EPSON
EXCEED YOUR VISION

EPSON OPOS ADK for .NET Manual

Application Development Guide

LineDisplay
(DM-D500)

Version 1.14.6   Dec. 2017

Notes

(1) Reproduction of any part of this documentation by any means is prohibited.

(2) The contents of this documentation are subject to change without notice.

(3) Comments and notification of any mistakes in this documentation are gratefully accepted.

(4) This software cannot be used with other equipment that the specified.

(5) EPSON will not be responsible for any consequences resulting from the use of any information in this documentation.

Trademarks

Microsoft®, Windows®, Windows Server® and Windows Vista® are trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. IBM® and PC/AT® are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. Epson® and ESC/POS® are registered trademarks of Seiko Epson Corporation. Other product and company names used herein are for identification purposes only and may be trademarks or registered trademarks of their respective companies. Epson disclaims any and all rights in those marks.

Contents

Chapter 1 Introduction

This section explains how to use LineDisplay when using EPSON OPOS ADK for .NET, including any related information and special notes regarding the device.

For details on the POS for .NET API, refer to the "UnifiedPOS Retail Peripheral Architecture Version 1.14.1" specification and the MSDN "POS for .NET v1.14.1 SDK Documentation":

http://www.omg.org/retail/unified-pos.htm
https://msdn.microsoft.com/en-us/library/bb429024(v=winembedded.4).aspx

1.1 Terminology

- “Unified POS Retail Peripheral Architecture Version 1.14.1” may be abbreviated as “UPOS”.
- “Microsoft POS for .NET” may be abbreviated as “POS.NET”.
- “EPSON OPOS ADK for .NET Version 1.14.6” may be abbreviated as “OPOS.NET”.
- “LineDisplay” may be abbreviated as “the device”.
- "The ServiceObject of LineDisplay provided by OPOS.NET” may be abbreviated as “ServiceObject”.
- “ErrorCode properties of PosControlException” may be abbreviated as “ErrorCode”.
- “ErrorCodeExtended properties of PosControlException” may be abbreviated as “ ErrorCodeExtended”.
- “Exception" indicates “PosControlException”.
- The EPSON original device constant used with this device is defined in “jp.co.epson.uposcommon.EpsonUPOSConst” and “jp.co.epson.uposcommon.EpsonLineDisplayConst”.

Chapter 2 Before Using LineDisplay

This chapter explains how to set up LineDisplay, as well as notes and restrictions on using it.

2.1 Device Setting

After checking the hardware model and the hardware settings, select the correct device using the SetupPOS utility. Refer to "Appendix C Hardware settings" for information on setting the hardware, and "Appendix B SetupPOS Settings" for information on how to use the SetupPOS utility.

2.2 Notes and Restrictions

2.2.1 General

- The flow control of the device supports only DTR/DSR.
- As the device is not equipped with the power recovery function, when rebooting, the **Claim** method is to be executed after the **Release** method.
- When the device is to be rebooted, turn off the power supply for at least 5 second before powering up.
- In the event that the device is rebooted in the midst of executing methods involved in transmitting to device, the data displayed may be corrupted.
- Because data transmitted by utilizing the DISP_DI_OUTPUT command of the **DirectIO** method is not checked by the ServiceObject, the ServiceObject may be obstructed and may lead to unexpected results.
- All the character codes (Unicode) expressed in strings are converted to single-byte codes based on the values set in the **CharacterSet** property. When doing byte-code conversions, sufficient care should be exercised if extended ASCII codes are specified.
- If the same port where POSPrinter is connected is used, the communication conditions of POSPrinter and LineDisplay should be adjusted.
- If used with POSPrinter connected, the status of the LineDisplay may become OFFLINE depending on the status of the POSPrinter. In this state, the methods involved in transmitting to the device (such as **DisplayText** method) will fail.

- If used with POSPrinter connected and when the Display connector of POSPrinter is used, set the DIP switch of “LineDisplay connection status” of POSPrinter or the memory switch to ON.
- If the LineDisplay is connected to IR-700 using the display connector, specify stand-alone connection in the COM4 port of SetupPOS setting.
- The device does not support blinking of display by character units. The setting and cancellation of blinking display apply to the whole screen.
- The escape sequence (ESC|kC) for performing blinking display is ignored. Similarly, the escape sequence (ESC|N) for canceling blinking display does not cancel out the blinking of display.
- The setting of the Attribute parameter of the **DisplayText(At)** method executed in the marquee initialization mode is reflected when coming out of the marquee initialization mode.
- Two types of screen display modes are available, the Character mode and the Graphic mode.
- The display mode created for the device window after the **Open** method is the Character mode
- The total number of windows including Character mode windows and Graphic mode windows that can be created cannot exceed the setting value of the **DeviceWindows** property.
- DM-D500 is not supported for TM-H6000V.

2.2.2 Character mode

In the Character mode window, image display is not supported, regardless of the setting value of **CapBitmap** property. However, the registration and deletion of images are possible through the **SetBitmap** method.

