | Get serial number |

- **Public Methods**

GetPrinterModel Get printer model name

Gets the character string of printer model name.

Syntax public String **GetPrinterModel()**

Return value Printer model name

GetBluetoothAddress Get Bluetooth address

Gets the character string of Bluetooth address.

Syntax public String **GetBluetoothAddress()**

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

GetIpAddress Get IP address

This API is not supported.
When called, it returns empty string.

Syntax public String **GetIpAddress()**

GetMacAddress Get MAC address

This API is not supported.
When called, it returns empty string.

Syntax public String **GetMacAddress()**

Gets the character string of serial number.

Syntax public String **GetSerialNumber**()

Return value Serial number

3.2.3 Enumerated Type (Enum)

(1) PrinterInterface

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

| Name | Description |
|------------|-------------|
| PRN_IF_BT | Bluetooth |
| PRN_IF_USB | USB |

(2) PrinterModel

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

| Name | Description |
|------------------|-------------|
| PRN_MODEL_MP_B20 | MP-B20 |

(3) Dithering

Enumerator used for dithering of **RegisterLogo** 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-B20 SERIES THERMAL PRINTER TECHNICAL REFERENCE" for details of printer information to get.

| Name | Description (Printer Information) |
|-------------------------------|--|
| GET_NV_MEM_CAP | Send NV graphics memory capacity |
| GET_NV_MEM_REM_CAP | Send NV graphics memory remaining capacity |
| GET_NV_MEM_KEYCODE_LIST | Send NV graphics key code list |
| GET_REM_USER_MEM_CAP_DEFRAG | Send remaining user area after defragment |
| GET_REM_USER_MEM_CAP | Send remaining user area |
| GET_FUNC_SET_RESP | Send function setting |
| GET_PRN_ID_MODEL | Send printer ID (Model ID) |
| GET_PRN_ID_TYPE | Send printer ID (Type ID) |
| GET_PRN_ID_ROM_VER | Send printer ID (ROM version ID) |
| GET_PRN_ID_FIRM_VER_MAIN | Send printer ID (Firmware version (main)) |
| GET_PRN_ID_MFR | Send printer ID send (Manufacturer) |
| GET_PRN_ID_MODEL_NAME | Send printer ID (Model name) |
| GET_PRN_ID_FIRM_VER_BOOT | Send printer ID (Firmware version (boot)) |
| GET_PRN_ID_FIRM_CHECKSUM_BOOT | Send printer ID (Firmware checksum (boot)) |
| GET_PRN_ID_FIRM_CHECKSUM_MAIN | Send printer ID (Firmware checksum (main)) |

| Name | Description (Printer Information) |
|---------------------------------------|---|
| GET_PRN_ID_FIRM_CHECKSUM | Send printer ID (Firmware checksum (main + boot)) |
| GET_MAINT_NUM_FEED_LINE | Send maintenance counter (Number of paper feed lines (in 100 dot-lines)) |
| GET_MAINT_NUM_HEAD_ACTIVE | Send maintenance counter (Number of thermal head activation times (in 100 dot-lines)) |
| GET_MAINT_DRIVE_TIME | Send maintenance counter (Drive time of printer (in minutes)) |
| GET_MAINT_NUM_FEED_LINE_INTEGRATION | Send maintenance counter (Number of paper feed lines (in 100 dot-lines) (integration)) |
| GET_MAINT_NUM_HEAD_ACTIVE_INTEGRATION | Send maintenance counter (Number of thermal head activation times (in 100 dot-lines) (integration)) |
| GET_MAINT_DRIVE_TIME_INTEGRATION | Send maintenance counter (Drive time of printer (in minutes) (integration)) |
| GET_HFONT_24_CHECKSUM | Send 1-byte font ID (24-dot font, checksum) |
| GET_HFONT_24_ID | Send 1-byte font ID (24-dot font, ID) |
| GET_HFONT_24_INT_CHAR | Send 1-byte font ID (24-dot font, registered international character) |
| GET_HFONT_16_CHECKSUM | Send 1-byte font ID (16-dot font, checksum) |
| GET_HFONT_16_ID | Send 1-byte font ID (16-dot font, ID) |
| GET_HFONT_16_INT_CHAR | Send 1-byte font ID (16-dot font, registered international character) |
| GET_FFONT_LANG | Send 2-byte font ID (Language) |
| GET_FFONT_STANDARD | Send 2-byte font ID (Standard) |
| GET_FFONT_COMPANY | Send 2-byte font ID (Company name) |
| GET_FFONT_CHECKSUM | Send 2-byte font ID (Checksum) |
| GET_FFONT_DATA_SIZE | Send 2-byte font ID (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 printing and batch processing |

(6) **Alignment**

Enumerator used for specifying the alignment of **SetStandardModeAlignment**.

