

Sublimation type Card Printer CX-120

Technical Document for Software Development

4th issue 15th Apr 2014

JVC KENWOOD Corporation

Notice

1. Before using the document, please read through the “Software License Agreement” at the end of CX-120 Startup Guide. You will not be allowed to develop any software by the use of the document unless you have agreed to accept the terms stated in the agreement.
2. We shall not be responsible for anything developed by the use of the documents.
3. The documents may be changed for improvements without prior notice.
4. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. In this book, the names for Microsoft® Windows® 8 operating system, Microsoft® Windows® 7 operating system, Microsoft® Windows Vista™ operating system, Microsoft® Windows® XP operating system and Microsoft® Windows® 2000 operating system are written as Windows 8, Windows 7, Windows Vista, Windows XP and Windows 2000 respectively.

<Revision Contents>

2006.2.15 1st issue	1 st Release.
2007.12.28 2nd issue	1) 2. Operational Environment : Added Windows Vista. 2) 5.11 Getting and Changing Printer Setting : Added Log Select function & Mode Select function. 3) 6.3 How to make IC Encoding program: Changed the procedure of the return code from the IC Encode DLL.
2010.06.08 3rd issue	1) 2. Operational Environment : Added Windows 7 32bit/64bit. 2) 3. System Configuration / 5.1 Program Construction : Added the 64bit file name of DLL.
2014.04.15 4th issue	1) 2. Operational Environment : Added Windows 8 32bit/64bit. Added Service Pack version of Windows 7, Vista and XP.

<Table of contents>

1. Overview	4
2. Operational Environment	4
3. System Configuration	5
4. About Card Printer	6
4.1 Command Structure	6
4.2 Card Position	6
5. Printer Control Function	7
5.1 Program Construction	7
5.2 Basic rule of the return code from functions	7
5.3 Process flow to issue the card	8
5.4 Scan Printer	9
5.5 Getting Printer Status	9
5.6 Printing	10
5.7 Moving & Discharging Card	13
5.8 Magnetic Encoding	15
5.9 IC Encoding	16
5.10 Initializing Printer	17
5.11 Getting and Changing Printer Setting	18
6. How to encode with Printer Driver	29
6.1 Inline Encoding	29
6.2 Encoding by using ExtEscape()	30
6.3 How to make IC Encoding program	31
7. How to change the driver setting	33
7.1 How to use ExtEscape()	33
7.2 Change the setting	34
7.3 Get the current setting	34
7.4 Programming sample	35
7.5 About the parameter	37
7.6 About the error code	39

Appendix 1. Error Code Table

Appendix 2. ASCII Code Table and Magnetic Data

1. Overview

This document explains the printer software from the point of software development view. If you install the printer software, the Printer Control DLL which is used by the printer driver will be also installed. Although Printing and Encoding is done by using the printer driver normally, you can do them by using Printer Control DLL directly without using Printer Driver. This shows how to use Printer Control DLL, and how to do IC and MAG encoding with Printer Driver.

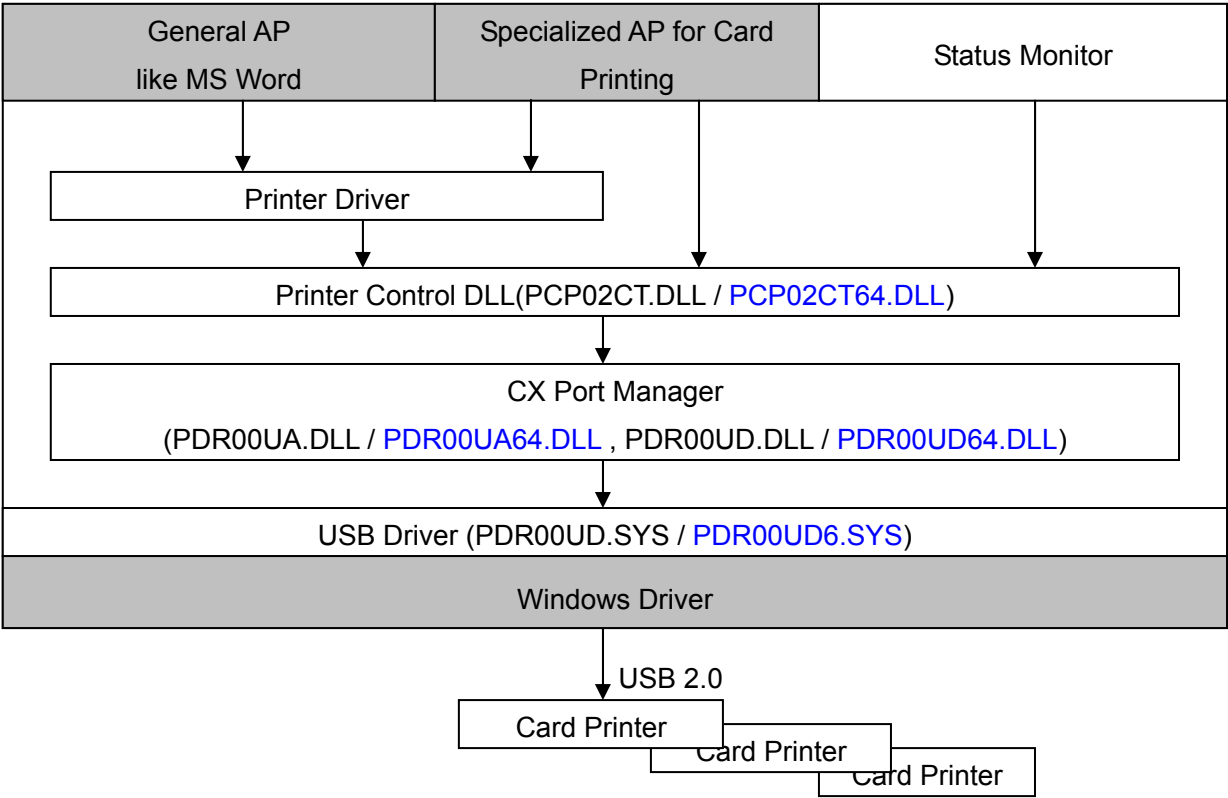
2. Operational Environment

Following table shows the operational environment of the software.

Item	Contents	Note
OS	Windows 8 32bit / 64bit Windows 7 (First Release, Service Pack 1) 32bit / 64bit Windows Vista (Service Pack 1/2) 32bit Windows XP (Service Pack 2/3) 32bit Windows 2000 Professional (Service Pack 4) 32bit	
Peripheral	CX-120 Card Printer	Card printer is connected to the PC with USB cable.

3. System Configuration

Following figure shows the configuration of the printer software. The software described on the white rectangle in the figure shows they are software which are installed from the CD-ROM attached to the printer.



Note: The file names are described such as (32bit / 64bit).

4. About Card Printer

4.1 Command Structure

The structure of the command to the printer is according to the SCSI rule. One of the printers being connected a PC can be specified by both the Slot number and the ID of the printer. It makes possible to control printers by 7. Almost functions require both Slot number and ID to specify the printer. Slot number is a value decided by system environment, and ID is the value which is subtracted by 1 from Unit number of the printer.

4.2 Card Position

The printer command may fail if the card position is not proper for the command. For example, MG Encoding command, which accesses the magnetic stripe on the card, will fail if the card is not positioned at MG Encoder Position. Read Position command is prepared to know the card position.

Card Position in the Card Printer

Card Position	Explanation
Print Origin	The position to start printing
Contact IC encoder position	The position to do Contact IC encoding.
No-Contact IC encoder position	The position to do No-Contact IC encoding.
MAG encoder position	The position to do MAG encoding.

5. Printer Control Function

5.1 Program Construction

They are functions in Printer Control DLL which is supplied as Windows DLL. They are installed when the printer driver is installed.