2.2.3 Graphic Mode

- Only one Graphic mode window can be created.
- As the positional information such as cursor position and the size information such as character size and window size are handled in dot units, specify the properties that handle the positional information or size information, and the parameters of methods in dot units. The properties and methods that are handled in dot units are as follows:
  - **Columns** property
  - **CursorColumn** property
  - **CursorRow** property
  - **Rows** property
  - **CreateWindow** method
  - **DisplayTextAt** method
  - **ScrollText** method
- In the Graphic mode window, because the display positions can be changed by dot units, it is possible to display overlapping characters, but the management for this function is not implemented. Due to this reason, redrawing the screen through method such as **RefreshWindow** when overlapping characters are displayed cannot be guaranteed. The overlapping of characters produced by changing font types midway also cannot be guaranteed.
- In the Graphic mode window, characters and images can be displayed simultaneously. In the event that due to the redrawing of screen through method such as **RefreshWindow**, characters and images appear to overlap in the display, images are given priority (front-most) in the display.
- In the event that the screen is redrawn using method such as **RefreshWindow** for other than the Graphic mode window, the Graphic mode window is shifted to the last layer at the back. For this reason, the Graphic mode window may appear to have disappeared.
- If the Graphic mode window is deleted, the contents displayed in the Graphic mode window are all cleared away.
- If the Graphic mode window has been created, the overall device operations become extremely slow. Be aware of the impediments imposed on the display of a large quantity of characters.
- Image display is not supported when in teletype. Due to this reason, if the escape sequence (ESC|#B) that performs image display is included in the *data* parameter of the **DisplayText(At)** method, an exception is thrown.

- There are 2 ways to display images, the direct display method through executing **DisplayBitmap** method, and the method that executes **DisplayText(At)** method containing the escape sequence (ESC|#B) that performs image display after carrying out registration through the **SetBitmap** method. The differences in the processing of these are as follows:
  - **Display through the DisplayBitmap method**
    Each time this method is executed, processing of image acquisition or image conversion is performed. Also in the multiple display of the same image and the redisplay of images once cleared by **ClearText** method, processing such as image acquisition or image conversion is performed each time **DisplayBitmap** method is executed.
  - **Display through the combination SetBitmap method and DisplayText(At) method**
    The processing of image acquisition or image conversion is performed at the time of registration through the **SetBitmap** method. At the time of displaying through the **DisplayText(At)** method, only the acquisition of already registered information is performed. The image registration can be done by using the download function of the device. For this reason, performance improves compared to the method of direct display through the **DisplayBitmap** method.

2.2.4 Escape Sequences

The ServiceObject supports display through escape sequences.

Escape sequences can be used inside the *data* parameter of the **DisplayText(At)** method. And, besides the ones described in the UPOS, those described below can be used. However, the blinking (ESC|kC) described in the UPOS Programming Guide is ignored.

Name	Code	Description
Bold	ESC|bC	Display character is in bold.
Single-sized	ESC|1C	Size is equally set, both vertically and horizontally.
Double-width	ESC|2C	Horizontal character width is made double-sized.
Double-height	ESC|3C	Vertical character height is made double-sized.
Double-sized	ESC|4C	Double-sized both vertically and horizontally.
Horizontal scaling	ESC|#hC	Character’s horizontal size is scaled by the specified multiple. The ‘#’ character is the horizontal scaling factor of character.
Vertical scaling	ESC|#vC	Character’s vertical size is scaled by the specified multiple. The ‘#’ character is the vertical scaling factor of character.

- **Bold (ESC|bC)**
    - This escape sequence sets the display attribute to bold (boldface).
    - Multi-byte characters, and characters defined through the **DefineGlyph** method are not displayed in bold.

- **Single-sized (ESC|1C)**
    - The character scaling is set equally, horizontally and vertically.
    - It is reflected in the Graphic mode window.

- **Double-width (ESC|2C)**
    - The character’s horizontal scaling is made double.
    - The double-width setting is valid until the normal escape sequence (ESC|N) or the escape sequence related to changing horizontal size is specified.
    - The function of this escape sequence is equal to the escape sequence ’ESC|2hC’ that sets horizontal scaling.
    - It is reflected in the Graphic mode window.

- **Double-height (ESC|3C)**
    - The character’s vertical scaling is made double.
    - The double-height setting is valid until the normal escape sequence (ESC|N) or the escape sequence related to changing vertical size is specified.
    - The function of this escape sequence is equal to the escape sequence ’ESC|2vC’ that sets vertical scaling.
    - It is reflected in the Graphic mode window.

- **Double-sized (ESC|4C)**
    - The character’s vertical and horizontal scaling are made double.
    - This setting is valid until the normal escape sequence (ESC|N) or the escape sequence related to changing size is specified.
    - It is reflected in the Graphic mode window.

- **Horizontal scaling (ESC|#hC)**
  - It sets the horizontal scaling of the character.
  - The ‘#’ character specifies the horizontal scaling factor. The valid range is from 1 to 8. If a value outside the valid range is specified, it is adjusted to be within the valid range.
  - The setting for horizontal scaling is valid until the normal escape sequence (ESC|N) or the escape sequence related to changing horizontal size is specified.
  - It is reflected in the Graphic mode window.

- **Vertical scaling (ESC|#vC)**
  - It sets the vertical scaling of the character.
  - The ‘#’ character specifies the vertical scaling factor. The valid range is from 1 to 8. If a value outside the valid range is specified, it is adjusted to be within the valid range.
  - The setting for vertical scaling is valid until the normal escape sequence (ESC|N) or the escape sequence related to changing vertical size is specified.
  - It is reflected in the Graphic mode window.

Chapter 3 Properties and Methods

3.1 Properties

The properties that differ from those functions described in UPOS are shown below.

3.1.1 CapPowerReporting property

Description

The notification capability of the device is identified.

This property is set to one of the following values.

Value	Meaning
PowerReporting.Standard	ServiceObject can determine between and provide notification of the two power statuses: OFF_OFFLINE (the device is turned off or offline) and ONLINE.

3.1.2 PowerState property

Description

The power supply status of the device is identified.

However, if the LineDisplay is connected to the POSPrinter, the value of this property may get updated depending on the power status of the POSPrinter.

3.1.3 DeviceEnabled property

Description

When set to “true”, the operation of the marquee mode and teletype mode is resumed based on the properties set at that time.

When set to “false”, the operation of the marquee mode and teletype mode is halted.

3.1.4 BlinkRate property

Description

The blinking synchronization of the device is set.