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

(7) **CharacterSet**

Enumerator used for selecting the character set of **SelectCharacterSet** and **GetCharacterSet**.
The initial value is as follows according to the language setting of WinCE device.

Japanese: **CODEPAGE_KATAKANA**
Other than Japanese: **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) |

(8) **InternationalCharacterSet**

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

The initial value is as follows according to the language setting of WinCE device.

Japanese: **INT_CHAR_SET_JAPAN**

Other than Japanese: **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 I |
| 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 |

(9) **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 X 12) regardless of specifying the character font in **SetCharacterFormatting**.

(10) **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) |

(11) **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 |

(12) **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 |

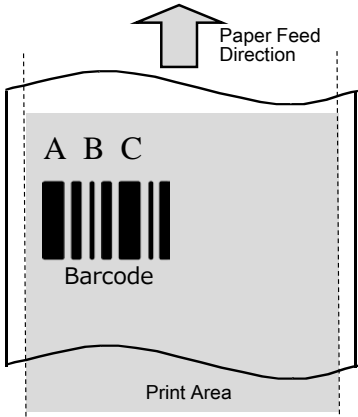
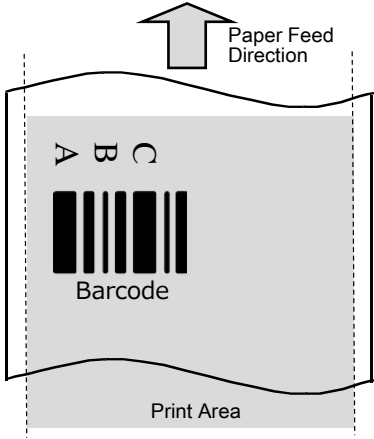
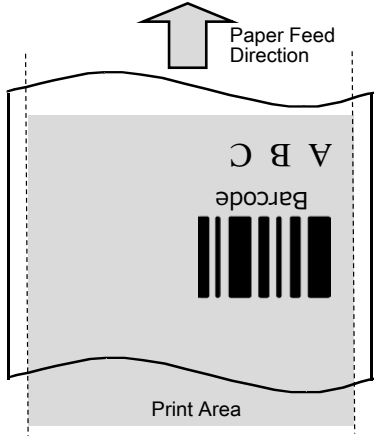
(13) **Reverse**

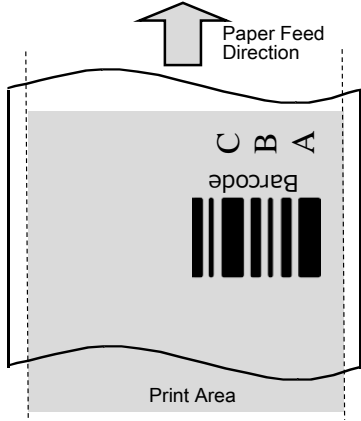
Enumerator used for specifying the reverse print of **SetCharacterFormatting**.

| 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 |

(14) 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  | Returns ERR_PARAM |
| ROTATE_90_TO_RIGHT |  | |
| ROTATE_180 |  | |

| Name | Description (" " : Initial Value) | |
|-------------------|---|---------------------------------|
| | SetCharacterFormatting | SetStandardModeBarcodeDirection |
| ROTATE_90_TO_LEFT |  | Returns ERR_PARAM |

(15) 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 |

(16) **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 |

(17) **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) |

(18) **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 |

(19) **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 |

(20) **Mode2Dcode**

Enumerator used for 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 |

| Name | Description | |
|--------------------------------|---|------------|
| MODE_2DCODE_MAXI_CODE_5 | MaxiCode | Mode5 |
| MODE_2DCODE_NONE | GS1 Databar Stacked GS1 Databar Stacked Omni-directional GS1 Databar Expanded Stacked | No setting |

