

EPSON ESCPOS provider

Version 1.0.0

User's guide

August 22 2017

【 remarks 】

This document is translated from Japanese into English by the machine translation.

【 revision history 】

Number of versions	Date	Content
1.0.0	2017-8-22	First edition.

Contents

1. Introduction	6
2. Outline of provider	7
2.1. Outline.....	7
2.2. Installation.....	7
2.3. Method property.....	8
2.3.1. CaoWorkspace::AddController method	8
2.3.1.1. Conn option.	10
2.3.2. CaoController::Execute method.....	10
2.3.3. CaoController::AddVariable method	10
3. Command list	12
3.1. Print instruction command	15
3.1.1. CaoController::Execute("PrintLF") command	15
3.1.2. CaoController::Execute("PrintFeedPaper") command	15
3.1.3. CaoController::Execute("PrintFeedLine") command	16
3.2. Changing line amount command.....	16
3.2.1. CaoController::Execute("ResetLineSpacing") command	16
3.2.2. CaoController::Execute("SetLineSpacing") command	16
3.3. Print character command	17
3.3.1. CaoController::Execute("SetRightSideCharacterSpacing") command	17
3.3.2. CaoController::Execute("SetPrintMode") command.....	17
3.3.3. CaoController::Execute("SetUnderLineMode") command	18
3.3.4. CaoController::Execute("SetEmphasizedMode") command	18
3.3.5. CaoController::Execute("SetDoubleStrikeMode") command	19
3.3.6. CaoController::Execute("SetFont") command	19
3.3.7. CaoController::Execute("SetInternationalCharacter") command	20
3.3.8. CaoController::Execute("SetRightRotate") command	21
3.3.9. CaoController::Execute("SetCharCodeTable") command	21
3.3.10. CaoController::Execute("SetSMModeUpsideDownMode") command	23
3.3.11. CaoController::Execute("SetFontSize") command	23
3.3.12. CaoController::Execute("SetWhiteBlackReverseMode") command	24
3.3.13. CaoController::Execute("SetSmoothingMode") command.....	24
3.4. Print form command	25
3.4.1. CaoController::Execute("GetPositioningInfo") command	25
3.4.2. CaoController::Execute("FeedPaperToPeelingPosition") command.....	26
3.4.3. CaoController::Execute("FeedPaperToCuttingPosition") command	26
3.4.4. CaoController::Execute("FeedPaperToStartingPosition") command	27

3.5. Print position command	27
3.5.1. CaoController::Execute("MoveHT") command	27
3.5.2. CaoController::Execute("MoveA") command	28
3.5.3. CaoController::Execute("SetHTPosition") command	28
3.5.4. CaoController::Execute("MoveR") command	29
3.5.5. CaoController::Execute("SetSModeAlignment") command	30
3.5.6. CaoController::Execute("SetSModeLeftMargin") command	30
3.5.7. CaoController::Execute("MoveSModeBeginningLine") command	31
3.5.8. CaoController::Execute("SetSModePrintAreaWidth") command	31
3.6. Mechanism control command.....	32
3.6.1. CaoController::Execute("CutPaper") command	32
3.7. Chinese character control command.....	33
3.7.1. CaoController::Execute("SetKanjiSetting") command.....	33
3.7.2. CaoController::Execute("StartKanjiMode") command.....	34
3.7.3. CaoController::Execute("SetKanjiUnderLineMode") command	34
3.7.4. CaoController::Execute("StopKanjiMode") command	35
3.7.5. CaoController::Execute("SetKanjiCodeSystem") command	35
3.7.6. CaoController::Execute("SetKanjiSpacing") command	35
3.7.7. CaoController::Execute("SetKanjiQuadrupleSizeMode") command	36
3.7.8. CaoController::Execute("SetKanjiFont") command.....	36
3.8. Supplementary functional command.....	37
3.8.1. CaoController::Execute("RTMPowerOff") command.....	37
3.8.2. CaoController::Execute("RTMClearBuffer") command	37
3.8.3. CaoController::Execute("InitializePrinter") command.....	38
3.8.4. CaoController::Execute("TestPrint") command.....	38
3.8.5. CaoController::Execute("GetPrinterID") command	39
3.8.6. CaoController::Execute("SetBasePitch") command.....	40
3.9. Original command.....	40
3.9.1. CaoController::Execute("Send") command	40
3.9.2. CaoController::Execute("SendAndReceive") command.....	41
3.9.3. CaoController::Execute("AddPrint") command.....	41
3.9.4. CaoController::Execute("AddPrint_JP") command	42
3.9.5. CaoController::Execute("PrintLF_JP") command.....	43
3.9.6. CaoController::Execute("AddPrint_EN") command.....	43
3.9.7. CaoController::Execute("PrintLF_EN") command.....	44
3.9.8. CaoController::Execute("AddPrint_DE") command.....	44
3.9.9. CaoController::Execute("PrintLF_DE") command.....	45

4. Variable list.....	47
4.1.1. System variable.....	47
5. Error code	48

1. Introduction

¹²The EPSON ESCPOS provider (henceforth ESCPOS provider) is ORiN2 CAO provider that controls the printer through the ESC/POS command. If it is a model that corresponds to the ESC/POS command, it is possible to control regardless of the manufacturer.

In this document, it explains the outline of the ESCPOS provider and the mounted CAO interface (function specification).

¹ ESC/POS is a registered trademark of Seiko Epson Ltd..

² FP-32L-LAN is used for confirming the operation.

2. Outline of provider

2.1. Outline

The ESCPOS provider is offering the method by `CaoController::Execute` as a method of executing the command. In `CaoController::Execute`, the ESC/POS command is sent and received by using Ethernet and the serial interface.

2.2. Installation

The ESCPOS provider module is composed of following DLL. The installation work is unnecessary when installing it with the installer of ORiN2 SDK. When you install it by hand powerTable2-1Please ..[noyou].. execute it.

Table2-1ESCPOS provider

File name	CaoProv.EPSON.ESCPOS.dll
ProgID	CaoProv. EPSON.ESCPOS
Registry registration	regsvr32 CaoProvEPSON.ESCPOS.dll
Blotting out of registry registration	regsvr32 /u CaoProvEPSON.ESCPOS.dll

2.3. Method property

2.3.1. CaoWorkspace::AddController method

The communication is connected in the ESCPOS provider referring to connected parameter for the communication at AddController. At this time, the communication form and the time-out are specified in the option.

Format AddController (< bstrCtrlName:VT_BSTR > and < bstrProvName:VT_BSTR >

<bstrPcName:VT_BSTR > [,<bstrOption:VT_BSTR>])

bstrCtrlName : The in controller name is arbitrary.
 bstrProvName : provider name fixation value = "CaoProv.EPSON.ESCPOS"
 bstrPcName : Execution machine name of in provider
 bstrOption : In optional character string

The list specified for an optional character string is shown as follows.