The values that can be set for this property are positive numbers of 1 and above. Actually, the value is corrected to the nearest setting value supported by the device. The correction is performed during the setting of the **BlinkRate** property, and the setting value after correction is reflected also in the blinking cycle of the device.

The correction methods for the setting value are as follows:

- Setting values are corrected to the nearest value that is divisible by 100. For example, the values of 50 – 149 are corrected to 100, the values of 150 – 249 to 200, and so on. However, the values of 1 – 49 are corrected to 100.
- The largest setting value after correction is 25400. If the value is set to greater than 25400, it is corrected to 25400.

3.1.5 CharacterSet property

Description

When the **DeviceEnabled** property is true, it is initialized according to the setting of “Multibyte character type” of the setup utility.

If the property value is set to 932, the display for ASCII code 0x5C is changed to the yen symbol ”¥”.

3.1.6 CharacterSetList property

Description

It is initialized according to the setting of “Multibyte character type” of the setup utility by the **Open** method.

3.1.7 DeviceBrightness property

Description

The device brightness is set in terms of percentages from 0 to 100.

The relation between brightness set at the device and the property setting value is as follows:

DeviceBrightness property	Device brightness
0 – 19	0%
20 – 39	20%
40 – 59	40%
60 – 79	60%
80 – 100	100%

3.1.8 DeviceColumns property

Description

It is initialized according to the setting of “Rows and Columns” of the setup utility by the **Open** method.

3.1.9 DeviceRows property

Description

It is initialized according to the setting of “Rows and Columns” of the setup utility by the **Open** method.

3.1.10 DeviceWindows property

Description

This property is always set to 4.

3.1.11 GlyphHeight property

Description

It is initialized according to the setting of “Rows and Columns” of the setup utility by the **Open** method.

3.1.12 GlyphWidth property

Description

It is initialized according to the setting of “Rows and Columns” of the setup utility by the **Open** method.

3.1.13 ScreenModeList property

Description

It is initialized according to the setting of “Rows and Columns” of the setup utility by the **Open** method.

3.2 Methods

The methods that differ from those functions described in UPOS are shown below.

3.2.1 Claim method

Description

When the **Claim** method is executed, the connection of the port to which the device is connected, and the acquisition and setting of device information are performed.

If the acquisition and setting of device information, or port connection fails, an exception is thrown.

The execution of the **Claim** method after executing the **Release** method redraws the contents that were displayed when the **Release** method was executed. However, if the **Close** method or the **Open** method was executed prior to executing the **Claim** method, there is no redrawing.

For pass-through connections, a time-out might occur causing an OFFLINE error to be returned for the **Claim** method. This is affected by the length of the serial cable, and occurs more frequently with longer cables (3m or longer). If this error occurs, execute the **Claim** method again.

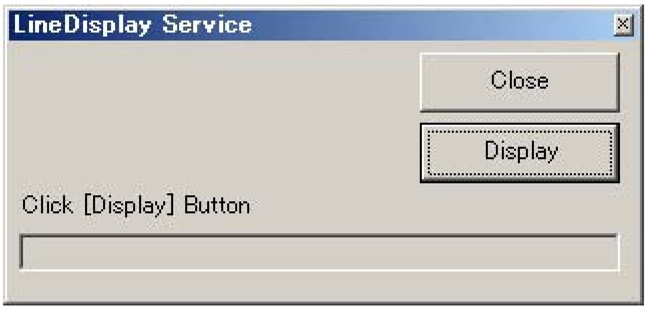
3.2.2 CheckHealth method

Description

This ServiceObject supports only “HealthCheckLevel.Interactive”.

It can be executed when all the windows are in Immediate mode.

When “HealthCheckLevel.Interactive” is executed, the following dialog box appears.



Click [Display] Button

The operations when the respective buttons are clicked are as follows:

- **[Display] button**

It performs testing.

The following character strings are displayed to the device:
• Interactive HCheck !!

• DeviceName=device name

- **[Close] button**

[Close] button

It ends the processing of **CheckHealth** method.

After executing the method, the following values can be acquired in the **CheckHealthText** property.

And besides, the following value is retrieved as the returned value of method.

Value	Meaning
Interactive HCheck: Canceled	Ends CheckHealth without doing anything.
Interactive HCheck: Complete	Ends CheckHealth after the last operation ends normally.
Interactive HCheck: Error - <*Message*>	Ends CheckHealth after the last operation ends with an error. The content of the error appears as a *Message*.

3.2.3 DirectIO method

Description

The **DirectIO** method can be used when the **DeviceEnabled** property is "true".

The functions supported by the **DirectIO** method are as follows:

Command	Outline of functions
DISP_DI_OUTPUT	Specified codes are transmitted to the device.
DISP_DI_GRAPHIC	The next window created by the **CreateWindow** method is in Graphic mode.
DISP_DI_SETLINESPACE	Sets the amount of space to shift during line feed.
DISP_DI_SETFONT	Specifies the display font.
DISP_DI_GETMODE	Obtains the window screen mode set in the current **CurrentWindow** property.
DISP_DI_GW_STYLE	Sets the transparent style of Graphic mode window.

- **DISP_DI_OUTPUT command**

Parameter	
command	DISP_DI_OUTPUT
data	Not used
object(byte[] type)	Transmission data

Description

It sends the data specified in the *object* parameter directly to device.

It is to be used only when sending ESC/POS commands to the device.

ServiceObject does not check data sent through this command. ESC/POS commands such as those that change the amount of line feed or font size obstruct the operations of ServiceObject thereafter, and should not be sent.

- **DISP_DI_GRAPHIC command**

Parameter	
command	DISP_DI_GRAPHIC
data	Not used
object	Not used

Description

When creating a Graphic mode window, execute the **DirectIO** method through this command.