(21) **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 setting |
| | 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 | |

(22) 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 setting |

(23) 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) |

(24) 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 | Connection status is not monitored. |
| ERR_INIT_FAILED | Failed to initialize. |
| ERR_DATA_SIZE_ZERO | 0 byte data size 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. |
| ERR_UNSUPPORTED_METHOD | The method is not supported. |

3.2.4 Exception

(1) PrinterException

- **Summary**

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

Public Methods

| Method | Function Summary |
|---------------------|------------------|
| GetErrorCode | Get error code |

- **Public Methods**

| | |
|---------------------|------------------------|
| GetErrorCode | Get error codes |
|---------------------|------------------------|

Gets the error code for the thrown exception.

Syntax public Errorcode **GetErrorCode()**

Return value Error code

Description See "3.2.3(24) **ErrorCode**" for details on the error.

Chapter 4

Sample Program

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

Confirm that WinCE application development environment is established. See "Chapter 2 How to Use the Library" for the required development environment.

4.1 How to Use

This section describes how to use the sample program.

4.1.1 CAB File

- (1) Copy the CAB file located under [System Drive]:\WindowsCE\Sample\cab folder to an appropriate folder on the WinCE device.
- (2) Execute the CAB file from File Explorer.
- (3) Execute sample programs from [Program] on WinCE device.

4.1.2 Project Format Program

- (1) Copy [System Drive]:\WindowsCE\Sample\source folder to local computer.
- (2) Execute the SLN file in the copied source folder from File Explorer.

4.2 Screen Layout

This section describes the screen layout of sample program installed in "4.1.1 CAB File".



| Item | Description |
|-------------------|--|
| Connection Type | Selects 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], printer search is started. By selecting a printer from the displayed printer list, the address of the selected printer is automatically input. |
| Various Functions | The buttons are for executing various functions. By scrolling the screen, you can find the methods and properties not displayed in the screen. See "Chapter 3 Function of the Library" for details of each method and property. |

4.3 Precaution

The sample program is subject to change without notice.

