



API 레퍼런스 가이드
Label Printer SDK
iOS

Rev.1.05

SLP-DX420 / DX423

SLP-DX220 / DX223

SLP-TX400 / TX403

SLP-TX420 / TX423

SLP-TX220 / TX223

SLP-DL410 / DL413

SRP-770III

■ 목차

1. 매뉴얼 안내.....	4
1-1 지원 정보.....	4
1-2 지원 Printer / Interface 목록	4
1-3 지원 Method 목록	5
2. 상수값.....	9
2-1 Result Codes.....	9
2-2 Alignment	10
2-3 Barcode HRI.....	11
2-4 MaxiCode Modes	12
2-5 1D Barcode Types	13
2-6 Barcode Origin Point	14
2-7 Error Correction Level	14
2-8 Data Compression Method	15
2-9 QRCode Model	15
2-10 Code 49 Starting Mode	16
2-11 Codablock Mode.....	17
2-12 Check Digit Option	17
2-13 RSS Barcode Type	18
2-14 Rotation Degrees.....	19
2-15 Device Fonts.....	20
2-16 Vector Fonts	21
2-17 Draw Block Options.....	21
2-18 Draw Circle Sizes	22
2-19 International Character Set	23
2-20 Code Pages.....	24
3. LabelPrinterSDK Class Reference	26
3-1 Overview	26
3-2 Methods	26
3-2-1 open	26
3-2-2 close	26
3-2-3 connectWithAddress.....	27
3-2-4 isConnected	27
3-2-5 disconnect.....	27
3-2-6 disconnectWithTimeout	28
3-2-7 getDeviceFontList	28
3-2-8 getVectorFontList.....	28
3-2-9 getDrawBlockOptions	29
3-2-10 getDrawCircleSizes	29
3-2-11 getBarcodeType1D.....	29
3-2-12 getBarcodeHRI	30
3-2-13 doPrint.....	30
3-2-14 drawTextDeviceFont	31
3-2-15 drawTextVectorFont.....	33
3-2-16 drawBarcode1D	35
3-2-17 drawBarcodeMaxiCode	36
3-2-18 drawBarcodePDF417	37
3-2-19 drawBarcodeQRCode	39
3-2-20 drawBarcodeDataMatrix	40

3-2-21 drawBarcodeAztec	41
3-2-22 drawBarcodeCode49	43
3-2-23 drawBarcodeCodaBlock	44
3-2-24 drawBarcodeMicroPDF.....	46
3-2-25 drawBarcodeIMB.....	47
3-2-26 drawBarcodeMSI.....	48
3-2-27 drawBarcodePlessey	50
3-2-28 drawBarcodeTLC39.....	51
3-2-29 drawBarcodeRSS.....	52
4. Appendix	53
4-1 권장 사용 순서도	53
4-1-1 General	53
4-1-2 컨텐츠 출력 하기.....	54
4-1-3 2개 이상의 컨텐츠를 한 면에 출력 할 때.	55
4-2 Sample Codes	56
4-2-1 접속 / 접속 해제 하기.....	56
4-2-2 Printer의 Buffer에 그린 내용을 출력 하기	56
4-2-3 Printer의 Buffer에 사각형 상자 출력 하기.	56
4-2-4 Printer의 Buffer에 원형 상자 출력 하기.	57
4-2-5 Printer의 Buffer에 Device Font 출력 하기.....	57
4-2-6 Printer의 Buffer에 Vector Font 출력 하기.	57
4-2-7 Printer의 Buffer에 1D Barcode 출력 하기.....	58
4-2-8 Printer의 Buffer에 PDF417 Barcode 출력 하기.....	58
4-2-9 Printer의 Buffer에 QRCode Barcode 출력 하기.	58
4-2-10 Printer의 Buffer에 DataMatrix Barcode 출력 하기.....	59
4-2-11 Printer의 Buffer에 MaxiCode Barcode 출력 하기.....	59
4-2-12 Printer의 Buffer에 Aztec Barcode 출력 하기.....	59
4-2-13 Printer의 Buffer에 Code49 Barcode 출력 하기.....	60
4-2-14 Printer의 Buffer에 Codablock Barcode 출력 하기.	60
4-2-15 Printer의 Buffer에 MicroPDF417 Barcode 출력 하기.....	60
4-2-16 Printer의 Buffer에 IMB Barcode 출력 하기.....	61
4-2-17 Printer의 Buffer에 MSI Barcode 출력 하기.....	61
4-2-18 Printer의 Buffer에 Plessey Barcode 출력 하기.....	61
4-2-19 Printer의 Buffer에 TLC39 Barcode 출력 하기.	62
4-2-20 Printer의 Buffer에 RSS Barcode 출력 하기.....	62
4-3 Sample Application	63
4-3-1 접속 / 접속 해제	63
4-3-2 Device Font 출력.....	63
4-3-3 Vector Font 출력	64
4-3-4 Image 출력	64
4-3-5 다른 기능 이용하기.....	65
4-3-6 사각형 출력하기	65
4-3-7 원 출력하기	66