After executing this command and until it is canceled, the execution of the **CreateWindow** method creates a Graphic mode window. At that time, specify the parameter of the **CreateWindow** method in dot units.

Because only 1 Graphic mode window can be created, when this command is executed when a Graphic mode window is already present, an exception is thrown.

This command is canceled when any of the processes below is performed. The command cancellation is performed regardless of the success or failure of these processes.

- Executing the **CreateWindow** method
- Executing the **Release** method
- Setting the **DeviceEnabled** property to false

- **DISP_DI_SETLINESPACE command**

Parameter

command	DISP_DI_SETLINESPACE
data	Specify the amount of line feed in dot units.
object	Not used

Description

It sets the amount of space to shift during a line feed.

This setting is valid for the Graphic mode window. This setting is reflected in the displayed characters when **DisplayText(At)** method is next executed. The newly set up amount of space to shift is not reflected in the characters already displayed.

When using with double-height characters, execute **DisplayText(At)** method after setting the amount of space to shift by combining with the scaling factor that is actually used.

The recommended amounts of space to shift are as follows:

Vertical scaling factor	DISP_DI_FONT_A	DISP_DI_FONT_B
1	16	8
2	32	16
3	48	24
4	64	32

Default: 16

- **DISP_DI_SETFONT command**

Parameter	
command	DISP_DI_SETFONT
data	Specify any of the following. • DISP_DI_FONT_A • DISP_DI_FONT_B
object	Not used

Description

It specifies the display font.

This setting is valid for the Graphic mode window. This setting is reflected in the displayed characters when **DisplayText(At)** method is next executed. The newly set up display font is not reflected in the characters already displayed.

The values that can be set and the functions of the *data* parameter are as follows:

data	Outline of functions
DISP_DI_FONT_A	Sets the display font to the A font (16-dot height x 8-dot width).
DISP_DI_FONT_B	Sets the display font to the B font (8-dot height x 6-dot width).

Default: DISP_DI_FONT_A

- **DISP_DI_GETMODE command**

Parameter	
command	DISP_DI_GETMODE
data	Not used
object	Not used

Description

It obtains the screen mode of the current window.

The current window is the window whose value is set in the **CurrentWindow** property.

The obtained screen mode value is stored in the data member of the DirectIOData object which is the return value of this method.

The value stored in the *data* member is as follows:

Return value	**Meaning**
DISP_DI_MODE_CHARACTER	The current window is a Character mode window.
DISP_DI_MODE_GRAPHICS	The current window is a Graphic mode window.

- **DISP_DI_GW_STYLE command**

Parameter	
command	DISP_DI_GW_STYLE
data	Specify any of the following. • DISP_DI_GW_NORMAL • DISP_DI_GW_TRANSPARENT
object	Not used

Description

It sets the transparent style for the created Graphic mode window.

As the settings of this command are referred to during the creation of the Graphic mode window, when this command is executed when a Graphic mode window is already present an exception is thrown.

When in the transparent style, the Graphic mode window display is displayed overlapping the display in the background of the Graphic mode window. As it may be impossible to make out overlapping characters, care should be taken in specifying the display positions.

The values that can be set and the functions of the *data* parameter are as follows:

data	**Outline of functions**
DISP_DI_GW_NORMAL	The window will not be created in the transparent style.
DISP_DI_GW_TRANSPARENT	The window will be created in the transparent style.

Default: DISP_DI_GW_NORMAL

3.2.4 ResetStatistics Method

- **Parameter type: *Microsoft.PointOfService.StatisticCategories***

Parameter	
Microsoft.PointOfService.StatisticCategories	
	Specify one of the following. • StatisticCategories.Upos • StatisticCategories.Manufacturer • StatisticCategories.All

Description

Of the items included in the specified category, only the items for which O appeared for the permission reset in "Appendix-E DeviceStatistics" are reset.

All the statistics supported by the ServiceObject are defined in UPOS. If "StatisticCategories.Manufacturer" is specified, nothing is reset.

- **Parameter type: *String[]***

Parameter	
String[]	An array of the item names to reset

Description

Of the items included in the specified category, only the items for which O appears for the reset permission in "Appendix-E DeviceStatistics" are reset when "U_", "M_", or an empty string is specified for item names.

If an illegal item name or non-resettable item name is included, this method reports an error. When this happens, correctly specified items are also not reset.

All the statistics supported by the ServiceObject are defined in UPOS. If "M_" is specified, nothing is reset.

3.2.5 ResetStatistic Method

Description

Of the items included in the specified category, only the items for which O appears for the reset permission in ”Appendix-E DeviceStatistics” are reset when “U_”, “M_”, or an empty string is specified for item names.

If an illegal item name or non-resettable item name is specified, this method reports an error.

All the statistics supported by the ServiceObject are defined in UPOS. If “M_” is specified, nothing is reset.

3.2.6 RetrieveStatistics Method

- **Parameter type: *Microsoft.PointOfService.StatisticCategories***

Parameter

Microsoft.PointOfService.StatisticCategories

Specify one of the following.

- StatisticCategories.Upos
- StatisticCategories.Manufacturer
- StatisticCategories.All

Description

The Statistics supported by ServiceObject are all defined in UPOS. If “StatisticCategories.Manufacturer” is specified, the minimum information specified by UPOS (the 4 items; UPOS version, manufacturer name, device name, and device category) is acquired.

- **Parameter type: *String[]***

Parameter	
String[]	An array of the item names to retrieve

Description

If an illegal item name is included, this method reports an error.

The Statistics supported by ServiceObject are all defined in UPOS. If “M_” is specified, the minimum information specified by UPOS (the 4 items; UPOS version, manufacturer name, device name, and device category) is acquired.

- **Parameter type: None**

Description

The information of all defined items is retrieved.