Name	OS	File Name
Printer Control DLL	32bit	PCP02CT.DLL
	64bit	PCP02CT64.DLL

Note: Use the PCP02CT.DLL of 32bit version when you use 32bit application software on 64bit OS.

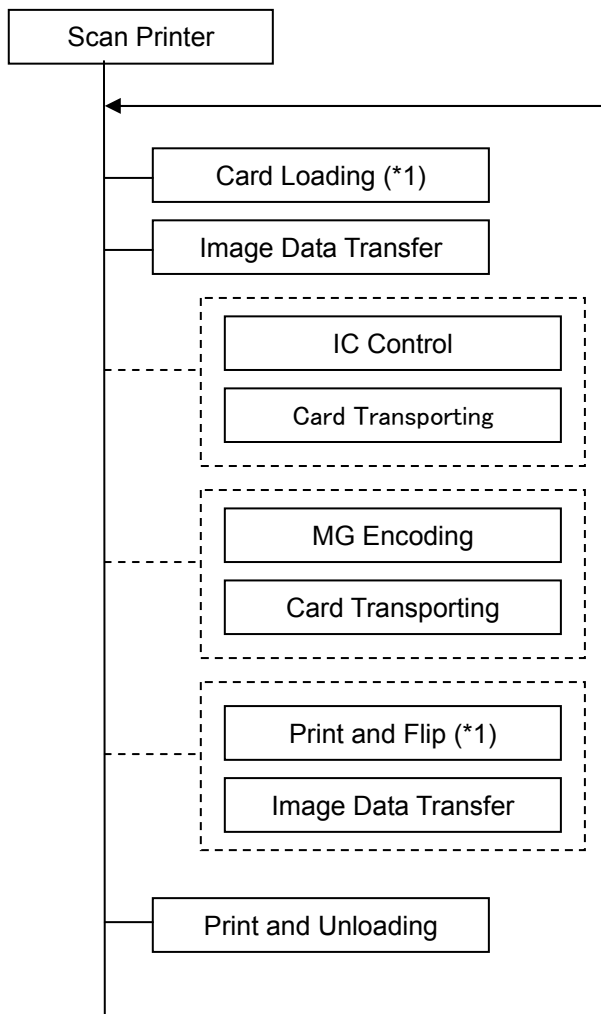
5.2 Basic rule of the return code from functions

All function returns Return Code according to the following common rules.

- 0: It means success.
- Positive value: It means the command was not accepted because the printer or function is busy. The state is temporary and it will be eliminated automatically. The calling process should retry after a little later.
- Negative value: It means error. "Appendix 1. Error Code Table" shows the detail.

5.3 Process flow to issue the card

Printer Control DLL identifies the printer with Slot Number and ID. So, the 1st procedure is to find Slot Number and ID to access the printer. Slot Number and ID of the printer keep same value while it is plugged into PC. The basic command structure to the printer is based on SCSI. Following figure shows standard processing to issue a card.



Note

(*1) To make printing speed up, these commands should not wait the end of processing. It is done by setting Immediate Flag 1.

(*2) When a command is issued to the printer, a card must be placed at the appropriate position for the command. For example, a card must be positioned at MG Encoder when MG encoding starts. Both Card Loading and Card Transporting functions have Destination parameter, and you can know the card position in the printer with Read Position function.

<Hint> Normally, Printer Driver loads the card, and encodes and prints. But if there is a card in the printer already, Printer Driver's processing is done to the card in the printer. By using this function, you can do pre-processing such as encoding by using Printer Control DLL before Printer Driver's process.

Caution: If you use both Printer Control DLL and Printer Driver, Windows Spooler had better be disabled. If any printing data is pooled in Spooler, problem will happen as DLL controls Card Printer directly.

5.4 Scan Printer

5.4.1 Scan Printer Functions

No	Function Name	Explanation
1	int CXCMD_ScanCX120 (int *piSlot, int *piID)	Search for the printer from the first, and return Slot number and ID. The result is set at piSlot and piID.
2	int CXCMD_ScanCX120Next (int *piSlot, int *piID)	Search for the printer from the next of the printer specified by piSlot and piID. The result is set at piSlot and piID.

5.5 Getting Printer Status

5.5.1 Test Unit Ready Function

No.	Function Name	Explanation
1	int CXCMD_TestUnitReady (int iSlot, int iID)	Check the printer condition by issuing Test Unit Ready command to the printer.

5.5.2 Read Position Function

No.	Function Name	Explanation
1	int CXCMD_ReadPosition (int iSlot, int iID, BYTE *pbyBuffer)	Get card position by issuing Read Position command to the printer. Read Position data is set at pbyBuffer.

1) Read Position Data format

Bit Byte	7	6	5	4	3	2	1	0
0	Reserved					PU	Reserved	
1~6	Reserved							
7	Position							

PU (Position Unknown):

- 0: Card is in the printer
- 1: No Card in the printer

Note: Printer reports PU = 1 even if any card is in the card hopper. Printer generates Check Condition while the card is moving.

Position (Card Position. This is effective only when PU is 0)

- 0: At the print origin
- 1: At MG Encoder
- 2: At IC Contact
- 3: At IC Contactless

5.6 Printing

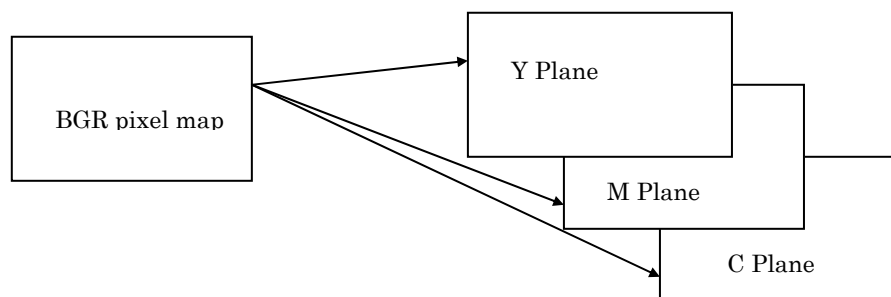
5.6.1 Image Out Function

No	Function Name	Explanation
1	int CXCMD_ImageOut (int iSlot, int iLD, BYTE * pbyPlane, int iLength, int iColor, int iBuffer)	Send Image Data to the printer. All color, which are Y, M, C, K and OP, are send separately. If there are some colors being not sent, such colors will be printed with the color data in the printer when it is printed. <ul style="list-style-type: none">• pbyPlane: Pointer to the color data. Color data must be 1006x633 bytes.• iLength: Length of color data. This must be 1006x633• iColor: Color of the color data 0: Y 1: M 2: C 3: K 5: OP• iBuffer: Color buffer of the printer which color data is stored. 0: Buffer-0 1: Buffer-1

1) How to translate RGB to YMC planes

As CX-120 only supports YMC plane to print colored image, RGB must be transformed to Y, M and C plane. Normally, the formula to translate RGB to YMC is as follows.

$Y = 255-B$, $M = 255-G$, $C = 255-R$.