Table2-2Optional character string of CaoWorkspace::AddController

Option	Meaning
Conn =< connected parameter >	Indispensability. A communication form and the connected parameter are set. Please refer to 2.3.1.1.
MyIP =[<local IP address >[: local port number]]	Arbitrariness. NIC can be selected by specifying Internet Protocol address in this.. option when two or more NIC is used. It is automatically selected when omitting it. When Internet Protocol address not allocated in a local machine is specified, the error is returned.
Timeout =< timeout period >	Arbitrariness. The timeout period when sending and receiving it is specified. (millisecond) (default: 3000)
ConnTimeout =< timeout period >	Arbitrariness. Timeout period when TCP is connected. (millisecond) (default: 500)
KeepAlive =< Keep alive time >	Arbitrariness. The keep alive time when TCP is communicated is set. (millisecond) (default: 1000)

2.3.1.1. Conn option.

Connected parameter character string of optional Conn is shown as follows. A possible omission is shown here in the square bracket (""). Moreover, the underlined part under the explanation of each parameter shows the default value when the option is not specified.

Ethernet device

"Conn=ETH:<IP Address> [:Port No>]"

"Conn=UDP:<IP Address> [:Port No>]"³

<IP Address> : Internet Protocol address of connected equipment.

Example:"127.0.0.1","192.168.0.1"

<Port No> : Connected port number. 9100,9101... It is possible to specify it voluntarily.

RS-232C device

"Conn=COM:<COM Port>[:<BaudRate>[:<Parity>:<DataBits>:<StopBits>[:<Flow>]]]"³

<COM Port> : COM port number. '1'-COM1,'2'-COM2, ...

<BaudRate> : Transmission rate. 2400, 4800,9600,19200,38400,57600,115200.

<Parity> : Parity. 'N'-NONE,'E'-EVEN,'O'-ODD

<DataBits> : Number of data bits. '7'-7bit,'8'-8bit.

<StopBits> : Number of stop bits. '1'-1bit,'2'-2bit.

<Flow> : Flow control. '1'..Xon/Xoff.. '2'-- hardware control.

It is possible to specify it by taking OR.

(default: Flow control '0'-none)

2.3.2. CaoController::Execute method

The Execute method of the CaoController class is a method for the execution of the command. The command name is specified in the first argument and the parameter of the command is specified for the second argument. Please refer to the chapter 3.

Format Execute(<bstrCommandName:VT_BSTR> [, vntParam:VT_VARIANT])

BstrCommandName: In command name

vntParam : In parameter

2.3.3. CaoController::AddVariable method

The AddVariable method of the CaoController class is a method for making the variable object. The variable list that can be used. Please refer to the chapter4.

Format AddVariable (<bstrCommandNane>[, <bstrOption>])

³ It becomes outside the support range though it confirms to the place where the UDP communication and the RS-232C communication can be connected because it doesn't confirm the operation with a real machine.

bstrVariableName : In variable identifier

bstrOption : In optional character string

3. Command list

This chapter explains each command of the `CaoController::Execute` method. The command list that corresponds in the ESCPOS provider is shown as follows.

Please refer to the command reference for detailed operation of each command.

The method of offering the command reference is different depending on the model. Prescribed procedure after model used is confirmed, please.

Table3-1CaoController::Execute command list

ESC/POS Command	Command	Function	Page
Print instruction command			
LF	PrintLF	Print changing line	P. 15
ESC J	PrintFeedPaper	Print and paper feed	P. 15
ESC d	PrintFeedLine	Paper feed of print and n line	P. 16
Changing line amount command			
ESC 2	ResetLineSpacing	It sets it to the amount of an initial changing line.	P. 16
ESC 3	SetLineSpacing	Setting of amount of changing line	P. 16
Print character command			
ESC SP	SetRightSideCharacterSpacing	Setting of amount of character of right space	P. 17
ESC !	SetPrintMode	Batch specification of print mode	P. 17
ESC -	SetUnderLineMode	Specification and release of underline	P. 18
ESC E	SetEmphasizedMode	Specification and release of enhanced print character	P. 18
ESC G	SetDoubleStrikeMode	Specification and release of double print	P. 19
ESC M	SetFont	Selection of character font	P. 19
ESC R	SetInternationalCharacter	Selection of international character	P. 20
ESC V	SetRightRotate	90-degree specification and release of right rotation of character	P. 21
ESC t	SetCharacterCodeTable	Selection of character-code table	P. 21
ESC {	SetSModeUpsideDownMode	Specification and release of headstand print	P. 23
GS !	SetFontSize	Specification of font size	P. 23
GS B	SetWhiteBlackReverseMode	Specification and release of monochrome reversing print	P. 24
GS b	SetSmoothingMode	Specification and release of smoothing	P. 24
Print form command			
FS (L <fn=48>	GetPositioningInfo	Reception of location information	P. 25
FS (L <fn=65>	FeedPaperToPeelingPosition	Paper feed to flaking off position	P. 26
FS (L <fn=66>	FeedPaperToCuttingPosition	Paper feed to cutting position	P. 26

FS (L <fn=67>	FeedPaperToStartingPosition	Paper feed to head putting out position	P. 27
Print position command			
HT	MoveHT	Horizontal tab	P. 27
ESC \$	MoveA	Specification of absolute position	P. 28
ESC D	SetHTPosition	Setting of the horizontal tab position	P. 28
ESC ¥	MoveR	Specification of relative position	P. 29
ESC a	SetSMModeAlignment	Positional arrangement	P. 30
GS L	SetSMModeLeftMargin	Specification of left margin	P. 30
GS T	MoveSMModeBeginningLine	Movement of print position to head of line	P. 31
GS W	SetSMModePrintAreaWidth	Setting of width of print area	P. 31
Mechanism control command			
GS V	CutPaper	Cutting of form	P. 32
Chinese character control command			
FS !	SetKanjiSetting	Batch specification of print mode of Chinese character	P. 33
FS &	StartKanjiMode	Beginning of Chinese character mode	P. 34
FS -	SetKanjiUnderLineMode	Specification and release of Chinese character underline	P. 34
FS .	StopKanjiMode	End of Chinese character mode	P. 35
FS C	SetKanjiCodeSystem	Setting of Kanji character code system	P. 35
FS S	SetKanjiSpacing	Setting of amount of Chinese character space	P. 35
FS W	SetKanjiQuadrupleSizeMode	Specification and release of four time Chinese character corner character	P. 36
FS (A <fn=48>	SetKanjiFont	Selection of Chinese font	P. 36
Supplementary functional command			
DLE DC4 <fn=2>	RTMPowerOff	Execution of power-off processing	P. 37
DLE DC4 <fn=8>	RTMClearBuffer	Buffer clearness	P. 37
ESC @	InitializePrinter	Initialization of printer	P. 38
GS (A	TestPrint	Execution of test print	P. 38
GS I	GetPrinterID	Printer ID acquisition	P. 39
GS P	SetBasePitch	Setting of basic calculation pitch	P. 40
Original command			
-	Send	The specified command format is transmitted.	P. 40
-	SendAndReceive	The specified command format is transmitted, and the response is received.	P. 41
-	AddPrint	The character string is stored in the print buffer.	P. 41
-	AddPrint_JP	A Japanese character string is stored in the print buffer.	P. 42
-	PrintLF_JP	Japanese print changing line	P. 43
-	AddPrint_EN	An English character string is stored in the print buffer.	P. 43
-	PrintLF_EN	English print changing line	P. 44
-	AddPrint_DE	A German character string is stored in the print buffer.	P. 44