3.2.7 RetrieveStatistic Method

Description

If an illegal item name is included, this method reports an error.

If multiple item names separated by commas are specified (UPOS Specification), an error is reported.

The Statistics supported by ServiceObject are all defined in UPOS. If “M_” is specified, the minimum information specified by UPOS (the 4 items; UPOS version, manufacturer name, device name, and device category) is acquired.

3.2.8 UpdateStatistics Method

- **Parameter type: *Microsoft.PointOfService.Statistic[]***

Parameter	
Microsoft.PointOfService.Statistic[]	Specifies *Microsoft.PointOfService.Statistic* array for which item names and new values have been set.

Description

Of the items included in the specified category, only the items for which O appears for the update permission in ”Appendix-E DeviceStatistics” are updated when "U_", "M_", or an empty string is specified for item names. If an illegal item name or non-updatable item name is included, this method reports an error. In this case, correctly specified items are also not updated.

The Statistics supported by ServiceObject are all defined by UPOS. If “M_” is specified, nothing is updated.

- **Parameter type: *Microsoft.PointOfService.StatisticCategories***

Parameter	
Microsoft.PointOfService.StatisticCategories	Specify one of the following. •StatisticCategories.Upos •StatisticCategories.Manufacturer •StatisticCategories.All
Object	Specify the new value after updating.

Description

Of the items included in the specified category, only the items for which O appeared for the update permission in ”Appendix-E DeviceStatistics” are updated.

All the statistics supported by the ServiceObject are defined in UPOS. If “StatisticCategories.Manufacturer” is specified, nothing is update.

3.2.9 UpdateStatistic Method

Description

Of the items included in the specified category, only the items for which O appears for the update permission in ”Appendix-E DeviceStatistics” are updated when "U_", "M_", or an empty string is specified for item names. If an illegal item name or non-updatable item name is specified, this method reports an error.

The Statistics supported by ServiceObject are all defined by UPOS. If “M_” is specified, nothing is updated.

3.2.10 CreateWindow method

Description

It can create a new window.

The theoretical size of the window that can be created is as follows:

	Character mode	Graphic mode
No. of rows	50	Value of **MaximumY** property
No. of columns	200	Value of **MaximumX** property

3.2.11 DisplayText method

Description

It can display characters.

The escape sequence (ESC|kC) that enables blinking display is ignored.

The escape sequence (ESC|N) that cancels the display does not cancel blinking display.

3.2.12 DisplayTextAt method

Description

It can display characters starting from specified position.

The escape sequence (ESC|kC) that enables blinking display is ignored.

The escape sequence (ESC|N) that cancels the display does not cancel blinking display.

3.2.13 DefineGlyph method

Description

It performs download registration of a character for the specified character code.

This method performs only definition. When displaying the registered character, the display of character for character codes that are already defined is by the **DisplayText(At)** method.

If a character that is already displayed is being registered, it is not immediately reflected on the display, but the registered content is reflected when the display is refreshed through such method as **RefreshWindow**.

This registered data is valid until the **Close** method is executed.
The registration processing differs depending on the presence of the Graphic mode window as follows:

- **If Graphic mode window is created**

Registration can be done through the DISP_DI_SETFONT command of the **DirectIO** method for each font that can be set. Registration is possible regardless of the font style that is currently set.
The registration data for download registration is managed separately for each font.

During registration, the maximum size that can be created, the range of character codes that can be set, and the maximum registered quantity are as follows:

	DISP_DI_FONT_A	**DISP_DI_FONT_B**
Vertical size	16	7
Horizontal size	8	5
Range of character codes	0x20 - 0x7E	0x20 - 0x7E
Max. registration qty.	95	95

- **If Graphic mode window is not created (includes case of Graphic mode window having been deleted)**

During registration, the data size that can be created is regulated by the properties of **GlyphWidth** and **GlyphHeight**.

3.2.14 ReadCharacterAtCursor method

Description

It reads the character displayed at the current cursor position, and returns the character code.

Processing details differ depending on window mode.

- **If Character mode window**

If character information is not present at the cursor position, it returns 127 (decimal number).

For a multi-byte character, it is returned after being split into the upper byte and the lower byte. The upper byte is returned for the current cursor position, and the lower byte is returned for 1 position after the cursor position.

- **If Graphic mode window**

If character information is not present at the cursor position, it returns 0.

The character information can be obtained only from the upper leftmost position of the character. If acquisition is performed for this position, the character code of the character can be obtained. For other positions, 0 is returned for the character information. For example, if **DisplayTextAt** method is executed for cursor position 0,0 and display character 1 (character code 49), then as the outcome of the execution, 49 is obtainable at the cursor position 0,0 while 0 is obtainable at other cursor positions.

For a multi-byte character, it is returned after being split into the upper byte and the lower byte. The upper byte is returned for the current cursor position, and the lower byte is returned for the cursor position advanced by the character size. Obtaining the character information can be done from the upper leftmost position of each character.

3.2.15 DisplayBitmap method

Description

JPEG files, GIF files, and Windows BMP files can be specified.

This method is valid for the Graphic mode window only. In the Character mode window, an exception is thrown regardless of the setting value in the **CapBitmap** property.

3.2.16 SetBitmap method

Description

JPEG files, GIF files, and Windows BMP files can be specified.

As the image registration is performed by using the download function of the device, the following restrictions apply:

- The image that is on display cannot be reregistered or deleted. For this reason, when the image number on display is set into the *bitmapNumber* parameter, an exception is thrown.
- If the registration region of the device is exceeded during registration, an exception is thrown because image registration is impossible.
- The total size of the region where registration can be made to the device is 4096 bytes.
- The formula for calculating the size used by 1 image is as follows:
  Size = (((No. of vertical dots + 7) / 8) x No. of horizontal dots)

Appendix A Revision history

A.1 EPSON OPOS ADK for .NET 1.14.6

(1) Microsoft POS for .NET 1.14.1 is supported.