2) About K

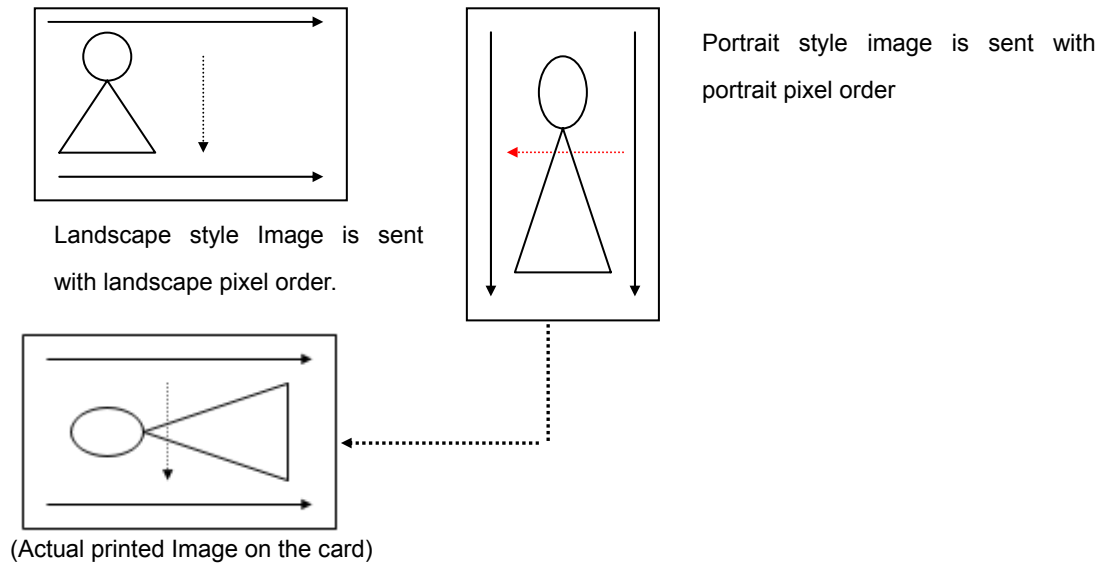
K ink is the ink specialized for black text printing. The data for K ink is Boolean, only the not zero part is printed with black color. It makes the quality of black text better.

3) About OP

OP is a protection layer coated over the ink. The data for OP is Boolean, only the not zero part is coated on the card.

4) About the order of the image sent to printer

The sending order of pixels to printer must be always Landscape order even if the image is created with portrait style. If the design is portrait style like right side of the following figure, pixel order must be adjusted by the application program.



- The solid line means the order of the pixels in a line to send to the printer.
- The dashed line means the order of the lines to send to the printer. The pixel at the top of left corner must be sent at first, and the pixel at the bottom of right corner must be sent at last.

5.6.2 Print Function

No	Function Name	Explanation
1	int CXCMD_Print (int iSlot, int iID, int iColor, int iBuffer, int bFlip, int blmmed)	<p>Print image on the card. The card is moved to Print Position after it is printed and turned over.</p> <ul style="list-style-type: none"> iColor: Color being printed. bit0: 1 if YMC is to be printed. bit1: 1 if K is to be printed. bit2: 1 if OP is to be printed. iBuffer: Color buffer of the printer which color data is stored. 0: Buffer-0 1: Buffer-1 bFlip: Flag whether the card is turned over or not. TRUE: Done after printing ends. FALSE: Not done. Note) bFlip is always regarded as TRUE. blmmed: Flag whether the function returns immediately after sending command to the printer or not. TRUE: Immediately after sending the command. FALSE: After the card is printed.
2	int CXCMD_PrintAndEject (int iSlot, int iID, int iColor, int iBuffer, int bFlip, int blmmed)	<p>Print image on the card and unload it from Card Exit. The card will be turned over after printing ends if bFlip is TRUE.</p> <ul style="list-style-type: none"> iColor: Color being printed. bit0: 1 if YMC is to be printed. bit1: 1 if K is to be printed. bit2: 1 if OP is to be printed. iBuffer: Color buffer of the printer which color data is stored. 0: Buffer-0 1: Buffer-1 bFlip: Flag whether the card is turned over or not. TRUE: Done when the card is discharged. FALSE: Not done. blmmed: Flag whether the function returns immediately after sending command to the printer or not. TRUE: Immediately after sending the command. FALSE: After the card is printed.

Note: Error will be returned if these are called when the card is not located at Print Origin. Positive value will be returned if it is issued while the printer is on the way of moving card or printing. Retry with a little waiting if positive value is returned.

5.7 Moving & Discharging Card

5.7.1 Card Load Function

No	Function Name	Explanation
1	int CXCMD_LoadCard (int iSlot, int iLD, int iDest, int bFlip, int blmmed)	Load a card to the specified position in the printer. <ul style="list-style-type: none">• iDest: Destination of the card. 0: At Print Position 1: At MG Encoder 2: At Contact IC Encoder 3: At Proximity IC Encoder• bFlip: Flag whether the card is turned over or not. TRUE: Done. FALSE: Not done.• blmmed: Flag whether the function returns immediately after sending command to the printer or not. TRUE: Immediately after sending the command. FALSE: After the card is loaded.

Note: Error will be returned if this is used when there is a card in the printer, and positive value will be returned if it is issued while the printer is on the way of moving card or printing. Retry with a little waiting if positive value is returned.

5.7.2 Card Move Function

No	Function Name	Explanation
1	int CXCMD_MoveCard (int iSlot, int iLD, int iDest, int bFlip, int blmmed)	Move a card in the printer to the specified position. <ul style="list-style-type: none">• iDest: Destination of the card. 0: At Print Position 1: At MG Encoder 2: At Contact IC Encoder 3: At Proximity IC Encoder• bFlip: Flag whether the card is turned over or not. TRUE: Turn over. FALSE: Not turn.• blmmed: Flag whether the function returns immediately after sending command to the printer or not. TRUE: Immediately after sending the command. FALSE: After the card is moved.

Note: Error will be returned if this is used when there is no card in the printer, and positive value will be returned if it is issued while the printer is on the way of moving card. Retry with a little waiting if positive value is returned.

5.7.3 Card Unload Function

No	Function Name	Explanation
1	int CXCMD_EjectCard (int iSlot, int iLD, int bFlip, int blmmed)	Unload the card from Card Exit. <ul style="list-style-type: none">• bFlip: Flag whether the card is turned over or not. TRUE: Being turned over and unloaded. FALSE: Not done.• blmmed: Flag whether the function returns immediately after sending command to the printer or not. TRUE: Immediately after sending the command. FALSE: After the card is unloaded.
2	int CXCMD_EjectCardFromNGExt (int iSlot, int iLD, int blmmed)	Unload the card from NG Card Exit. <ul style="list-style-type: none">• blmmed: Flag whether the function returns immediately after sending command to the printer or not. TRUE: Immediately after sending the command. FALSE: After the card is unloaded.

Note: Error will be returned if they are used when there is no card in the printer, and positive value will be returned if it is issued while the printer is on the way of moving card. Retry with a little waiting if positive value is returned.

5.8 Magnetic Encoding

5.8.1 Writing Function

No	Function Name	Explanation
1	int CXCMD_WriteMagData (int iSlot, int iID, BYTE *pbyBuff, int iLength, int iMagFormat)	Write MG data to the magnetic stripe on the card. <ul style="list-style-type: none">• pbyBuff: Pointer to the memory having MG data. The data must be 7unit ASCII code.• iLength: MG data length in byte.• iMagFormat: Type of MG Data<ul style="list-style-type: none">0x07: JIS2 Track1(7unit come, 69 characters at most)0x16: ISO Track1(6unit code, 76 characters at most)0x24: ISO Track2(4 unit code, 37 characters at most)0x34: ISO Track3(4 unit code, 104 characters at most)

Note: Error will be returned if this is used when the card is not located at MAG encoder position, and positive value will be returned if it is issued while the printer is on the way of moving card. Retry with a little waiting if positive value is returned.