1. 매뉴얼 안내

이 SDK 매뉴얼에는 iOS용 응용 프로그램 개발에 필요한 Library 에 대한 내용이 기술되어 있습니다.

저희 (주)빅솔론은 제품의 기능과 품질 향상을 위하여 지속적인 개선을 하고 있습니다. 이로 인하여 제품의 사양과 매뉴얼의 내용은 사전 통보 없이 변경될 수 있습니다.

1-1 지원 정보

- iOS 6.0 이상

1-2 지원 Printer / Interface 목록

Method/Property	Ethernet	Wifi	Bluetooth
SLP-DX420	○	○	X
SLP-DX423	○	○	X
SLP-DX220	○	○	○
SLP-DX223	○	○	○
SLP-TX420	○	○	X
SLP-TX423	○	○	X
SLP-TX220	○	○	X
SLP-TX223	○	○	X
SLP-DL410	○	○	X
SLP-DL413	○	○	X
SLP-TX400	○	○	X
SLP-TX403	○	○	X
SRP-770III	○	○	X

1-3 지원 Method 목록

Method		SLP-DX420/DX423/DX220/DX223/ TX420/TX423/TX220/TX223/DL410/DL413/ TX400/TX403/SRP-770III
General	<i>open</i>	○
	<i>close</i>	○
Connection	<i>connectWithAddress:port:</i>	○
	<i>isConnected</i>	○
	<i>disconnect</i>	○
	<i>disconnectWithTimeout:</i>	○
Draw Text	<i>drawTextDeviceFont:</i> xPosition: yPosition: fontSelection: fontWidth: fontHeight: rightSideCharacterSpacing : fontRotation: reverse: bold: textAlignment:	○
	<i>drawTextVectorFont:</i> xPosition: yPosition: fontSelection: fontWidth: fontHeight: rightSideCharacterSpacing : fontRotation: reverse: bold: italic: textAlignment:	○

Method		SLP-DX420/DX423/DX220/DX223/ TX420/TX423/TX220/TX223/DL410/DL413/ TX400/TX403/SRP-770III
Draw Barcode	<i>drawBarcode1D:</i> xPosition: yPosition: barcodeType: widthNarrow: widthWide: height: hri: quietZoneWidth: rotation:	○
	<i>drawBarcodeMaxiCode:</i> xPosition: yPosition: mode:	○
	<i>drawBarcodePDF417:</i> xPosition: yPosition: maximumRowCount: maximumColumnCount: errorCorrectionLevel: dataCompressionMethod: printBarcodeText: barcodeOriginPoint: moduleWidth: barHeight: rotation:	○
	<i>drawBarcodeQRCode:</i> xPosition: yPosition: barcodeSize: model: errorColectionLevel: rotation:	○
	<i>drawBarcodeDataMatrix:</i> xPosition: yPosition: barcodeSize: reverse: rotation:	○
	<i>drawBarcodeAztec:</i> xPosition: yPosition: barcodeSize: extendedChannel: errorCorrectionLevel: menuSymbol: numberOfSymbols: optionalID: rotation:	○