-	PrintLF_DE	German print changing line	P. 45
---	------------	----------------------------	-------

3.1. Print instruction command

3.1.1. **GaoController::Execute("PrintLF")** command

The print of data in the print buffer and the paper feed of one line are executed.

Please printout contents can be specified for the parameter, and specify it by byte array or the character string.

Format PrintLF ([<vntDataArray>])

1

<vntDataArray> : In printout contents (byte array specification) (VT_UI1 | VT_ARRAY)

Return value : None

Format PrintLF ([<bstrData>])

2

<bstrData> : In printout contents (character string specification) (VT_BSTR)

Return value : None

Usage

example

```
Dim vntDataArray As Variant

vntDataArray = Array(&h31, &h32, &h33, &h34, &h35) ' 12345
ctrl.Execute "PrintLF", vntDataArray
ctrl.Execute "PrintLF", "12345"
```

3.1.2. **GaoController::Execute("PrintFeedPaper")** command

The paper feed of the print of data in the print buffer and byteLine × basis calculation pitch is executed.

Format PrintFeedPaper (<byteLine>)

<byteLine> : Amount of in paper feed (VT_UI1)

Return value : None

Usage

example

```
ctrl.Execute "PrintFeedPaper", 100
```

3.1.3. CaoController::Execute("PrintFeedLine") command

The print of data in the print buffer and the paper feed of the byteLine line are executed.

Format PrintFeedPaper (<byteLine>)

<byteLine> : The in paper feed number of lines (VT_UI1)
Return value : None

Usage

example

```
ctrl.Execute "PrintFeedLine", 1
```

3.2. Changing line amount command

3.2.1. CaoController::Execute("ResetLineSpacing") command

The amount of changing line is set to "Amount of an initial changing line".

Format ResetLineSpacing ()

Argument : None
Return value : None

Usage

example

```
ctrl.Execute "SelectDefaultLineSpacing"
```

3.2.2. CaoController::Execute("SetLineSpacing") command

The amount of changing line is set in byteData × basis calculation pitch.

Format SetLineSpacing (<byteData>)

<byteData> : Amount of in changing line (VT_UI1)
Return value : None

Usage

example

```
ctrl.Execute "SetLineSpacing", 100
```

3.3. Print character command**3.3.1. CaoController::Execute("SetRightSideCharacterSpacing") command**

The amount of the character of right space is set to $\text{byteData} \times \text{basis calculation pitch}$.

Format SetRightSideCharacterSpacing (<byteData>)

<byteData> : Amount of in right space (VT_UI1)

Return value : None

Usage**example**

```
ctrl.Execute "SetRightSideCharacterSpacing", 100
```

3.3.2. CaoController::Execute("SetPrintMode") command

The print mode is specified in the lump.

<i>ByteData : Bit</i>	Binary number	Function	Hexadeci mal number	Decimal number
0	0	Font 1 is selected.	00	0
	1	Font 2 is selected.	01	1
1	-	(reservation)	-	-
2	-	(reservation)	-	-
3	0	The enhanced print character is released.	00	0
	1	The enhanced print character is specified.	08	8
4	0	The length twice expansion is released.	00	0
	1	The length twice expansion is specified.	10	16
5	0	The horizontal twice expansion is specified.	00	0
	1	The horizontal twice expansion is released.	20	32
6	-	(reservation)	-	-
7	0	The addition of the underline is released.	00	0
8	1	The addition of the underline is specified.	80	128

Format SetPrintMode (<byteData>)

<byteData> : In print mode (VT_UI1)
Return value : None

Usage

example

```
ctrl.Execute "SetPrintMode", 1
```

3.3.3. CaoController::Execute("SetUnderLineMode") command

The underline is specified and released.

<i>byteData</i>	Function
0,48	The addition of the underline is released.
1,49	The addition of the setting of the underline of the width of one dot and the underline is specified.
2,50	The addition of the setting of the underline of the width of two dots and the underline is specified.

Format SetUnderLineMode (<byteData>)

<byteData> : In underline setting (VT_UI1)
Return value : None

Usage

example

```
ctrl.Execute "SetUnderLineMode", 1
```

3.3.4. CaoController::Execute("SetEmphasizedMode") command

The enhanced print character is specified and released.

When the least significant bit of byteData is 0, the enhanced print character is released.

When the least significant bit of byteData is one, the enhanced print character is specified.

Format SetEmphasizedMode (<byteData>)

<byteData> : In enhanced print character setting (VT_UI1)

Return value : None

Usage

example

```
ctrl.Execute "SetEmphasizedMode", 1
```

3.3.5. CaoController::Execute("SetDoubleStrikeMode") command

A double print is specified and released.

When the least significant bit of byteData is 0, a double print is released.

When the least significant bit of byteData is one, a double print is specified.

Format SetDoubleStrikeMode (<byteData>)

<byteData> : In double print setting (VT_UI1)

Return value : None

Usage

example

```
ctrl.Execute "SetDoubleStrikeMode", 1
```

3.3.6. CaoController::Execute("SetFont") command

The character font is selected.

<i>byteData</i>	Font
0,48	Font A
1,49	Font B
2,50	Font C
3,51	Font D
4,52	Font E
97	Special font A
98	Special font B

Format SetFont (<byteData>)

<byteData> : In character font (VT_UI1)

Return value : None

Usage**example**

```
ctrl.Execute "SetFont", 1
```

3.3.7. CaoController::Execute("SetInternationalCharacter") command

An international character is selected.