5.8.2 Reading Function

No	Function Name	Explanation
1	int CXCMD_ReadMagData (int iSlot, int iID, BYTE *pbyBuff, int *piLength, int iMagFormat)	Read MG data from the magnetic stripe on the card <ul style="list-style-type: none">• pbyBuff: Pointer to the memory storing MG data. 7unit ASCII code is stored.• piLength: MG data length is set.• iMagFormat: Type of MG Data<ul style="list-style-type: none">0x07: JIS2 Track1(7unit come, 69 characters at most)0x16: ISO Track1(6unit code, 76 characters at most)0x24: ISO Track2(4 unit code, 37 characters at most)0x34: ISO Track3(4 unit code, 104 characters at most)

Note: Error will be returned if this is used when the card is not located at MAG encoder position, and positive value will be returned if it is issued while the printer is on the way of moving card. Retry with a little waiting if positive value is returned.

5.9 IC Encoding

5.9.1 IC Control Function

No	Function Name	Explanation
1	int CXCMD_ICControl (int iSlot, int iID, int iIType, int iAction)	Issues IC Control Command to the printer. <ul style="list-style-type: none">• iIType: Type of IC Encoder 0: Contact IC Encoder 1: Proximity IC Encoder• iAction: 0: IC Contacting 1: IC Releasing Note) IC Contacting and Releasing must be done even if the operation is for the Proximity IC Encoder.

Note1: Error will be returned if this is used when the card is not located at the appropriate IC encoder position, and positive value will be returned if it is issued while the printer is on the way of moving card. Retry with a little waiting if positive value is returned.

Note2: CXCMD_MoveCard() is rejected as error after CXCMD_ICControl() is issued with iAction=0. It is required to issue CXCMD_ICControl() with iAction=1 before moving card even if it is for No-Contact IC encoding.

5.10 Initializing Printer

5.10.1 Rezero Function

No.	Function Name	Explanation
1	int CXCMD_RezeroUnit (int iSlot, int iID)	Initialize printer by using Rezero command. Printer discharges the card and adjusts both Retransfer position and Ink position.

5.11 Getting and Changing Printer Setting

5.11.1 Inquiry Function

No	Function Name	Explanation
1	int CXCMD_StandardInquiry (int iSlot, int iID, BYTE *pbyBuffer)	Issue Inquiry command to the printer and get Standard Inquiry Data. <ul style="list-style-type: none"> pbyBuffer: Pointer to the memory to store data. The size of memory must be enough to store data. Note) Data format is explained at “1) Standard Inquiry Data format” in this chapter.
2	int CXCMD_VPDInquiry (int iSlot, int iID, int iPage, BYTE *pbyBuffer)	Issue Inquiry command to the printer and get Vital Product Data. <ul style="list-style-type: none"> iPage: Type of data. 0x01: Head Information Data. pbyBuffer: Pointer to the memory to store data. The size of memory must be enough to store data. Note) Data format is explained at “2) Inquiry Head Information Data format” in this chapter.

1) Standard Inquiry Data format

Byte \ Bit	7	6	5	4	3	2	1	0
0	Peripheral Qualifier			Peripheral Device Type				
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	1	0
3	Reserved				0	0	1	0
4	Additional Length (33H)							
5~6	Reserved							
7	0	0	0	0	0	0	0	0
8~15	Vendor Identification “VDS” (ASCII)							
16~31	Product Identification “CX-120” (ASCII)							
32~35	Product Revision Level “????” (ASCII)							
36	Config Revision Level (Binary)							
37	Table Revision Level (Binary)							
38			LA	ICLI	ICJI	ICII	MGJI	MGII
39~42	Laminator Unit Product Revision Level “????” (ASCII)							
43~55	Reserved							

MGII (MG ISO Unit Installed): 1 if MG ISO Unit is installed, 0 if not.

MGJI (MG JIS Unit Installed): 1 if MG JIS Unit is installed, 0 if not.

ICII (IC Contact ISO Installed): 1 if IC Contact ISO Unit is installed, 0 if not

ICJI (IC Contact JIS Installed): 1 if IC Contact JIS Unit is installed, 0 if not

ICLI (IC Contactless Installed): 1 if IC Contactless Unit is installed, 0 if not.(always 1)

LA(Laminator Attached): 1 if Laminator unit is attached, 0 if not

2) Inquiry Head Information Data format

Byte \ Bit	7	6	5	4	3	2	1	0
0	Peripheral Qualifier			Peripheral Device Type				
1	01H (Page Code)							
2	Reserved							
3~7	(N-3)H (Page Length)							
8	Head Information (ASCII, Only top of 8 characters are effective. Remainder is filled with NULL)							
27								

5.11.2 Log Sense Function

No	Function Name	Explanation
1	int CXCMD_LogSelect (int iSlot, int iID)	Issue Log Select command to clear Printed Card Counter in the printer. It is non volatile counter which is counted up by one every when a card is printed. You can know the value by Log Sense command.
2	int CXCMD_LogSense (int iSlot, int iID, int iPage, BYTE *pbyBuffer)	Issue Log Sense command to the printer and get Log Sense data in the printer. <ul style="list-style-type: none"> iPage: Type of data 0x06: Data in Non-medium error page 0x38: Data in Counters page 0x3a: Data in Medium Quantity page pbyBuffer: Pointer to the memory to store data. The size of memory must be enough to store data. Note) Data format is explained at “1) Log Sense Data format” in this chapter.

1) Log Sense Data format

(1) Non-medium error page (Page Code=0x06)

Byte \ Bit	7	6	5	4	3	2	1	0
0	Reserved		Page Code (06H)					
1	Reserved							
2	(MSB) <div>Page Length (0008H)</div> (LSB)							
3								
4	(MSB) <div>Parameter Code (0000H)</div> (LSB)							
5								
6	0	1	0	0	00b		0	0
7	Parameter Length (0004H)							
8	(MSB) <div>Non-medium error count(00H)</div> (LSB)							
9								
10								
11								

* This is not supported. The contents of “Non-medium error count” is always zero.

(2) Counter page (Page Code=0x38)

Bit Byte	7	6	5	4	3	2	1	0
0	Reserved		Page Code (38H)					
1	Reserved							
2	(MSB)Page Length (0018H)(LSB)							
3								
4	(MSB)Parameter Code (0000H)(LSB)							
5								
6	0	0	0	0	00b		0	0
7	Parameter Length (0004H)							
8	(MSB)Total count(LSB)							
9								
10								
11								
12	(MSB)Parameter Code (0001H)(LSB)							
13								
14	0	0	0	0	00b		0	0
15	Parameter Length (0004H)							
16	(MSB)Free count(LSB)							
17								
18								
19								
20	(MSB)Parameter Code (0002H)(LSB)							
21								
22	0	0	0	0	00b		0	0
23	Parameter Length (0004H)							
24	(MSB)Head count(LSB)							
25								
26								
27								

(3) Medium quantity page (Page Code=0x3a)

Byte \ Bit	7	6	5	4	3	2	1	0
0	Reserved		Page Code (3AH)					
1	Reserved							
2	(MSB) <div>Page Length (000AH)</div> (LSB)							
3								
4	(MSB) <div>Parameter Code (0000H)</div> (LSB)							
5								
6	0	0	0	0	00b		0	0
7	Parameter Length (0001H)							
8	Ink quantity							
9	(MSB) <div>Parameter Code (0001H)</div> (LSB)							
10								
11	0	0	0	0	00b		0	0
12	Parameter Length (0001H)							
13	Card quantity							

Ink quantity;

0: Empty ~ 50: Full FFH: Unknown ink is attached

Card quantity;