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

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 | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ |
| D0 | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ |
| 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 | Đ | 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 | | - | + | + | + | + | + | + | + | + | + | + | + |
| D0 | ⌌ | ⌌ | ⌌ | ⌌ | ⌌ | ⌌ | ⌌ | ⌌ | ⌌ | ⌌ | ⌌ | ⌌ | ⌌ | ⌌ | ⌌ | ⌌ |
| 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 | À | 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 | · | · | · | · | · | · | · | · | · | · | · | · |
| B0 | ☐ | ☐ | ☐ | | | | | | | | | | | | | |
| C0 | L | L | T | | - | + | + | + | + | + | + | + | + | + | + | + |
| D0 | ⌌ | ⌌ | ⌌ | ⌌ | ⌌ | ⌌ | ⌌ | ⌌ | ⌌ | ⌌ | ⌌ | ⌌ | ⌌ | ⌌ | ⌌ | ⌌ |
| 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 | Þ | ƒ |
| A0 | á | í | ó | ú | ñ | Ñ | ä | ö | ¿ | ¬ | ½ | ¼ | ¡ | « | » | |
| B0 | ☐ | ☐ | ☐ | | | | | | | | | | | | | |
| C0 | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ |
| D0 | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ |
| E0 | α | β | Γ | π | Σ | σ | μ | τ | φ | θ | Ω | δ | ∞ | φ | ε | Π |
| F0 | ≡ | ± | ≥ | ≤ | | | ÷ | ≈ | ° | • | • | √ | 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 | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ã | Ã | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ |
| D0 | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | € | Í | Î | Ï | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ | ⌞ |
| E0 | Ó | β | Ô | Ò | Õ | 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 | Α | Β | Γ | Δ | Ε | Ζ | Η | Θ | Ι | Κ | Λ | Μ | Ν | Ξ | Ο | Π |
| 90 | Ρ | Σ | Τ | Υ | Φ | Χ | 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 | Ͱ | ͱ | Ͳ | ͳ | ʹ | ͵ | Ͷ | ͷ | ͸ | ͹ | ͺ | ͻ | ͼ | ͽ | Ϳ | ̀ |
| C0 | ́ | ͂ | ̓ | ̈́ | ͅ | ͆ | ͇ | ͈ | ͉ | ͊ | ͋ | ͌ | ͍ | ͎ | ͏ | ͐ |
| D0 | ͑ | ͒ | ͓ | ͔ | ͕ | ͖ | ͗ | ͘ | ͙ | ͚ | ͛ | ͜ | ͝ | ͞ | ͟ | ͠ |
| E0 | ω | ά | έ | ή | ϊ | ί | ό | ύ | ϖ | Ά | Έ | Ή | Ί | Ό | 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 | Ł | Ł | Ł | | | ã | Ã | Ł | Ł | Ł | Ł | Ł | Ł | = | Ł | α |
| D0 | đ | Đ | Ê | Ë | È | € | Í | Î | Ï | Ј | Г | ■ | ■ | | Ì | ■ |
| 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 | ☐ | ☐ | ☐ | | | х | Х | и | И | ¶ | ¶ | ¶ | ¶ | й | Й | ₱ |
| C0 | Ł | Ł | Ł | | | к | К | Ł | Ł | Ł | Ł | Ł | Ł | = | Ł | α |
| D0 | л | Л | м | М | н | Н | о | О | п | П | г | ■ | ■ | П | я | ■ |
| E0 | Я | р | Р | с | С | т | Т | у | У | ж | Ж | в | В | ь | 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 | | | | | |
| A0 | - | £ | £ | £ | £ | £ | £ | £ | £ | £ | £ | £ | £ | £ | £ | £ |
| B0 | • | ١ | ٢ | ٣ | ٤ | ٥ | ٦ | ٧ | ٨ | ٩ | ف | ؛ | س | ص | ش | ؟ |
| C0 | ¢ | ء | آ | أ | ؤ | ع | ئ | ب | ا | ة | ث | ج | ح | خ | د | ذ |
| D0 | ذ | ر | ز | س | ش | ص | ض | ط | ظ | ع | غ | ف | ق | ك | خ | ع |
| E0 | - | ف | ق | ك | خ | ع | ف | ق | ك | خ | ع | ف | ق | ك | خ | ع |
| F0 | - | " | ن | ه | ه | ي | ي | ي | ي | ي | ي | ي | ي | ي | ي | ■ |

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 | Ť | Ž | Ž | | |
| 90 | ‘ | ’ | “ | ” | • | - | - | ™ | š | > | š | ť | ž | ž | | |
| A0 | ˘ | ˘ | Ł | Ł | Ł | Ł | Ł | Ł | Ł | Ł | Ł | Ł | Ł | Ł | Ł | Ł |
| B0 | ° | ± | † | ’ | μ | ¶ | • | ¶ | ¶ | ¶ | ¶ | ¶ | ¶ | ¶ | ¶ | ¶ |
| C0 | Ř | Á | Ā | Ā | Ā | Ā | Ā | Ā | Ā | Ā | Ā | Ā | Ā | Ā | Ā | Ā |
| D0 | Đ | Ň | Ň | Ó | Ô | Ö | Ö | × | Ř | Ů | Ú | Ú | Ú | Ú | Ý | Ť |
| E0 | ř | á | ā | ā | ā | ā | ā | ā | ā | ā | ā | ā | ā | ā | ā | ā |
| F0 | đ | ň | ň | ó | ô | ö | ö | ÷ | ř | ů | ú | ú | ú | ú | ý | ť |

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 | ъ | ѓ | ; | ѓ | ” | ... | † | ‡ | € | ‰ | Љ | < | Њ | ќ | ћ | џ |
| 90 | ђ | ‘ | ’ | “ | ” | • | - | - | ™ | љ | > | њ | ќ | ћ | џ | |
| A0 | ѣ | ѝ | Ј | Ѡ | Г | І | Ѕ | Ё | Є | « | ¬ | - | ® | İ | | |
| B0 | ° | ± | І | і | г | μ | ¶ | • | ё | № | е | » | ј | Ѕ | 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 | í | 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 | Ϊ | Ϋ | 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 | ϊ | ϋ | 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.)