A.2 EPSON OPOS ADK for .NET 1.12

(1) Microsoft POS for .NET 1.12 is supported.

A.3 EPSON OPOS ADK for .NET 1.11

(1) Microsoft POS for .NET 1.11 is supported.

A.4 EPSON OPOS ADK for .NET 1.9

(1) Microsoft POS for .NET 1.1 is supported.

A.5 EPSON OPOS ADK for .NET 1.8

POS Device driver complied with Microsoft POS for .NET 1.0 specification has been provided.

This version has been created based on EPSON OPOS ADK 2.40.

The following shows the difference between this version and EPSON OPOS ADK 2.40.

Differences from EPSON OPOS ADK 2.40

(1) The ErrorCode and ErrorCodeExtended that are received as notification of exceptions have been reviewed.

(2) Commands that can be used in the **DirectIO** method are reorganized.

(3) The **DeviceEnabled** property’s status has been deleted from the issuance condition of queued event. As a result, even if the **DeviceEnabled** property's status is false, event notification may still be produced.

(4) When **Open** method is executed, communication control class instance is generated. If communication control class instance cannot be generated, an exception is thrown when the **Open** method is executed.

(5) The Internal and External of the **CheckHealth** method has been changed to not supported.

(6) The theoretical window size that can be created by the **CreateWindow** method has been changed to 50 rows vertically and 200 columns horizontally.

(7) Character codes that are supported have been changed. As a result, 254, 255, and 998 have been deleted from the **CharacterSetList** property.

(8) When **RefreshWindow** method within the marquee initialization mode is executed, the contents that are redrawn on screen have been changed. The change of display contents due to **DisplayText(At)** method or **ClearText** method and such others executed within the marquee initialization mode is reflected to internal data at the time of coming out of the marquee initialization mode. As a result, the changes done through executing **DisplayText(At)** method and such others within the marquee initialization mode are not reflected to the display even if **RefreshWindow** method is executed.

(9) The handling method of the display attribute (Attribute parameter) has been changed.

Even if the display attribute has been changed by **DisplayText(At)** method in other windows, the change to display attribute is not performed. The display attribute used when coming out of the marquee initialization mode is not the display attribute of the **DisplayText(At)** method executed at the end for all windows. The display attribute of the **DisplayText(At)** method executed at the end of the marquee initialization mode window is used.

(10) The image registration method has been changed from the **DISP_DI_SETIMAGE** command of the **DirectIO** method to the **SetBitmap** method. Due to this, the number of images that can be registered is changed from 255 to 100.

(11) The **CapBitmap** property has been changed to true. When creating the Graphic mode window, creation is not possible if the viewport size and the theoretical window size are not vertically and horizontally the same.

(12) The **MaximumX** property has been changed to 256.

(13) The **MaximumY** property has been changed to 64.

(14) The image registration method has been changed to use only the download function method. Due to this, the following functions have been deleted:

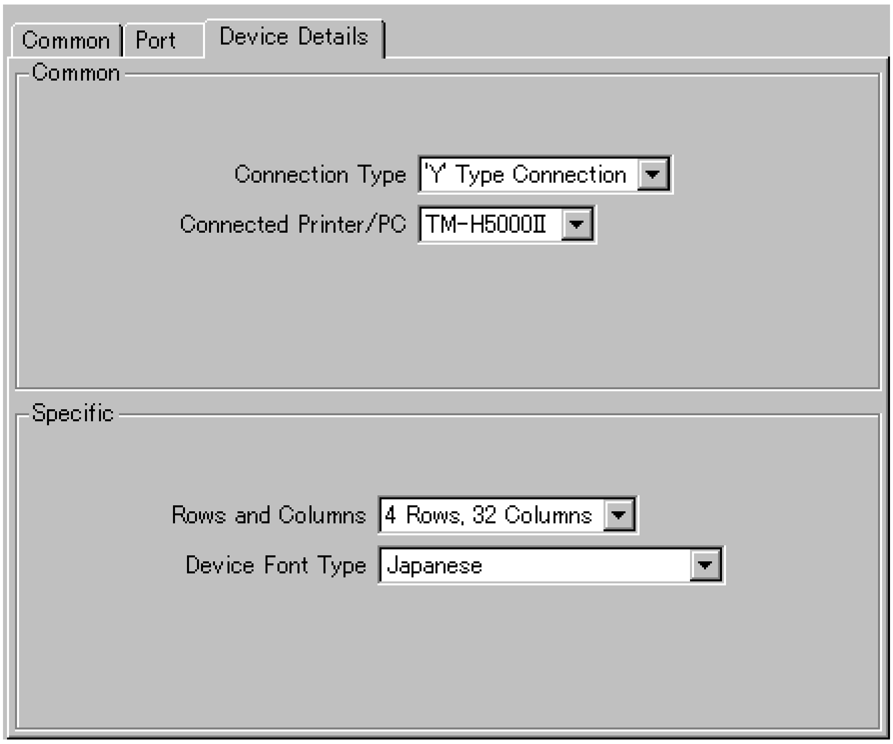
- The function that registers to NVRAM by utilizing the "TMFlash Logo utility".
- The function that displays the image registered n NVRAM.
- The function that deletes image from NVRAM.

(15) GIF and JPEG have been added as image formats that can be displayed or registered.

(16) DISP_DI_SETLINESPACE command has been added to DireciIO method.

This command is used to set the amount of space to shift during a line feed.

Appendix B SetupPOS Settings



Common

Connection Type 'Y' Type Connection

Connected Printer/PC TM-H5000II

Specific

Rows and Columns 4 Rows, 32 Columns

Device Font Type Japanese

B.1 “Connection Type” combobox

Description

It sets the connection type of LineDisplay.