0: Any 1:None

5.11.3 Mode Sense/Select Function

No	Function Name	Explanation
1	int CXCMD_ModeSense (int iSlot, int iID, int iPage, BYTE *pbyBuffer)	<p>Issue Mode Sense command to the printer and get Mode Sense data.</p> <ul style="list-style-type: none"> iPage: Type of data 0x23: Data in Ink Sense Page 0x28: Data in Print Unit Information Data Page 0x2a: Data in MG Encode Unit Information Data page 0x2b: Data in Adjust Print Position Page 0x3a: Data in JIS Mode Page pbyBuffer: Pointer to the memory to store data. The size of memory must be enough to store data. 4 byte Mode Parameter Header is added before the Mode Sense Data. So the size of memory must greater than (4 + Size of Mode Sense Data) bytes. <p>Note) Data format is explained at “1) Mode Sense Data format” in this chapter.</p>
2	int CXCMD_ModeSelect (int iSlot, int iID, int iPage, BYTE *pbyData)	<p>Issue Mode Select command to the printer and change setting of the printer. New setting is stored non volatile memory in the printer.</p> <p>iPage: Type of data 0x23: Data for Ink Sense Page 0x28: Data for Print Unit Information Data Page 0x2a: Data for MG Encode Unit Information Data page 0x2b: Data for Adjust Print Position Page 0x3a: Data for JIS Mode Page</p> <p>pbyData: Pointer to the memory which Mode Select Data is set. 4bytes vacant area must be prepared at the top of memory. As it is used to make Mode Parameter Header by CXCMD_ModeSelect(), Mode Select Data must be set from 5th bytes of the memory.</p> <p>Note) Data format is explained at “1) Mode Sense / Mode Select Data format” in this chapter.</p>

1) Mode Sense / Mode Select Data format

(1) Mode Parameter Header

Byte \ Bit	7	6	5	4	3	2	1	0
0	Mode Parameter Length (Used when Mode Select)							
1	Reserved							
2	Reserved	0	0	1	Reserved			
3	Block Descriptor Length (00H)							

The data of Mode Parameter Header is filled by the function CXCMDI_ModeSelect().

(2) Ink Sense Page Data format (Page Code=0x23)

Byte \ Bit	7	6	5	4	3	2	1	0
*4	0	0	Page Code (23H)					
*5	Page Specific Parameter Length (26H)							
6	Ink Code(Read Only)							
*7	Reserved							
8	(MSB) Number of Set of Ink Pannel(Binary, Read Only) (LSB)							
9								
10-11	Reserved							
12	Lot Number(ASCII, Read Only) Only top of 6 characters are effective. Remainder is filled with NULL							
43								

The data marked by * is filled by the function CXCMDI_ModeSelect().

Ink Code (Read Only: The type of ink which is attached to the printer):

0: YMC-K-OP

1: K

3: YMC-K-UV

4: YMC-K-OP-K

252: Unknown Ink (Invalid TAG Data)

253: Unknown Ink (Communication error with TAG)

254: Unknown Ink (No TAG)

255: Unknown Ink (Communication error with Reader/Writer)

* If Ink Cod is from 252 to 255, Lot Number is filled with NULL.

(3) Print Unit Information Data Sense Page Data format(Page Code=0x28)

Bit Byte	7	6	5	4	3	2	1	0
*4	0	0	Page Code (28H)					
*5	Page Specific Parameter Length (30H)							
6-7	Reserved							
8	(MSB)	Basic Resolution X (300, Read Only)						(LSB)
9								
10	(MSB)	Basic Resolution Y (300, Read Only)						(LSB)
11								
12-15	Reserved							
16	(MSB)	Card Size X (1013, Read Only)						(LSB)
17								
18	(MSB)	Card Size Y (639, Read Only)						(LSB)
19								
20	Reserved							
21	MG Mode							
22	Contact IC Mode(Read Only)							
23	Contactless IC Mode(Read Only)							
24	Laminator Mode(Read Only)							
25	Reserveed							
26	Card Code(Read Only)							
27	Reserved							
28	(MSB)	Print Position X(0, Read Only)						(LSB)
29								
30	(MSB)	Print Position Y(0, Read Only)						(LSB)
31								
32	(MSB)	Print Size X (1006, Read Only)						(LSB)
33								
34	(MSB)	Print Size Y (633, Read Only)						(LSB)
35								
36-37	Reserved							
38	YMC Level							
39	YMC Print Mode							
40-44	Reserved							
45	Resin Black Level							

46	Resin Black Print Mode
47	OP Level
48	UV Level
48-53	Reserved

The data marked by * is filled by the function CXCMDI_ModeSelect().

Basic Resolution X = Horizontal resolution in DPI.

Basic Resolution Y = Vertical resolution in DPI

Card Size X = Horizontal card size in pixels of printer

Card Size Y = Vertical card size in pixels of printer

MG Mode;

0: No MAG encoder 1: ISO MAG Encoder is installed

2: JIS MAG Encoder is installed

Contact IC Mode; 0: Not Installed 1: Installed

Contactless IC Mode; 0: Not Installed 1: Installed

Laminator Mode; 0: Not Installed 1: Installed

Card Code (Kind of card);

0: Standard size

Print Position X = Printable X coordinates from the left side of card edge in printer pixels.

Print Position Y = Printable Y coordinates from the upper side of card edge in printer pixels.

Print Size X = Printable width in pixels of printer.

Print Size Y = Printable height in pixels of printer.

YMC Level(Energy to print YMC ink on the card);

0:Lowest ~ 6:Highest (Standard value is 3)

YMC Print Mode;

0:Standard 1:Fine

Resin Black Level(Energy to print K ink on the card);

0:Lowest ~ 6:Highest (Standard value is 3)

Resin Black Print Mode;

0:Standard 1:Fine

OP Level(Energy to print OP ink on the card);

0:Lowest ~ 6:Highest (Standard value is 3)

UV Level (Energy to print UV ink on the card);

0:Lowest ~ 6:Highest (Standard value is 3)

(4) MG Encode Unit Information Data Page Data format(Page Code=0x2A)

Byte \ Bit	7	6	5	4	3	2	1	0
*4	0	0	Page Code (2AH)					
*5	Page Specific Parameter Length (08H)							
6	ISO Mode							
7-10	Reserved							
11	Retry Count							
12-13	Reserved							

The data marked by * is filled by the function CXCMDI_ModeSelect().

ISO Mode(Type of ISO MAG encoder)

0: No encoder

1: 300 Oe (Lo-Co)

2: 2750 Oe (Hi-Co)

Retry Count = Auto retry count when MAG access error happens. It must be less than 4.

(5) Adjust Print Position Page Data format(Page Code=0x2E)

Byte \ Bit	7	6	5	4	3	2	1	0
*4	0	0	Page Code (2BH)					
*5	Page Specific Parameter Length (06H)							
6	Reserved							
7	Offset X							
8	Reserved							
9	Offset Y							
10-11	Reserved							

The data marked by * is filled by the function CXCMDI_ModeSelect().

Offset X = Adjust Horizontal print beginning position. The value from 0 to 14 can be set.

Offset Y = Adjust Vertical print beginning position. The value from 0 to 14 can be set.

* Standard value is 7.

(6) MAG JIS Mode Page Data format(Page Code=0x3A)

Byte \ Bit	7	6	5	4	3	2	1	0
*4	0	0	Page Code (3AH)					
*5	Page Specific Parameter Length (08H)							
6	Reserved							
7	JIS Mode							
8-13	Reserved							

The data marked by * is filled by the function CXCMDI_ModeSelect().

JIS Mode(Type of JIS MAG encoder)

0: No JIS MAG Encoder 1: Lo-Co 2:Hi-Co

6. How to encode with Printer Driver

The printer drivers offers a function for encoding. Normally, It is not possible to pass the encode data to the printer driver. We offers two kinds of special way to it to the printer driver.

6.1 Inline Encoding

Encoding data can be passed to the printer driver as print data. A text preceding by the predefined prefix is not printed but encoded. If the text is regarded as the encoding data, both prefix and the text will be not printed. Following table shows the prefix and its meaning.