<i>byteData</i>	Name of a country
0	The United States
1	France
2	Germany
3	Britain
4	Denmark I
5	Sweden
6	Italy
7	Spain I
8	Japan
9	Norway
10	Denmark II
11	Spain II
12	Latin America
13	South Korea
14	Slovene/Croatia
15	China
16	Vietnam
17	Arabia
<i>byteData</i>	Name of a country (language)
66	India (Devanagari)
67	India (Bengali)
68	India (Tamil)
69	India (Telugu)
70	India (Assamese)
71	India (Oriya)
72	India (Kannada)
73	India (Malayalam)
74	India (Gujarati)
75	India (Punjabi)
82	India (Marathi)

Format SetInternationalCharacter (<byteData>)

<byteData> : Set of character in each country of in (VT_UI1)

Return value : None

Usage**example**

```
ctrl.Execute "SetInternationalCharacter", 0
```

3.3.8. CaoController::Execute("SetRightRotate") command

A right line is specified and the character is released 90 times.

<i>byteData</i>	Function
0,48	The right rotation character is released 90 times.
1,49	The right rotation character (one dot of the method of side interval) is specified 90 times.
2,50	The right rotation character (1.5 horizontal dot interval) is specified 90 times.

Format SetRightRotate (<byteData>)

<byteData> : Rotation specification of in character (VT_UI1)

Return value : None

Usage**example**

```
ctrl.Execute "SetRightRotate", 1
```

3.3.9. CaoController::Execute("SetCharacterCodeTable") command

The character-code table is selected.

<i>byteData</i>	Character kind of selection page
0	..page.. 0 PC437: USA and Standard Europe
1	..page.. 1 Katakana
2	..page.. 2 PC850: Multilingual
3	..page.. 3 PC860: Portuguese
4	..page.. 4 PC863: Canadian-French
5	..page.. 5 PC865: Nordic
6	..page.. 6 Hiragana and simple Chinese character
7	..page.. 7 Simple Chinese character
8	..page.. 8 Simple Chinese character
11	..page.. 11 PC851: Greek
12	..page.. 12 PC853: Turkish
13	..page.. 13 PC857: Turkish
14	..page.. 14 PC737: Greek
15	..page.. 15 ISO8859-7: Greek

16	..page.. 16 WPC1252
17	..page.. 17 PC866: Cyrillic #2
18	..page.. 18 PC852: Latin 2
19	..page.. 19 PC858: Euro
20	..page.. 20 Thai Character Code 42
21	..page.. 21 Thai Character Code 11
22	..page.. 22 Thai Character Code 13
23	..page.. 23 Thai Character Code 14
24	..page.. 24 Thai Character Code 16
25	..page.. 25 Thai Character Code 17
26	..page.. 26 Thai Character Code 18
30	..page.. 30 TCVN-3: Vietnamese
31	..page.. 31 TCVN-3: Vietnamese
32	..page.. 32 PC720: Arabic
33	..page.. 33 WPC775: Baltic Rim
34	..page.. 34 PC855: Cyrillic
35	..page.. 35 PC861: Icelandic
36	..page.. 36 PC862: Hebrew
37	..page.. 37 PC864: Arabic
38	..page.. 38 PC869: Greek
39	..page.. 39 ISO8859-2: Latin 2
40	..page.. 40 ISO8859-15: Latin 9
41	..page.. 41 PC1098: Farsi
42	..page.. 42 PC1118: Lithuanian
43	..page.. 43 PC1119: Lithuanian
44	..page.. 44 PC1125: Ukrainian
45	..page.. 45 WPC1250: Latin 2
46	..page.. 46 WPC1251: Cyrillic
47	..page.. 47 WPC1253: Greek
48	..page.. 48 WPC1254: Turkish
49	..page.. 49 WPC1255: Hebrew
50	..page.. 50 WPC1256: Arabic
51	..page.. 51 WPC1257: Baltic Rim
52	..page.. 52 WPC1258: Vietnamese
53	..page.. 53 KZ-1048: Kazakhstan
66	..page.. 66 Devanagari
67	..page.. 67 Bengali
68	..page.. 68 Tamil
69	..page.. 69 Telugu
70	..page.. 70 Assamese
71	..page.. 71 Oriya
72	..page.. 72 Kannada
73	..page.. 73 Malayalam
74	..page.. 74 Gujarati
75	..page.. 75 Punjabi
82	..page.. 82 Marathi
254	..page.. 254
255	..page.. 255

Format SetCharacterCodeTable (<byteData>)

<byteData> : Page number in in character-code table (VT_UI1)
 Return value : None

Usage**example**

```
ctrl.Execute "SetCharacterCodeTable", 0
```

3.3.10. CaoController::Execute("SetSMModeUpsideDownMode") command

The headstand print is specified and released.

When the least significant bit of byteData is 0, the headstand print is released.

When the least significant bit of byteData is one, the headstand print is specified.

Format SetSMModeUpsideDownMode (<byteData>)

<byteData> : In headstand print setting (VT_UI1)
 Return value : None

Usage**example**

```
ctrl.Execute "SetSMModeUpsideDownMode", 1
```

3.3.11. CaoController::Execute("SetFontSize") command

The font size is set.

A horizontal magnification of the character is specified by bits 6, 5, and 4 of byteData, and the vertical direction magnification of the character is specified by bits 2, 1, and 0 (0xxx0xxxb).

Specification of vertical direction magnification Specification of horizontal magnification

Bit			Hexadecimal number	Decimal number	Magnification
6	5	4			
0	0	0	00	0	One time (standard)
0	0	1	10	16	Twice (side twice)
0	1	0	20	32	Three times
0	1	1	30	48	Four times

Bit			Hexadecimal number	Decimal number	Magnification
6	5	4			
0	0	0	00	0	One time (standard)
0	0	1	01	1	Twice (length twice)
0	1	0	02	2	Three times
0	1	1	03	3	Four times

1	0	0	40	64	Five times
1	0	1	50	80	Six times
1	1	0	60	96	Seven times
1	1	1	70	112	Eight times

1	0	0	04	4	Five times
1	0	1	05	5	Six times
1	1	0	06	6	Seven times
1	1	1	07	7	Eight times

Format SetFontSize (<byteData>)

<byteData> : In font size setting (VT_UI1)

Return value : None

Usage

example

```
ctrl.Execute "SetFontSize", 18      Three twice in length ... times in width
```

3.3.12. CaoController::Execute("SetWhiteBlackReverseMode") command

The monochrome reversing print is specified and released.

When the least significant bit of byteData is 0, the monochrome reversing print is released.

When the least significant bit of byteData is one, the monochrome reversing print is specified.

Format SetWhiteBlackReverseMode (<byteData>)

<byteData> : In monochrome reversing print setting (VT_UI1)

Return value : None

Usage

example

```
ctrl.Execute "SetWhiteBlackReverseMode", 1
```

3.3.13. CaoController::Execute("SetSmoothingMode") command

Smoothing is specified and released.

When the least significant bit of byteData is 0, smoothing is released.

When the least significant bit of byteData is one, smoothing is specified.