Item	Meaning
Stand Alone	LineDisplay is used as a single unit.
Pass Through	LineDisplay is used by connecting through pass-through to POSPrinter or PC.
‘Y’-Type Connection	LineDisplay is used by connecting through Y-connection to POSPrinter or PC.

- **If set to Pass Through or 'Y'-Type Connection**

  "Connected Printer/PC" combobox is displayed, and printer or PC needs to be set.

Default: Stand Alone

B.2 "Connected Printer/PC" combobox

Description

It sets the POSPrinter or PC used as the host device.

If Pass Through or 'Y'-Type Connection is set in the "Connection Type" combobox, this combobox is displayed.

When using LineDisplay on a USB-connected printer when multiple printers have been registered, select the proper "connected port" in the "Port" configuration window.

Default: TM-H6000III

B.3 "Device Font Type" combobox

Description

Set the multibyte character font for LineDisplay

State	Meaning
Japanese	Set LineDisplay to use the Japanese language.
Simplified Chinese (GB2312)	Set LineDisplay to use the Simplified Chinese (GB2312) language.

Default: Japanese

B.4 “Rows and Columns” combobox

Description

Set the screen mode used by LineDisplay.

State	Meaning
2 Rows, 20 Columns	Set to use 2x20 as the screen mode.
2 Rows, 32 Columns	Set to use 2x32 as the screen mode.
3 Rows, 32 Columns	Set to use 3x32 as the screen mode.
4 Rows, 32 Columns	Set to use 4x32 as the screen mode.
8 Rows, 42 Columns	Set to use 8x42 as the screen mode.

Default: 4 Rows, 32 Columns

Appendix C Hardware Settings

C.1 DIP switch settings

Set the DIP switches of this device as follows:

1) Serial I/F

DIP-SW1

No.	Setting	
1	OFF	Recommended
2	OFF	Fixed as OFF
3	OFF	Fixed as OFF
4	OFF	Note 1
5	OFF	Note 1
6	ON	Note 2
7	OFF	Note 2
8	ON	Note 2

DIP-SW2

No.	Setting	
1	OFF	Fixed as OFF
2	OFF	Fixed as OFF
3	OFF	Fixed as OFF
4	OFF	Fixed as OFF
5	OFF	Fixed as OFF
6	OFF	Fixed as OFF
7	OFF	Fixed as OFF
8	OFF	Fixed as OFF

[Note1] Numbers 4 and 5 of DIP-SW1 set the parity.

DIP Switch 1 Parity Settings

SW No.	Function	ON	OFF	Default
1-4	Parity check	Parity	No parity	OFF
1-5	Parity selection	Even parity	Odd parity	OFF

[Note2] Numbers 6 – 8 of DIP-SW1 set the communication speed.

DIP Switch 1 Transmission Speed Switching

SW1-6	SW1-7	SW1-8	Baud Rate (bps)
ON	ON	ON	2400
OFF	ON	ON	4800
ON	OFF	ON	9600
OFF	OFF	ON	19200
ON	ON	OFF	38400
OFF	ON	OFF	57600
ON	OFF	OFF	115200
OFF	OFF	OFF	(reserved)

2) USB I/F

DIP-SW1

No.	Setting	
1	OFF	Fixed as OFF
2	OFF	Fixed as OFF
3	OFF	Fixed as OFF
4	OFF	Fixed as OFF
5	OFF	Fixed as OFF
6	OFF	Fixed as OFF
7	OFF	Fixed as OFF
8	ON	Fixed as ON

DIP-SW2

No.	Setting	
1	OFF	Fixed as OFF
2	OFF	Fixed as OFF
3	OFF	Fixed as OFF
4	OFF	Fixed as OFF
5	OFF	Fixed as OFF
6	OFF	Fixed as OFF
7	OFF	Fixed as OFF
8	OFF	Fixed as OFF

C.2 Memory switch settings

Set the memory switch to the condition when shipped from factory.

C.3 Jumper switch settings

When using with standalone connection or pass-through connection, set the jumper switches of the option stand as follows:

Connection type	JP1 settings	JP2 settings
Standalone connection	Short 2 and 3	Short 2 and 3
Pass-through connection	Short 1 and 2	Short 1 and 2

Appendix D Default Values of Properties

Initial value and acceptable setting values for each property

D.1 Common settings

Property	Setting value / Defined value	Acceptable setting values
CapCompareFirmwareVersion	false	-
CapPowerReporting	PowerReporting.Standard	-
CapStatisticsReporting	true	-
CapUpdateFirmware	false	-
CapUpdateStatistics	true	-
CheckHealthText	“”	-
Claimed	false	-
DeviceEnabled	false	true, false
FreezeEvents	false	true, false
PowerNotify	PowerNotifycation.Disabled	PowerNotifycation.Disabled, PowerNotifycation.Enabled
PowerState	PowerState.Unknown	-
State	ControlState.Closed	-
DeviceDescription	“EPSON DM-D500 LineDisplay”	-
DeviceName	“DM-D500”	-
CapBlink	DisplayBlink.All	-
CapBitmap	true	-
CapBlinkRate	true	-
CapBrightness	true	-
CapCharacterSet	Refer to “Settings that affect the screen mode”.	Refer to “Settings that affect the screen mode”.
CapCursorType	DisplayCursors.None	-
CapCustomGlyph	true	-
CapDescriptors	false	-
CapHMarquee	true	-
CapICharWait	true	-
CapMapCharacterSet	false	-
CapReadBack	DisplayReadBack.Single	-
CapReverse	DisplayReverse.Each	-