Method		SLP-DX420/DX423/DX220/DX223/ TX420/TX423/TX220/TX223/DL410/DL413/ TX400/TX403/SRP-770III
Draw Barcode	drawBarcodeCode49: xPosition: yPosition: widthNarrow: widthWide: height: hri: startingMode: rotation:	○
	drawBarcodeCodaBlock: xPosition: yPosition: widthNarrow: widthWide: height: securityLevel: numberOfCharactersPerrow: mode: numberOfRowToEncode:	○
	drawBarcodeMicroPDF: xPosition: yPosition: moduleWidth: barcodeHeight: mode: rotation:	○
	drawBarcodeIMB: xPosition: yPosition: printBarcodeText: rotation:	○
	drawBarcodeMSI: xPosition: yPosition: widthNarrow: widthWide: height: checkDigitSelection: printCheckDigitInHRI: hri: rotation:	○
	drawBarcodePlessey: xPosition: yPosition: widthNarrow: widthWide: height: printCheckDigit: hri: rotation:	○

Method		SLP-DX420/DX420/DX220/DX223/ TX420/TX423/TX220/TX223/DL410/DL413/ TX400/TX403/SRP-770III
Draw Barcode	drawBarcodeTLC39: xPosition: yPosition: widthNarrow: widthWide: height: rowHeightOfMicroPDF417: narrowWidthOfMicroPDF417: rotation:	○
	drawBarcodeRSS: xPosition: yPosition: barcodeType: magnification: separatorHeight: barcodeHeight: segmentWidth: rotation:	○
drawBlock	drawBlock: startPosY: endPosX: endPosY: option: thickness:	○
drawCircle	drawCircle: startPosY: sizeSelection: multiplier:	○
drawImage	drawImage: stratPosX: startPosY: width:	○

2. 상수값

제공하는 SDK에서 사용하는 상수 값으로
“LabelPrinterSDK_Defines.h” 파일에 정의되어 있습니다.

2-1 Result Codes

Methods 에서 특정 기능을 수행 후 반환하는 결과 값 입니다.

```
typedef enum{
    _SDK_RESULT_SUCCESS                = 0x0000,
    _SDK_RESULT_FAIL                    = 0xF000,
    _SDK_RESULT_FAIL_NOT_SUPPORT_FUNCTION,
    _SDK_RESULT_FAIL_NO_OPEN           ,
    _SDK_RESULT_FAIL_OPEN_ALREADY      ,
    _SDK_RESULT_FAIL_NO_CONNECT        ,
    _SDK_RESULT_FAIL_CONNECT_ALREADY   ,
    _SDK_RESULT_FAIL_WRITE_ERROR       ,
    _SDK_RESULT_FAIL_READ_ERROR        ,
    _SDK_RESULT_FAIL_INVALID_PARAMETER ,
}_SDK_RESULT_CODES_;
```

각각의 설명은 아래와 같습니다.

Code	Value	Description
_SDK_RESULT_SUCCESS	0X0000	성공
_SDK_RESULT_FAIL	0XF000	실패
_SDK_RESULT_FAIL_NOT_SUPPORT_FUNCTION	0XF001	지원되지 않음
_SDK_RESULT_FAIL_NO_OPEN	0XF002	SDK가 OPEN 되지 않았습니다.
_SDK_RESULT_FAIL_OPEN_ALREADY	0XF003	SDK가 이미 OPEN 되었습니다.
_SDK_RESULT_FAIL_NO_CONNECT	0XF004	프린터가 접속되지 않았습니다.
_SDK_RESULT_FAIL_CONNECT_ALREADY	0XF005	프린터가 이미 Connected 상태 입니다.
_SDK_RESULT_FAIL_WRITE_ERROR	0XF006	데이터 전송에 실패했습니다.
_SDK_RESULT_FAIL_READ_ERROR	0XF007	데이터 수신에 실패했습니다.
_SDK_RESULT_FAIL_INVALID_PARAMETER	0XF008	잘못된 파라미터 입니다.