Format SetSmoothingMode (<byteData>)

<byteData> : In smoothing setting (VT_UI1)

Return value : None

Usage**example**

```
ctrl.Execute "SetSmoothingMode", 1
```

3.4. Print form command**3.4.1. CaoController::Execute("GetPositioningInfo") command**

The location information of the label paper and the [burakkuma-ku] paper is received.

•Location information A becomes the next table.

Bit	Binary number	Function	Hexadecimal number	Decimal number
0	0	It is not a flaking off position.	00	0
	1	It stands by to the flaking off position.	01	1
1	0	It is not a cutting position.	00	0
	1	It stands by to the cutting position.	02	2
2	0	It is not a head putting out position.	00	0
	1	It stands by to the head putting out position.	04	4
3 ~ 5	-	(reservation)	-	-
6	1	Fixation	40	64
7	0	Fixation	00	0

•Location information B becomes the next table.

Bit	Binary number	Function	Hexadecimal number	Decimal number
0	0	The head putting out of the label can operate now.	00	0
	1	The head putting out operation of the label is impossible now.	01	1
1	0	The head putting out of the next label can operate.	00	0
	1	The head putting out operation of the next label is impossible.	02	2
2 ~ 5	-	(reservation)	-	-
6	1	Fixation	40	64
7	0	Fixation	00	0

Format GetPositioningInfo ()

Argument : None

Return value : The out location information is returned (VT_UI1 | VT_ARRAY).
0:Location information A
1:Location information B

Usage**example**

```
Dim vntResult as Variant
vntResult = ctrl.Execute "GetPositioningInfo"
```

3.4.2. CaoController::Execute("FeedPaperToPeelingPosition") command

Paper is sent to the flaking off position.

<i>byteData</i>	Function
48	The paper feed to the following flaking off position is executed. However, it doesn't operate when standing by to the flaking off position.
49	The paper feed to the following flaking off position is executed. However, the paper feed to the following flaking off position is executed further when standing by to the flaking off position.

Format FeedPaperPeelingPosition (<byteData>)

<byteData> : In paper feed mode (VT_UI1)

Return value : None

Usage**example**

```
ctrl.Execute "FeedPaperPeelingPosition", 48
```

3.4.3. CaoController::Execute("FeedPaperToCuttingPosition") command

Paper is sent to the cutting position.

<i>byteData</i>	Function
48	The paper feed to the following cutting position is executed. However, it doesn't operate when standing by to the cutting position.
49	The paper feed to the following cutting position is executed. However, the paper feed to the following cutting position is executed further when standing

	by to the cutting position.
--	-----------------------------

Format FeedPaperToCuttingPosition (<byteData>)

<byteData> : In paper feed mode (VT_UI1)

Return value : None

Usage

example

```
ctrl.Execute "FeedPaperToCuttingPosition", 48
```

3.4.4. CaoController::Execute("FeedPaperToStartingPosition") command

Paper is sent to the head putting out position.

<i>byteData</i>	Function
48	The paper feed to the head putting out position of the next label is executed. However, it doesn't operate when standing by to the head putting out position.
49	The paper feed to the following head putting out position is executed. However, the paper feed to the following head putting out position is executed further when standing by to the head putting out position.
50	The paper feed to the head putting out position of the label is executed now. However, it doesn't operate when standing by to the head putting out position.

Format FeedPaperStartingPosition (<byteData>)

<byteData> : In paper feed mode (VT_UI1)

Return value : None

Usage

example

```
ctrl.Execute "FeedPaperStartingPosition", 48
```

3.5. Print position command

3.5.1. CaoController::Execute("MoveHT") command

The print position is moved to the following horizontal tab.

Format MoveHT ()

Argument : None
Return value : None

Usage**example**

```
ctrl.Execute "MoveHT"
```

3.5.2. CaoController::Execute("MoveA") command

Print..position..print..area..left end..basic..calculation..pitch..position..move.

Format MoveA (<byteNL>, <byteNH>)

1

<byteNL> : In absolute position (VT_UI1)
<byteNH> : In absolute position (VT_UI1)
Return value : None

Format MoveA (<wordPOS>)

2

<wordPOS> : In absolute position (VT_UI2)
wordPOS = (byteNL + byteNH × 256)
Return value : None

Usage**example**

```
ctrl.Execute "MoveA", Array(1, 2)
ctrl.Execute "MoveA", 513
```

3.5.3. CaoController::Execute("SetHTPosition") command

The horizontal tab position is set.

VntDataArray specifies the digit number from the left end of the print area to the horizontal tab position.

The number of horizontal tabs that can be set is up to 32.

When the horizontal tab position array is not set (VT_EMPTY or VT_NULL), the set horizontal tab is

cleared.

Format SetHTPosition (<vntDataArray>)

1

<vntDataArray> : In horizontal tab position array (VT_UI1 | VT_ARRAY)
Return value : None

Format SetHTPosition ()

2

Argument : None
Return value : None

Usage

example

```
ctrl.Execute "SetHTPosition", Array(8, 16, 24, 32)
ctrl.Execute "SetHTPosition"
```

3.5.4. CaoController::Execute("MoveR") command

Print..position..present location..basic..calculation..pitch..position..move.

When the positive number (plus) is moved left when moving from present location to the right side, the negative number (minus) is specified.

Format MoveR (<byteNL>, <byteNH>)

1

<byteNL> : In relative position (VT_UI1)
<byteNH> : In relative position (VT_UI1)
Return value : None

Format MoveR (<shortPos>)

2

<shortPos> : In relative position (VT_I2)
shortPos = (byteNL + byteNH × 256)
Return value : None

Usage**example**

```
ctrl.Execute "MoveR", Array(1, 2)
ctrl.Execute "MoveR", 513
ctrl.Execute "MoveR", -513
```

3.5.5. CaoController::Execute("SetSMModeAlignment") command

The arrangement of printed data is set.

<i>byteData</i>	Function
0,48	It sets it to "Left justification".
1,49	It sets it to "Center justification".
2,50	It sets it to "Right justification".

Format SetSMModeAlignment (<byteData>)

<byteData> : In arrangement position (VT_UI1)

Return value : None

Usage**example**

```
ctrl.Execute "SetSMModeAlignment", 1
```

3.5.6. CaoController::Execute("SetSMModeLeftMargin") command

Left..margin..printable area..left end..basic..calculation..pitch..set.

Format SetSMModeLeftMargin (<byteNL>, <byteNH>)

1

<byteNL> : In left margin position (VT_UI1)

<byteNH> : In left margin position (VT_UI1)

Return value : None

Format SetSMModeLeftMargin (<wordPos>)

2

<wordPos> : In left margin position (VT_UI2)
 $\text{wordPos} = (\text{byteNL} + \text{byteNH} \times 256)$
 Return value : None

Usage**example**

```
ctrl.Execute "SetSModeLeftMargin", Array(1, 2)
ctrl.Execute "SetSModeLeftMargin", 513
```

3.5.7. CaoController::Execute("MoveSModeBeginningLine") command

The printer is put into the state of "Head of the line" after specified operation is executed, and the print position is moved to the head of the line.

byteData	Function
0,48	Data in a present print buffer is deleted.
1,49	Data in a present print buffer is printed.