CapScreenMode	false	-
CapVMarquee	false	-
BlinkRate	1600	1 and above
CharacterSet	Refer to “Settings that affect the change of language”.	Refer to “Settings that affect the change of language”.
CharacterSetList	Refer to “Settings that affect the change of language”.	Refer to “Settings that affect the change of language”.
Columns	Refer to “Settings that affect the screen mode”.	Refer to “Settings that affect the screen mode”.
CurrentWindow	0	0 – 4
CursorColumn	0	Within the range of 0 to Columns Property
CursorRow	0	Within the range of 0 to (Rows Property - 1)
CursorType	DisplayCursors.None	-
CursorUpdate	true	true, false

CustomGlyphList	RangeOfCharacters[1]
	RangeOfCharacters[0].From = 0x20
	RangeOfCharacters[0].To = 0x7e
		-

DeviceBrightness	100	0 – 100
DeviceColumns	Refer to “Settings that affect the screen mode”.	Refer to “Settings that affect the screen mode”.
DeviceDescriptors	0	-
DeviceRows	Refer to “Settings that affect the screen mode”.	Refer to “Settings that affect the screen mode”.
DeviceWindows	4	-
GlyphHeight	Refer to “Settings that affect the screen mode”.	Refer to “Settings that affect the screen mode”.
GlyphWidth	Refer to “Settings that affect the screen mode”.	Refer to “Settings that affect the screen mode”.
InterCharacterWait	0	0 and above
MapCharacterSet	false	-
MarqueeFormat	DisplayMarqueeFormat.Walk	DisplayMarqueeFormat.Walk, DisplayMarqueeFormat.Place
MarqueeRepeatWait	0	0 and above
MarqueeType	DisplayMarqueeType.None	DisplayMarqueeType.None, DisplayMarqueeType.Init, DisplayMarqueeType.Left, DisplayMarqueeType.Right
MarqueeUnitWait	0	0 and above
MaximumX	256	-
MaximumY	64	-
Rows	Refer to “Settings that affect the screen mode”.	Refer to “Settings that affect the screen mode”.
ScreenMode	0	0
ScreenModeList	Refer to “Settings that affect the screen mode”.	Refer to “Settings that affect the screen mode”.

D.2 Settings that affect the change of language

Language	Property	Setting value / Defined value	Acceptable setting values
Japanese	CharacterSet	932	Any of CharacterSetList

	CharacterSetList	int[11]
		int[0] = 437
		int[1] = 850
		int[2] = 852
		int[3] = 858
		int[4] = 860
		int[5] = 863
		int[6] = 865
		int[7] = 866
		int[8] = 932
		int[9] = 999
		int[10] = 1252
			-

Simplified Chinese (GB2312)	CharacterSet	437	Any of CharacterSetList

	CharacterSetList	int[11]
		int[0] = 437
		int[1] = 850
		int[2] = 852
		int[3] = 858
		int[4] = 860
		int[5] = 863
		int[6] = 865
		int[7] = 866
		int[8] = 936
		int[9] = 999
		int[10] = 1252
			-

D.3 Settings that affect the screen mode

Screen mode	Property	Setting value / Defined value	Acceptable setting values
2x20	CapCharacterSet	CharacterSetCapability.Kana	-
	Columns	20	-
	DeviceColumns	20	-
	DeviceRows	2	-
	GlyphHeight	16	-
	GlyphWidth	8	-
	Rows	2	-

	ScreenModeList	DisplayScreemMode[1]
		DisplayScreemMode[0].Row = 2
		DisplayScreemMode[0].Column = 20
			-

4x32	CapCharacterSet	CharacterSetCapability.Kanji	-
	Columns	32	-
	DeviceColumns	32	-
	DeviceRows	4	-
	GlyphHeight	16	-
	GlyphWidth	8	-
	Rows	4	-

	ScreenModeList	DisplayScreemMode[1]
		DisplayScreemMode[0].Row = 4
		DisplayScreemMode[0].Column = 32
			-

3x32	CapCharacterSet	CharacterSetCapability.Kanji	-
	Columns	32	-
	DeviceColumns	32	-
	DeviceRows	3	-
	GlyphHeight	16	-
	GlyphWidth	8	-
	Rows	3	-

	ScreenModeList	DisplayScreemMode[1]
		DisplayScreemMode[0].Row = 3
		DisplayScreemMode[0].Column = 32
			-

2x32	CapCharacterSet	CharacterSetCapability.Kanji	-
	Columns	32	-
	DeviceColumns	32	-
	DeviceRows	2	-
	GlyphHeight	16	-
	GlyphWidth	8	-
	Rows	2	-

	ScreenModeList	DisplayScreemMode[1]
		DisplayScreemMode[0].Row = 2
		DisplayScreemMode[0].Column = 32
			-

8x42	CapCharacterSet	CharacterSetCapability.Kana	-
	Columns	42	-
	DeviceColumns	42	-
	DeviceRows	8	-
	GlyphHeight	7	-
	GlyphWidth	5	-
	Rows	8	-

	ScreenModeList	DisplayScreemMode[1]
		DisplayScreemMode[0].Row = 8
		DisplayScreemMode[0].Column = 42
			-

Appendix E DeviceStatistics

The summary for the Statistics function of the device is shown.

XML definition name	Description	Reset Permission	Update Permission
UnifiedPOSVersion	UPOS version	×	×
DeviceCategory	Device category	×	×
ManufactureName	Manufacturer	×	×
ModelName	Device name	×	×
SerialNumber	Serial number	×	×
ManufactureDate	Manufacturing date	×	×
MechanicalRevision	Device revision	×	×
FirmwareRevision	Firmware version	×	×
Interface	Interface	×	×
InstallationDate	Installation date	×	×
HoursPoweredCount	Length of time in operation	O	O
CommunicationErrorCount	Communication error count	O	O
OnlineTransitionCount	Number of times displayed	O	O

O:Permitted
×:Not permitted