2-2 Alignment

정렬값을 정의 하는 속성 입니다.

```
typedef enum{
    _SDK_ALIGNMENT_LEFT           = 0,
    _SDK_ALIGNMENT_RIGHT         = 1,
    _SDK_ALIGNMENT_CENTER        = 2,
    _SDK_ALIGNMENT_STRING_FROM_RIGHT_2_LEFT = 2,
}__SDK_ALIGNMENTS;
```

각각의 설명은 아래와 같습니다.

Code	Value	Description
_SDK_ALIGNMENT_LEFT	0	왼쪽정렬
_SDK_ALIGNMENT_RIGHT	1	오른쪽 정렬
_SDK_ALIGNMENT_CENTER	2	가운데 정렬
_SDK_ALIGNMENT_STRING_FROM_RIGHT_2_LEFT	2	문자열을 오른쪽에서 왼쪽 방향으로 출력 합니다.

[Discussion]

_SDK_ALIGNMENT_CENTER는 Vector Font 에서 유효한 정렬 값입니다.

해당 값을 Device Font 출력에 사용 할 경우

_SDK_ALIGNMENT_STRING_FROM_RIGHT_2_LEFT와 같이 동작 할 수 있으며,

두 값의 Integer value 는 동일합니다.

2-3 Barcode HRI

HRI(Human Readable Interpretation) 가 지원되는 바코드 출력 시 HRI 의 위치와 Font 의 속성 입니다.

```
typedef enum{
    _SDK_BARCODE_HRI_NONE                = 0,
    _SDK_BARCODE_HRI_BELOW_FONTSIZE1    = 1,
    _SDK_BARCODE_HRI_ABOVE_FONTSIZE1    = 2,
    _SDK_BARCODE_HRI_BELOW_FONTSIZE2    = 3,
    _SDK_BARCODE_HRI_ABOVE_FONTSIZE2    = 4,
    _SDK_BARCODE_HRI_BELOW_FONTSIZE3    = 5,
    _SDK_BARCODE_HRI_ABOVE_FONTSIZE3    = 6,
    _SDK_BARCODE_HRI_BELOW_FONTSIZE4    = 7,
    _SDK_BARCODE_HRI_ABOVE_FONTSIZE4    = 8,
}__SDK_BARCODE_HRI;
```

각각의 설명은 아래와 같습니다.

Code	Value	Description
_SDK_BARCODE_HRI_NONE	0	HRI 를 사용하지 않습니다.
_SDK_BARCODE_HRI_BELOW_FONTSIZE1	1	HRI 위치 : 바코드의 하단 Font Size : 1
_SDK_BARCODE_HRI_ABOVE_FONTSIZE1	2	HRI 위치 : 바코드의 상단 Font Size : 1
_SDK_BARCODE_HRI_BELOW_FONTSIZE2	3	HRI 위치 : 바코드의 하단 Font Size : 2
_SDK_BARCODE_HRI_ABOVE_FONTSIZE2	4	HRI 위치 : 바코드의 상단 Font Size : 2
_SDK_BARCODE_HRI_BELOW_FONTSIZE3	5	HRI 위치 : 바코드의 하단 Font Size : 3
_SDK_BARCODE_HRI_ABOVE_FONTSIZE3	6	HRI 위치 : 바코드의 상단 Font Size : 3
_SDK_BARCODE_HRI_BELOW_FONTSIZE4	7	HRI 위치 : 바코드의 하단 Font Size : 4
_SDK_BARCODE_HRI_ABOVE_FONTSIZE4	8	HRI 위치 : 바코드의 상단 Font Size : 4

[Discussion]

파라미터로 사용시 유효한 값으로 변경되어 적용 될 수 있습니다.