Format MoveSModeBeginningLine (<byteData>)

<byteData> : In operation mode (VT_UI1)
 Return value : None

Usage**example**

```
ctrl.Execute "MoveSModeBeginningLine", 0
```

3.5.8. CaoController::Execute("SetSModePrintAreaWidth") command

Print..area..width..basic..calculation..pitch..set.

Format SetSModePrintAreaWidth (<byteNL>, <byteNH>)

1

<byteNL> : Width of in print area (VT_UI1)
 <byteNH> : Width of in print area (VT_UI1)
 Return value : None

Format SetSMoDePrintAreaWidth (<wordPOS>)

2

<wordPos> : Width of in print area (VT_UI2)
 wordPos = (byteNL + byteNH × 256)
 Return value : None

Usage

example

```
ctrl.Execute "SetSMoDePrintAreaWidth", Array(1, 2)
ctrl.Execute "SetSMoDePrintAreaWidth", 513
```

3.6. Mechanism control command

3.6.1. GaoController::Execute("CutPaper") command

Cutting the form is executed.

Cutting the form specified with vntMode is executed.

Function name	Function	vntMode	Cutting shape
< function A>	Cutting of form	0, 48	Full cutting
		1, 49	Partial cut
< function B>	Paper feed and cutting of form	65	Full cutting
		66	Partial cut
< function C>	Reservation of form cutting position	97	Full cutting
		98	Partial cut
< function D>	Paper feed, cutting form, and head putting out of form	103	Full cutting
		104	Partial cut

Each function operation is as follows.

Function name	Function
< function A>	Cutting the form is executed.
< function B>	After paper feed, even (..vntPos × vertical direction.. ..basis.. ..calculation.. pitch) cuts the form.
< function C>	It sets, and it sets it as a form cutting position of (..vntPos × vertical direction.. ..basis.. ..calculation.. pitch). When the position comes to the auto cutter position by the print and paper feed, etc. , the form is cut.
< function D>	The paper feed to (..vntPos × vertical direction.. ..calculation.. pitch) and the form are cut

D>	and after it executes it, the head putting out of the form is done by backing feeding.
----	--

Format CutPaper (<vntMode> [, <vntPos>])

<vntMode> : In cutting mode (VT_UI1)
 <vntPos> : In cutting position (VT_UI1)
 Return value : None

Usage

example

```
Dim vntDataArray As Variant

vntDataArray = Array(&h31, &h32, &h33, &h34, &h35) ' 12345
ctrl.Execute "PrintLF", vntDataArray
ctrl.Execute "PrintLF", "12345"
```

3.7. Chinese character control command

3.7.1. CaoController::Execute("SetKanjiSetting") command

The decoration of the character of the multi bytecode character is specified in the lump.

ByteData : Bit	Binary number	Function	Hexadecimal number	Decimal number
0	0	(reservation)	00	0
1	0	(reservation)	00	0
2	0	The horizontal twice expansion is released.	00	0
	1	The horizontal twice expansion is specified.	04	4
3	0	The length twice expansion is released.	00	0
	1	The length twice expansion is specified.	08	8
4 ~ 6	0	(reservation)	00	0
7	0	The addition of the Chinese character underline is released.	00	00
	1	The addition of the Chinese character underline is specified.	80	128

Format SetKanjiSetting (<byteData>)

<byteData> : In Chinese character mode (VT_UI1)
 Return value : None

Usage**example**

```
ctrl.Execute "SetKanjiSetting", 4
```

3.7.2. CaoController::Execute("StartKanjiMode") command

The Chinese character mode is begun.

Format StartKanjiMode ()

Argument : None

Return value : None

Usage**example**

```
ctrl.Execute "StartKanjiMode"
```

3.7.3. CaoController::Execute("SetKanjiUnderLineMode") command

The Chinese character underline is specified and released.

<i>byteData</i>	Function
0,48	The addition of the Chinese character underline is released.
1,49	The addition of the setting of the Chinese character underline of the width of one dot and the underline is specified.
2,50	The addition of the setting of the Chinese character underline of the width of two dots and the underline is specified.

Format SetKanjiUnderLineMode (<byteData>)

<byteData> : In Chinese character underline setting (VT_UI1)

Return value : None

Usage**example**

```
ctrl.Execute "SetKanjiUnderLineMode", 1
```

3.7.4. CaoController::Execute("StopKanjiMode") command

The Chinese character mode is ended.

Format StopKanjiMode ()

Argument : None

Return value : None

Usage

example

```
ctrl.Execute "StopKanjiMode"
```

3.7.5. CaoController::Execute("SetKanjiCodeSystem") command

The JIS code system or the shifted JIS code system in a Japanese specification is selected.

byteData	Function
0,48	The JIS code system is selected.
1,49	The shifted JIS code system is selected.

Format SetKanjiCodeSystem (<byteData>)

<byteData> : In Kanji character code system (VT_UI1)

Return value : None

Usage

example

```
ctrl.Execute "SetKanjiCodeSystem", 1
```

3.7.6. CaoController::Execute("SetKanjiSpacing") command

The amount of left space and the amount of right space of the multi bytecode character are set.

The amount of left space is set to byteLSpace × basis calculation pitch.

The amount of right space is set to byteRSpace × basis calculation pitch.

Format SetKanjiSpacing (<byteLSpace>, <byteRSpace>)

<byteLSpace> : Amount of in left space (VT_UI1)

<byteRSpace> : Amount of in right space (VT_UI1)
 Return value : None

Usage**example**

```
ctrl.Execute "SetKanjiSpacing", Array(1, 2)
```

3.7.7. CaoController::Execute("SetKanjiQuadrupleSizeMode") command

Four time multi bytecode character corner character is specified and released.

The corner character four times when the least significant bit of byteData is 0 is released and the normal size is specified.

The corner character four times when the least significant bit of byteData is one is specified.

Format SetKanjiQuadrupleSizeMode (<byteData>)

<byteData> : In Four time corner character setting (VT_UI1)
 Return value : None

Usage**example**

```
ctrl.Execute "SetKanjiQuadrupleSizeMode", 1
```

3.7.8. CaoController::Execute("SetKanjiFont") command

The font of the multi bytecode character (Chinese font) is selected.

byteData	Function
0,48	Chinese font A is selected.
1,49	Chinese font B is selected.
2,50	Chinese font C is selected.