No	Prefix	Max Length	Text after prefix
1	~?0	69	Text is processed as the data for JIS-2 MAG encoding.
2	~?1	76	Text is processed as the data for ISO track1 MAG encoding
3	~?2	37	Text is processed as the data for ISO track2 MAG encoding
4	~?3	104	Text is processed as the data for ISO track3 MAG encoding
5	~?4		Text is processed as the data for the contact IC encoding
6	~?5		Text is processed as the data for the non-contact IC encoding

Encoding Prefix

*Column “Max Length” shows the maximum number of MAG encoding characters.

Note:

- Encoding setting of the printer driver property sheet must be enabled.
- The prefix and text must be successive, and their font must be same.
- Effective code for MG encoding is shown at “Appendix 2. ASCII Code Table and Magnetic Data”.
- In JIS-2 encoding, JIS Katakana characters are transformed to ASCII characters by inserting SI / SO control code by the printer driver.
- In case of both Contac IC encoding and Non-Contact IC encoding, the data being passed to IC Encode DLL(See 6.3) is ASCII single byte code. If other characters than ASCII are described, the value being passed to Encode DLL will not be guaranteed.

Following figure is an example for ISO MAG Track1 encoding. If following picture is printed, “12345678” will be encoded and “~?112345678” will be not printed.



6.2 Encoding by using ExtEscape()

You can pass the encoding data to the printer driver by using ExtEscape() function of WIN32 API.

(1) Parameter to ExtEscape() function

```
int ExtEscape (
    HDC          hdc,           // handle to the device context.
    Int          nEscape,       // Escape ID
    int          cbInput,       // size of encoding data
    LPCSTR       lpzInData,     // encoding data
    int          cbOutput, // unused.
    LPSTR        lpzOutData    // unused.
);
```

List of Escape ID

No	Escape ID	Explanation
1	9010	Non-contact IC encoding.
2	9011	Contact IC encoding.
3	9020	Magnetic encoding [JIS]
4	9021	Magnetic encoding [ISO 1st track]
5	9022	Magnetic encoding [ISO 2nd track]
6	9023	Magnetic encoding [ISO 3rd track]

(2) Return Code from ExtEscape() function

It'll return greater than zero if the function is successful. It means not the result of encoding but the result of sending encoding data.

Note:

- ExtEscape() must be called between StartPage() and EndPage(). And if both sides are printed, ExtEscape() must be done at the 1st printing side.
- Encode setting of the printer driver property sheet must be enabled.

6.3 How to make IC Encoding program

The driver does not offer the actual IC encoding function. When IC Encoding data is passed to the printer driver, it calls IC Encode DLL after making the printer ready for IC Encoding.

If you would like to do IC encoding, you must make IC Encode DLL and locate it in the system32 folder.

(1) File name of IC Encode DLL

PDR01IC0.DLL : for Non-contact IC encoding

PDR01IC1.DLL : for Contact IC encoding

(2) Function Prototype of IC Encode DLL

```
int stdcall Encode (  
    LPCSTR      lpSzInData,      // encoding data  
    int         cbInput,          // size of encoding data  
);
```

(3) Return Code from IC Encode DLL

It must return zero when the function ends successfully. If not zero is returned, Printer Driver will discharge the card.

Note:

The processing of the printer driver to the return code from IC Encode DLL is changed as follows. This is applied to the printer driver version 3.0.0.7 and the later.

- 0: Success. The printer driver continues printing.
- 0x1001: Fail. The printer driver does not display an error dialog. It discharges the card and retries on a new card.
- 0x1002: Fail. The printer driver does not display an error dialog. It discharges the card and cancels the current Print Job.
- Negative, and other positive value than above: The printer driver displays an error dialog, and does further processing according to the operator's choice.

Note:

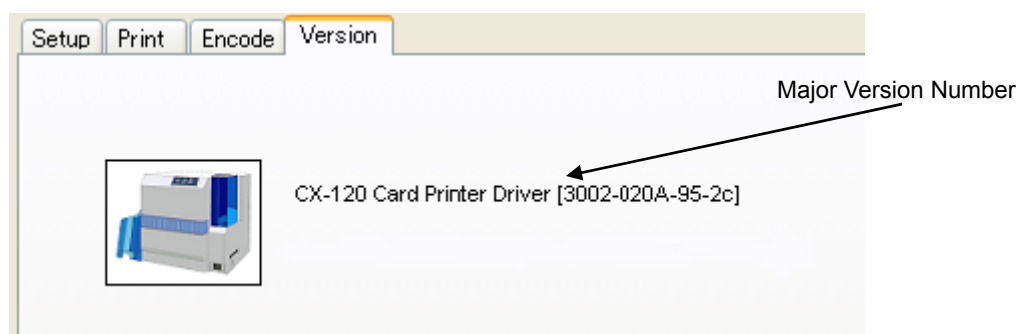
- The IC encoding by multiple printer drivers in a PC is not supported.

7. How to change the driver setting

The setting of the printer driver can be referred and changed by using ExtEscape() function.

Note:

- 1) This function is available for the driver which is the major version number is 3 or more. The major version number is shown on Version Tab sheet of Printer Driver dialog.
- 2) You must call ExtEscape() function from CreateDC() to StartDoc() for this purpose.
- 3) The modified setting is effective till the printer handle is destroyed by DeleteDC().



Version Tag Screen

7.1 How to use ExtEscape()

```
int ExtEscape (  
    HDC          hdc,           // handle to the device context.  
    Int          nEscape,       // Escape ID  
    int          cbParameter,   // size of the parameter data  
    LPCSTR       lpzParameter,  // pointer to the parameter data  
    int          cbResult,      // size of the result area  
    LPSTR        lpzResult     // pointer to the result area  
);
```

- nEscape : Set 9100 for this purpose.
- cbParameter : Set the length of Parameter in byte.
- lpzParameter : Pointer to the Parameter memory
- cbResult : Set the length of Result memory in byte.
- lpzResult : Pointer to the Result memory.

7.2 Change the setting

1) Format of the parameter

Name	Command Code	ID	Size	Data(New value)
Length in byte	1	2	2	n
Value	'S'(0x53)	*1	*2	*1

*1: Refer to “7.5 About the parameter”.

*2: Data length in byte.

2) Format of the result

(1) In case of success

Name	Error Code	Reserved
Length in byte	1	4
Value	0x00	Not defined

(2) In case of error

Name	Error Code	Error Code-A	Error Code-B
Length in byte	1	2	2
Value	0xff	*1	*1

*1: Refer to “7.6 About the error code”.

7.3 Get the current setting

1) Format of the parameter

Name	Command Code	ID
Length in byte	1	2
Value	'G'(0x47)	*1

*1: Refer to “7.5 About the parameter”.

2) Format of the result

(1) In case of success

Name	Error Code	ID	Size	Data(Current value)
Length in byte	1	2	2	n
Value	0x00	*1	*2	*1

*1: Refer to “7.5 About the parameter”.

*2: Data length in byte.

(2) In case of error

Name	Error Code	Error Code-A	Error Code-B
Length in byte	1	2	2
Value	0xff	*1	*1

*1: Refer to “7.6 About the error code”.