2-4 MaxiCode Modes

Maxi code 바코드 출력 시 사용 하는 바코드 옵션에 대한 속성 입니다.

```
typedef enum{  
    _SDK_MAXICODE_MODE_0           = 0,  
    _SDK_MAXICODE_MODE_2           = 2,  
    _SDK_MAXICODE_MODE_3           = 3,  
    _SDK_MAXICODE_MODE_4           = 4,  
}_SDK_MAXICODE_MODE;
```

각각의 설명은 아래와 같습니다.

Code	Value	Description
_SDK_MAXICODE_MODE_0	0	MaxiCode Mode 0
_SDK_MAXICODE_MODE_2	2	MaxiCode Mode 2
_SDK_MAXICODE_MODE_3	3	MaxiCode Mode 3
_SDK_MAXICODE_MODE_4	4	MaxiCode Mode 4

2-5 1D Barcode Types

Code 바코드 출력 시 사용 하는 바코드 옵션에 대한 속성 입니다.

```
typedef enum{
    _SDK_BARCODE_TYPE_CODE39                = 0,
    _SDK_BARCODE_TYPE_CODE128              = 1,
    _SDK_BARCODE_TYPE_I2Of5                = 2,
    _SDK_BARCODE_TYPE_CODABAR              = 3,
    _SDK_BARCODE_TYPE_CODE93               = 4,
    _SDK_BARCODE_TYPE_UPC_A                = 5,
    _SDK_BARCODE_TYPE_UPC_E                = 6,
    _SDK_BARCODE_TYPE_EAN13                = 7,
    _SDK_BARCODE_TYPE_EAN8                 = 8,
    _SDK_BARCODE_TYPE_EAN128              = 9,
    _SDK_BARCODE_TYPE_CODE11               = 10,
    _SDK_BARCODE_TYPE_PLANET               = 11,
    _SDK_BARCODE_TYPE_INDUSTRIAL_2Of5     = 12,
    _SDK_BARCODE_TYPE_STANDARD_2Of5      = 13,
    _SDK_BARCODE_TYPE_LOGMARS              = 14,
    _SDK_BARCODE_TYPE_UPC_EAN_EXTENSIONS  = 15,
    _SDK_BARCODE_TYPE_POSTNET              = 16,
}__SDK_BARCODE_TYPE;
```

각각의 설명은 아래와 같습니다.

Code	Value	Description
_SDK_BARCODE_TYPE_CODE39	0	Code39
_SDK_BARCODE_TYPE_CODE128	1	Code128
_SDK_BARCODE_TYPE_I2Of5	2	Interleaved 2of5
_SDK_BARCODE_TYPE_CODABAR	3	Codabar
_SDK_BARCODE_TYPE_CODE93	4	Code93
_SDK_BARCODE_TYPE_UPC_A	5	UPC-A
_SDK_BARCODE_TYPE_UPC_E	6	UPC-E
_SDK_BARCODE_TYPE_EAN13	7	EAN13
_SDK_BARCODE_TYPE_EAN8	8	EAN8
_SDK_BARCODE_TYPE_EAN128	9	UCC/EAN128
_SDK_BARCODE_TYPE_CODE11	10	Code11
_SDK_BARCODE_TYPE_PLANET	11	Planet
_SDK_BARCODE_TYPE_INDUSTRIAL_2Of5	12	Industrial 2of5
_SDK_BARCODE_TYPE_STANDARD_2Of5	13	Standard 2of5
_SDK_BARCODE_TYPE_LOGMARS	14	Logmars
_SDK_BARCODE_TYPE_UPC_EAN_EXTENSIONS	15	UPC/EAN Extensions
_SDK_BARCODE_TYPE_POSTNET	16	Postnet

2-6 Barcode Origin Point

바코드 위치의 기준점에 대한 속성입니다.

```
typedef enum{
    _SDK_BARCODE_ORIGIN_POINT_CENTER           = 0,
    _SDK_BARCODE_ORIGIN_POINT_UpperLeft       = 1,
}__SDK_BARCODE_ORIGIN_POINT;
```

각각의 설명은 아래와 같습니다.