Format SetKanjiFont (<byteData>)

<byteData> : In Chinese character font setting (VT_UI1)
 Return value : None

Usage

example

```
ctrl.Execute "SetKanjiFont", 0
```

3.8. Supplementary functional command

3.8.1. CaoController::Execute("RTMPowerOff") command

The preparation processing of the power-off is executed, and power supply OFF notification is received.

As for this command, processing ends at the stage where power supply OFF notification was normally received.

The error occurs when power supply OFF notification cannot be received by some causes.

Format RTMPowerOff ()

Argument : None

Return value : None

Usage**example**

```
ctrl.Execute "RTMPowerOff"
```

3.8.2. CaoController::Execute("RTMClearBuffer") command

After all data in the receive buffer and the print buffer is deleted, a clear response is received.

As for this command, processing ends at the stage where a clear response was normally received.

The error occurs when a clear response cannot be received by some causes.

Format RTMClearBuffer ()

Argument : None

Return value : None

Usage**example**

```
Dim vntResult as Variant  
vntResult = ctrl.Execute "RTMClearBuffer"
```

3.8.3. CaoController::Execute("InitializePrinter") command

⁴Clearness in the print buffer and a part of set value of the command are returned to an initial value.

Internal is executed as for this command at AddController.

Format InitializePrinter ()

Argument : None

Return value : None

Usage

example

```
ctrl.Execute "InitializePrinter"
```

3.8.4. CaoController::Execute("TestPrint") command

The specified test print is executed.

BytePaper specifies the target form for the test print shown in the next table.

<i>byteData</i>	Function
0,48	Basic seat (machine glazed paper)
1,49 2,50	Machine glazed paper

ByteTestPattern specifies the kind of the test print shown in the next table.

<i>byteTestPattern</i>	Function
1,49	Hexadecimal dump
2,50	State print of printer
3,51	Roll pattern print
64	Automatic setting of form layout

Format TestPrint (<bytePape>, <byteTestPattern>)

<bytePaper> : Kind of in form (VT_UI1)

<byteTestPattern> : Kind of in test print (VT_UI1)

Return value : None

Usage

example

```
ctrl.Execute "TestPrint", Array(1, 1)
```

⁴ 初期値に戻るコマンドの内容に関してはコマンドリファレンスを参照してください

3.8.5. **GaoController::Execute("GetPrinterID")** command

Specified printer ID or printer information is received.

• Specified printer ID is received.

<i>byteData</i>	Printer ID	Specification
1,49	Model ID	Model
2,50	Type ID	Function that supports it
3,51	Version ID	Version of firmware

• Specified printer information A is received.

<i>byteData</i>	Kind of printer information	Specification
33	Type information	Function that supports it
35,36,96,110	Please refer to model information.	Please refer to model information.

• Specified printer information B is received.

<i>byteData</i>	Kind of printer information	Specification
65	Firmware version	Version of firmware
66	Manufacturer name	Manufacturer name
67	Model name	Model name
68	Serial number	Serial number of product
69	Kind of multi national language font	Japanese specification "KANJI JAPANESE"
		Simplified Chinese characters Chinese specification "CHINA GB18030" or "CHINA GB2312"
		Traditional Chinese characters Chinese specification "TAIWAN BIG-5"
		Korean specification "KOREA C-5601C"
		South Asia specification "THAI 1 PASS"
111	Please refer to model information.	Please refer to model information.
112	Please refer to model information.	Please refer to model information.

Format GetPrinterID (<byteData>)

<byteData> : In acquisition number (VT_UI1)
 Return value : Out receive data
 1-3 and acquisition number = 49-51
 Printer ID (VT_UI1)
 33, 35, and acquisition number = 96,110
 Printer information A (VT_UI1 | VT_ARRAY)
 Acquisition number =65-69
 Printer information B (VT_BSTR)

Acquisition number =111,112

Printer information B (VT_UI | VT_ARRAY)

Usage

example

```
Dim vntResult as Variant
vntResult = ctrl.Execute "GetPrinterID", 0
```

3.8.6. CaoController::Execute("SetBasePitch") command

A horizontal basic calculation pitch is set and the basic calculation pitch in x mm 25.4/{1/x inch } and the vertical direction is set to y mm 25.4/{1/y inch }.

(byteX = 0)A horizontal basic calculation pitch is returned to an initial value at the time of it is time when ..drinking...

(byteY = 0)The basic calculation pitch in the vertical direction is returned to an initial value at the time of it is time when ..drinking...

Format SetBasePitch (<byteX>, <byteY>)

<byteX>	:	Horizontal in basic calculation pitch (VT_UI1)
<byteY>	:	Vertical direction of in basic calculation pitch (VT_UI1)
Return value	:	None

Usage

example

```
ctrl.Execute "SetBasePitch", Array(1, 2)
```

3.9. Original command

3.9.1. CaoController::Execute ("Send") command

The specified command format is transmitted.

Format Send (<vntDataArray>)

<vntDataArray>	:	In transmission byte array (VT_UI1 VT_ARRAY)
Return value	:	None

Usage**example**

```
Dim vntdataArray As Variant

vntdataArray = Array(&h31, &h32, &h33, &h34, &h35, &h0A) ' 12345LF
ctrl.Execute "Send", vntdataArray
```

3.9.2. CaoController::Execute ("SendAndReceive") command

The specified command format is transmitted, and the response data is received.

It becomes a data reception waiting when the command without the response is executed and time-out error (0x80000900) occurs.

Format SendAndReceive (<vntdataArray>)

<vntdataArray> : In transmission byte array (VT_UI1 | VT_ARRAY)
 Return value : The response data that receives out is not processed at all and it
 returns it including the header.
 (VT_UI1 | VT_ARRAY)

Usage**example**

```
Dim vntdataArray As Variant
Dim vntResult as Variant

vntdataArray = Array(&h31, &h32, &h33, &h34, &h35) ' 12345
ctrl.Execute "Send", vntdataArray
vntdataArray = Array(&h10, &h14, &h08, &h01, &h03, &h14, &h01, &h06, &h02, &h08 ) ' DLE DC4 fn(8)
vntResult = ctrl.Execute("SendAndReceive", vntdataArray)
```

3.9.3. CaoController::Execute("AddPrint") command

The character string and the command data are stored in the print buffer.

The argument occurs when there is no transmitted data and illegal error (0x80070057) occurs.

Format AddPrint (<vntdataArray>)

1

<vntdataArray> : In transmission byte array (VT_UI1 | VT_ARRAY)
 Return value : None

Format AddPrint (<bstrData>)

2

<bstrData> : In transmission byte character string (VT_BSTR)
Return value : None

Usage

example

```
Dim vntDataArray As Variant

vntDataArray = Array(&h31, &h32, &h33, &h34, &h35) ' 12345
ctrl.Execute "AddPrint", vntDataArray
ctrl.Execute "AddPrint", "12345"
```

3.9.4. **GaoController::Execute("AddPrint_JP") command**

A Japanese character string is stored in the print buffer.