7.4 Programming sample

1) Change the setting of “Number of copies” to 100

```
int            escape_id;
unsigned short id, size;
unsigned long  data;
unsigned char  in[9], out[5];
int            ret;
unsigned short error_code;

escape_id = 9100; id = 257; size = 4; data = 100;
in[0]      = 'S';
in[1]      = (unsigned char)((id >> 8) & 0xFF);
in[2]      = (unsigned char)(id);
in[3]      = (unsigned char)((size >> 8) & 0xFF);
in[4]      = (unsigned char)(size);
in[5]      = (unsigned char)((data >> 24) & 0xFF);
in[6]      = (unsigned char)((data >> 16) & 0xFF);
in[7]      = (unsigned char)((data >> 8) & 0xFF);
in[8]      = (unsigned char)(data);

ret = ExtEscape ( hDC, escape_id, sizeof(in), (const char*)in, sizeof(out), (char*)out );
if (ret > 0) {          // Succeed in calling ExtEscape()
    if (out[0] == 0x00) {    // Succeed
        ;
    } else {                // Error happens in the driver
        // Get error code
        error_code = (unsigned short)((unsigned short)out[1] << 8 | out[2]);
    }
} else {                //Fail in the ExtEscape()
    ;
}
```

2) Refer to the setting of “Number of copies”.

```
int            escape_id;
unsigned short id, size;
unsigned long  data;
unsigned char  in[3], out[9];
int            ret;
unsigned short error_code;
```

```
escape_id = 9100; id = 257;
```

```
in[0]    = 'G';
in[1]    = (unsigned char)((id >> 8) & 0xFF);
in[2]    = (unsigned char)(id);
```

```
ret = ExtEscape ( hDC, escape_id, sizeof(in), (const char*)in, sizeof(out), (char*)out );
if (ret > 0) {          // Succeed in calling ExtEscape()
    if (out[0] == 0x00) {    // Succeed
        size = (unsigned short)((unsigned short)out[3] << 8 | out[4]);
        data = (unsigned long)((unsigned long)out[5] << 24 | (unsigned long)out[6] << 16 |
            (unsigned long)out[7] << 8 | out[8]);
    } else {                // Error happens in the driver
        // Get error code
        error_code = (unsigned short)((unsigned short)out[1] << 8 | out[2]);
    }
} else {                  //Fail in the ExtEscape()
    ;
}
```

7.5 About the parameter

Note: All value must be set with Big Endian.

Item Name	ATR	ID	Size	Data	Explanation
Number of copies	R/W	0x0101	0x0004	0x00000001~ 0x000003E7	The value must be from 1 to 999.
Card Load/Eject settings	R/W	0x0103	0x0004	0x00000000	Not turn the card.
				0x00000001	Turn the card before discharging
				0x00000002	Turn the card after loading
				0x00000003	Turn the card after loading and turn the card before discharging.
Print Print on both sides Print the back side first	R/W	0x0201	0x0004	0x00000000	Not print
				0x00000010	Single side printing.
				0x00000020	Both side printing. Print front side fast.
				0x00000021	Both side printing. Print back side fast.
Print mode [Front side]	R/W	0x0202	0x0004	0x00000001	Use YMC ink for front side printing.
				0x00000002	Use K ink for front side printing.
				0x00000003	Use YMCK ink for front side printing.
				0x00000011	Use YMC+OP ink for front side printing
				0x00000012	Use K+OP ink for front side printing
				0x00000013	Use YMCK+OP ink for front side printing
Print mode [Back side]	R/W	0x0203	0x0004	0x00000001	Use YMC ink for back side printing.
				0x00000002	Use K ink for back side printing.
				0x00000003	Use YMCK ink for back side printing.
				0x00000011	Use YMC+OP ink for back side printing
				0x00000012	Use K+OP ink for back side printing
				0x00000013	Use YMCK+OP ink for back side printing

Item Name	ATR	ID	Size	Data	Explanation
Using of Resin K ink [Front side]	R/W	0x0204	0x0004	0x00000010	Print black text of front side with K ink.
				0x00000011	Print black text of front side with K and print its background with YMC.
				0x00000020	Print black color of front side with K ink.
				0x00000021	Print black color of front side with K and print its background with YMC...
				0x00000040	Use page split function.
Using of Resin K ink [Back side]	R/W	0x0205	0x0004	0x00000010	Print black text of back side with K ink.
				0x00000011	Print black text of back side with K and print its background with YMC.
				0x00000020	Print black color of back side with K ink.
				0x00000021	Print black color of back side with K and print its background with YMC...
				0x00000040	Use page split function.
Rotate by 180 [Front side]	R/W	0x0206	0x0004	0x00000000	Not rotate the image of front side.
				0x00000001	Make the front side image up side down
Rotate by 180 [Back side]	R/W	0x0207	0x0004	0x00000000	Not rotate the image of back side.
				0x00000001	Make the back side image up side down
Magnetic encoding	R/W	0x0301	0x0004	0x00000000	Disable MG encoding.
				0x00000010	Enable MG encoding.
				0x00000011	Turn the card after MG encoding
Non-contact/Contact IC encoding	R/W	0x0302	0x0004	0x00000000	Disable IC encoding.
				0x00000010	Enable Contact IC encoding.
				0x00000011	Turn the card after Contact IC encoding.
				0x00000020	Enable No-Contact IC encoding.
				0x00000021	Turn the card after No-Contact IC encoding.
				0x00000030	Enable both Contact and No-Contact IC encoding.
				0x00000031	Turn the card after both Contact and No-Contact IC encoding.

7.6 About the error code

Note: All value is set with Big Endian.

No	Error code A	Error code B	Explanation
1	0x0901	0x0000	Invalid parameter was passed.
2	0x0902	*	Length of data area is not enough. Required length is returned at Error code B.

<Appendix 1 Error Code table>

The configuration of error is shown in the table bellow. The error code in the table is transformed to positive by the calculation “ (-1) * (Error code)”.

Error Code(HEXA)				Explanation
Bit31-24	Bit23-16	Bit15-8	Bit7-0	
0x01	Sense Key	ASC	ASCQ	Error code from Card Printer: From bit 0 to bit 23 is an error code sent from card printer.
0x02	00	XXX		Error of CX Port Manager. XXX means the contents of error.
0x02	01	XXX		Error of CX Port Manager: XXX is an Invalid SRB status value from Manager.
0x02	02	XXX		Error of CX Port Manager: XXX is an Invalid HA status value from Manager.
0x02	03	XXX		Error of CX Port Manager: XXX is an Invalid Target status value from Manager r.
0x09	XXX			Other Error. XXX means the contents of error.

A) Printer Error(0x01xxxxxx)

S K Y	A S C	A S Q	Explanation	Note
02	04	00	Not ready because of mechanical busy.	The printer is on working and cannot accept commands. This error is not returned as error from DLL functions. It is returned as busy
02	04	01	Not ready because the printer is on the way of initializing.	
02	04	80	Not ready because of door open	
02	04	81	Not ready because of no cleaning roller	
02	04	82	Not ready because of MG head cleaning.	
02	04	83	Not ready because the printer is in the Self-Test mode.	
02	04	84	Not ready because the printer is in the down load mode.	This mode will be continued till the printer is re-powered.
02	04	85	Not ready because the printer is in the offline mode.	This mode will be continued till the printer is re-powered.
02	3A	00	No card in the hopper.	
02	3A	80	Laminator is Not Ready	An error happens in Laminator. You can know what kind of error happens by seeing LCD panel on the Laminator. This error is detected by the printer only when the printer passes the card printed successfully to Laminator. So the card in the printer has only to be laminated. Lamination starts automatically when the problem in the laminator is solved.
02	FE	00	Certification failure	Password function is enabled and Certification fails.