The argument occurs when there is no transmitted data and illegal error (0x80070057) occurs.

It is convenient to use AddPrint_JP or PrintLF_JP to print Japanese (hiragana, katakana, and Chinese character).

AddPrint_JP is "Selection of an international character (Japan)", and "Selection of the character-code table. " (page one katakana) on the inside" It changes to "Selection of the Kanji character code system (shift JIS)".

After processing is completed, processing has been returned to "Selection of an international character (Japan)", "Selection of the character-code table (..page.. 0 PC437: USA and StandardEurope)", and "Selection (JIS) of the Kanji character code system".

Because abnormality occurs by the return processing, the setting becomes irregular though it succeeds in the transmission of the print buffer when processing abnormality (0x80100002) is generated after it transmits.

Format AddPrint_JP (<bstrData>)

<bstrData> : In transmission byte character string (VT_BSTR)
Return value : None

Usage

example

```
Ctrl.Execute "AddPrint_JP" and "..height.. [aiueoaiueo] sub-[iueo] ..12345abcde each other..."
```

3.9.5. CaoController::Execute("PrintLF_JP") command

A Japanese character string is stored in the print buffer and the paper feed of the print and one line is executed.

The argument occurs when there is no transmitted data and illegal error (0x80070057) occurs.

It is convenient to use AddPrint_JP or PrintLF_JP to print Japanese (hiragana, katakana, and Chinese character).

PrintLF_JP is "Selection of an international character (Japan)", and "Selection of the character-code table. " (page one katakana) on the inside" It changes to "Selection of the Kanji character code system (shift JIS)".

After processing is completed, processing has been returned to "Selection of an international character (Japan)", "Selection of the character-code table (..page.. 0 PC437: USA and StandardEurope)", and "Selection (JIS) of the Kanji character code system".

Because abnormality occurs by the return processing, the setting becomes irregular though it succeeds in the print processing when processing abnormality (0x80100002) is generated after it transmits.

Format PrintLF_JP ([<bstrData>])

<bstrData>	:	In transmission byte character string (VT_BSTR)
Return value	:	None

Usage

example

```
Ctrl.Execute "PrintLF_JP" and "..height.. [aiueoaiueo] sub-[iueo] ..12345abode each other... "
```

3.9.6. CaoController::Execute("AddPrint_EN") command

An English character string is stored in the print buffer.

The argument occurs when there is no transmitted data and illegal error (0x80070057) occurs.

It is convenient to use AddPrint_EN or PrintLF_EN to print English.

It changes to AddPrint_EN is "Selection of an international character (United States)", and "Selection of the character-code table. " (..page.. 0 PC437: USA and StandardEurope) on the inside".

After processing is completed, processing has been returned to "Selection of an international character (Japan)".

Because abnormality occurs by the return processing, the setting becomes irregular though it succeeds in the transmission of the print buffer when processing abnormality (0x80100002) is generated after it transmits.

Format AddPrint_EN (<bstrData>)

<bstrData> : In transmission byte character string (VT_BSTR)
Return value : None

Usage

example

```
ctrl.Execute "AddPrint_EN", "12345abcde"
```

3.9.7. CaoController::Execute("PrintLF_EN") command

A Japanese character string is stored in the print buffer and the paper feed of the print and one line is executed.

It is convenient to use AddPrint_EN or PrintLF_EN to print English.

It changes to PrintLF_EN is "Selection of an international character (United States)", and "Selection of the character-code table. " (..page.. 0 PC437: USA and StandardEurope) on the inside".

After processing is completed, processing has been returned to "Selection of an international character (Japan)".

Because abnormality occurs by the return processing, the setting becomes irregular though it succeeds in the print processing when processing abnormality (0x80100002) is generated after it transmits.

Format PrintLF_EN ([<bstrData>])

<bstrData> : In transmission byte character string (VT_BSTR)
Return value : None

Usage

example

```
ctrl.Execute "PrintLF_EN", "12345abcde"
```

3.9.8. CaoController::Execute("AddPrint_DE") command

A German character string is stored in the print buffer.

The argument occurs when there is no transmitted data and illegal error (0x80070057) occurs.

It is convenient to use AddPrint_DE or PrintLF_DE to print German.

It changes to AddPrint_DE is "Selection of an international character (Germany)", and "Selection of the character-code table. " (..page.. 0 PC437: USA and StandardEurope) on the inside".

After processing is completed, processing has been returned to "Selection of an international character (Japan)".

Because abnormality occurs by the return processing, the setting becomes irregular though it succeeds in the transmission of the print buffer when processing abnormality (0x80100002) is generated after it transmits.

Format AddPrint_DE (<bstrData>)

<bstrData> : In transmission byte character string (VT_BSTR)
Return value : None

Usage

example

```
ctrl.Execute "AddPrint_DE", "12345abcde[¥] {}" ' 12345abcdeA0Uaous
```

3.9.9. CaoController::Execute("PrintLF_DE") command

A German character string is stored in the print buffer and the paper feed of the print and one line is executed.

It is convenient to use AddPrint_DE or PrintLF_DE to print German.

It changes to PrintLF_DE is "Selection of an international character (Germany)", and "Selection of the character-code table. " (..page.. 0 PC437: USA and StandardEurope) on the inside".

After processing is completed, processing has been returned to "Selection of an international character (Japan)".

Because abnormality occurs by the return processing, the setting becomes irregular though it succeeds in the print processing when processing abnormality (0x80100002) is generated after it transmits.

Format PrintLF_DE ([<bstrData>])

<bstrData> : In transmission byte character string (VT_BSTR)
Return value : None

Usage

example

```
ctrl.Execute "PrintLF_DE", "12345abcde[¥] {}" ' 12345abcdeA0Uaous
```

4. Variable list

4.1.1. System variable

Table4-1CaoController class system variable list

Variable identifier	Data type	Explanation	Attribute	
			get	put
@MAKER_NAME	VT_BSTR	"EPSON maker name =" is returned.	-	-
@VERSION	VT_BSTR	Provider version information is returned.	-	-

5. Error code

In the ESCPOS provider, the following and peculiar the error code is defined.

About the ORiN2 commonness error, Please refer to the chapter of the error code of "ORiN2 Programming guide".

Table5-1Peculiar error code

Error name	Error number	Explanation
The reception packet is abnormal.	0x80100000	The receive data format is abnormal.
The specific data preparation is abnormal.	0x80100001	The equipment was not able to prepare specific data.
After it transmits, processing is abnormal.	0x80100002	It failed because of the postprocessing after it had transmitted.