S K Y	A S C	A S Q	Explanation	Note
03	0C	00	MG writing error happens.	
03	11	00	MG reading error happens.	
03	36	00	Ink failure.	It means that the communication between INK TAG fails. Although this state can be restored by pressing RESET switch of the printer, it cannot say further printing goes well because the printer does not know what kind of ink is attached
03	36	01	The diameter of the ink ribbon is too large.	It happens if the diameter at the take-up side bobbin becomes greater than 6cm. It is required to make the diameter less.
03	3B	81	Card JAM was detected during card loading.	It is required to confirm the card is set correctly in the card cassette.
03	3B	82	Card JAM was detected during card turning over.	It is required to remove the card from printer unit.
03	3B	83	Card JAM was detected during card transporting or printing.	It is required to remove the card from printer unit.
03	3B	84	Card jam was detected in the MG encoder	It is required to remove the card from printer unit.
03	3B	85	Card JAM was detected during card unloading.	It is required to remove the card from printer unit.
03	3B	86	Card JAM was detected in Laminator unit	This error happens when Card Jam in the laminator happens on 1st side laminating in case of both side laminating. It is required to initialize the printer by opening and closing printer front door after laminator becomes ready.
03	3B	88	Card JAM was detected during card unloading to the NG card exit.	It is required to remove the card from printer unit.

S K Y	A S C	A S Q	Explanation	Note
04	08	82	Communication error happens between Laminator unit	It means printer initialization fails by means of Laminator Not Ready. To restore from this state, both the printer and Laminator must be initialized.
04	36	81	Ink mark is not detected	To restore from this state, Ink must be changed.
04	40	01	Unit EEPROM failure	The printer must be re-powered to recover from this error.
04	40	02	Head EEPROM failure	The printer must be re-powered to recover from this error.
04	44	00	Hardware error happens.	The printer must be re-powered to recover from this error.
04	44	80	Firmware error happens in mechanical control process.	The printer must be re-powered to recover from this error.
04	44	81	Firmware error happens in system control process.	The printer must be re-powered to recover from this error.
04	44	82	Firmware error happens in pixel control process.	The printer must be re-powered to recover from this error.
04	44	83	Firmware error happens in host interface process.	The printer must be re-powered to recover from this error.
04	AB	00	MG unit mechanical error happens.	The printer must be re-powered to recover from this error.
04	C0	00	Flip unit mechanical error happens.	The printer must be re-powered to recover from this error.
04	C1	00	Head mechanical error happens.	The printer must be re-powered to recover from this error.
04	F6	00	Temperature in the printer is too hot.	This error never happens on the way of printing. It will be reported after printing ends. The printer must be re-powered to recover from this error.
04	F6	01	Temperature in the printer is too cool.	This error never happens on the way of printing. It will be reported after printing ends. The printer must be re-powered to recover from this error.
04	F8	00	Temperature of the thermal head is too hot.	This error never happens on the way of printing. It will be reported after printing ends. The printer must be re-powered to recover from this error.
04	F8	01	Temperature around the thermal head is too cool.	This error never happens on the way of printing. It will be reported after printing ends. The printer must be re-powered to recover from this error.

S K Y	A S C	A S Q	Explanation	Note
05	1A	00	Length of Parameter list is invalid.	
05	20	00	Invalid operation code in CDB	
05	20	80	IC encoder is not installed	
05	20	81	MG encoder is not installed	
05	24	00	Invalid field in CDB	
05	24	01	Invalid color code in CDB.	
05	24	80	USB failure	
05	25	00	Logical unit not supported	
05	26	01	Parameter is not supported	
05	26	02	Parameter value is invalid	
05	26	80	MG encoding data is invalid.	
05	26	81	Download data is invalid	
05	2C	00	Command sequence error	
06	28	00	CX-120 has been initialized.	
06	29	00	CX-120 has been powered.	
42	00	81	Ink end.	
42	00	82	End of Laminator Film.	The card in the printer is printed correctly when this error happens. Lamination begins automatically after the problem of laminator solves.

B) Driver Error (0x02xxxxxx)

Bit31-24 (0x02)	Bit23-16	Bit15-0	Explanation
0x02	00	1	Not enough memory
0x02	00	2	CX Port Manager is busy, and command cannot be accepted.
0x02	00	3	Command was aborted.
0x02	00	4	Time out
0x02	00	5	No SCSI card
0x02	00	6	CX Port Manager can not work. This means that the DLL of CX Port Manager could not be loaded. This is generated by CX-120 Control DLL.

Bit31-24 (0x02)	Bit23-16	Bit15-0	Explanation
0x02	01	XXXX	XXX is an Invalid SRB status value from CX Port Manager.
0x02	02	XXXX	XXX is an Invalid HA status value from CX Port Manager.
0x02	03	XXXX	XXX is an Invalid Target status value from CX Port Manager.

C) Others (0x09xxxxxx)

Bit31-24 (0x09)	Bit23-0	Explanation
0x09	1	Invalid parameter, such as NULL pointer
0x09	2	No printer is found.
0x09	3	Not enough memory
0x09	4	File Operation Error: fail to read file, or file content is wrong
0x09	5	Content of the DC is invalid: fail to get image from DC

<Appendix 2 ASCII Code Table and Magnetic Data >

4 unit code					
b4	b3	b2	b1		
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	2	2
0	0	1	1	3	3
0	1	0	0	4	4
0	1	0	1	5	5
0	1	1	0	6	6
0	1	1	1	7	7
1	0	0	0	8	8
1	0	0	1	9	9
1	0	1	0	A	:
1	0	1	1	B	;
1	1	0	0	C	<
1	1	0	1	D	=
1	1	1	0	E	>
1	1	1	1	F	?

Note) Special code

No.	Meaning	Character
1	Start Code	;
2	End Code	?
3	Separate Code	=
4	Code for hardware control	: < >

Start Code and End Code must not be used as MG Encoding data.

6 unit code									
					0	0	1	1	
					0	1	0	1	
b4	b3	b2	b1		0	1	2	3	
0	0	0	0	0		0	@	P	
0	0	0	1	1	!	1	A	Q	
0	0	1	0	2	“	2	B	R	
0	0	1	1	3	#	3	C	S	
0	1	0	0	4	\$	4	D	T	
0	1	0	1	5	%	5	E	U	
0	1	1	0	6	&	6	F	V	
0	1	1	1	7	‘	7	G	W	
1	0	0	0	8	(8	H	X	
1	0	0	1	9)	9	I	Y	
1	0	1	0	A	*	:	J	Z	
1	0	1	1	B	+	;	K	[
1	1	0	0	C	,	<	L	¥	
1	1	0	1	D	-	=	M]	
1	1	1	0	E	.	>	N	^	
1	1	1	1	F	/	?	O	_	

Note) Special code

No.	Meaning	Character
1	Start Code	%
2	End Code	?
3	Separate Code	^
4	Code for hardware control	! “ & ‘ * + , ; ; < = > @ [¥] _

Start Code and End Code must not be used as MG Encoding data.

7 unit code												
					b7	0	0	0	0	1	1	1
					b6	0	0	1	1	0	0	1
					b5	0	1	0	1	0	1	0
b4	b3	b2	b1		0	1	2	3	4	5	6	7
0	0	0	0	0				0	@	P	`	p
0	0	0	1	1				!	1	A	Q	a
0	0	1	0	2				"	2	B	R	b
0	0	1	1	3				#	3	C	S	c
0	1	0	0	4				\$	4	D	T	d
0	1	0	1	5				%	5	E	U	e
0	1	1	0	6				&	6	F	V	f
0	1	1	1	7				'	7	G	W	g
1	0	0	0	8				(8	H	X	h
1	0	0	1	9)	9	I	Y	i
1	0	1	0	A				*	:	J	Z	j
1	0	1	1	B				+	;	K	[{
1	1	0	0	C				,	<	L	¥	
1	1	0	1	D				-	=	M]	}
1	1	1	0	E				.	>	N	^	~
1	1	1	1	F				/	?	O	_	DEL

Note) Special code

No.	Meaning	Character
1	Start Code	0x7f (DEL)
2	End Code	0x7f (DEL)
3	Separate Code	^
4	Code for hardware control	! " & ' * + , : ; < = > @ [¥] _

Start Code and End Code must not be used as MG Encoding data.