Code	Value	Description
_SDK_BARCODE_ORIGIN_POINT_CENTER	0	기준점을 바코드의 한가운데로 지정합니다.
_SDK_BARCODE_ORIGIN_POINT_UpperLeft	1	기준점을 바코드의 좌측 상단으로 지정 합니다.

2-7 Error Correction Level

바코드의 손상에 대비한 에러 보정의 단계에 대한 속성입니다.

숫자가 레벨이 높을수록 더 큰 손상에 대한 보정이 가능하지만, Data 허용 길이가 감소할 수 있습니다.

```
typedef enum{
    _SDK_ERROR_CORRECTION_LEVEL0           = 0,
    _SDK_ERROR_CORRECTION_LEVEL1         = 1,
    _SDK_ERROR_CORRECTION_LEVEL2         = 2,
    _SDK_ERROR_CORRECTION_LEVEL3         = 3,
    _SDK_ERROR_CORRECTION_LEVEL4         = 4,
    _SDK_ERROR_CORRECTION_LEVEL5         = 5,
    _SDK_ERROR_CORRECTION_LEVEL6         = 6,
    _SDK_ERROR_CORRECTION_LEVEL7         = 7,
    _SDK_ERROR_CORRECTION_LEVEL8         = 8,
}__SDK_ERROR_CORRECTION_LEVEL
```

각각의 설명은 아래와 같습니다.

Code	Value	Description
_SDK_ERROR_CORRECTION_LEVEL0	0	손상 보정 Level 0
_SDK_ERROR_CORRECTION_LEVEL1	1	손상 보정 Level 1
_SDK_ERROR_CORRECTION_LEVEL2	2	손상 보정 Level 2
_SDK_ERROR_CORRECTION_LEVEL3	3	손상 보정 Level 3
_SDK_ERROR_CORRECTION_LEVEL4	4	손상 보정 Level 4
_SDK_ERROR_CORRECTION_LEVEL5	5	손상 보정 Level 5
_SDK_ERROR_CORRECTION_LEVEL6	6	손상 보정 Level 6
_SDK_ERROR_CORRECTION_LEVEL7	7	손상 보정 Level 7
_SDK_ERROR_CORRECTION_LEVEL8	8	손상 보정 Level 8

2-8 Data Compression Method

Data 압축 속성 입니다.

```
typedef enum{
    _SDK_DATA_COMPRESSION_METHOD_TEXT           = 0,
    _SDK_DATA_COMPRESSION_METHOD_NUMERIC       = 1,
    _SDK_DATA_COMPRESSION_METHOD_BINARY        = 2,
}__SDK_DATA_COMPRESSION_METHOD;
```

각각의 설명은 아래와 같습니다.

Code	Value	Description
_SDK_DATA_COMPRESSION_METHOD_TEXT	0	2char / codeword
_SDK_DATA_COMPRESSION_METHOD_NUMERIC	1	2.93 char / codeword
_SDK_DATA_COMPRESSION_METHOD_BINARY	2	1.2bytes / codeword

2-9 QRCode Model

QR code 바코드 출력 시 사용 하는 바코드 옵션에 대한 속성 입니다.

```
typedef enum{
    _SDK_QRCODE_MODEL_1                       = 1,
    _SDK_QRCODE_MODEL_2                       = 2,
}__SDK_QRCODE_MODE;
```

각각의 설명은 아래와 같습니다.

Code	Value	Description
_SDK_QRCODE_MODEL_1	1	QR Model 1
_SDK_QRCODE_MODEL_2	2	QR Model 2

2-10 Code 49 Starting Mode

Code49 바코드 출력 시 사용 하는 Starting Mode 대한 속성 입니다.

```
typedef enum{
    _SDK_STARTINGMODE_REGULAR_ALPHANUMERIC           = 0,
    _SDK_STARTINGMODE_MULTIPLE_READ_ALPHANUMERIC     = 1,
    _SDK_STARTINGMODE_REGULAR_NUMERIC                = 2,
    _SDK_STARTINGMODE_GROUP_ALPHANUMERIC             = 3,
    _SDK_STARTINGMODE_REGULAR_ALPHANUMERIC_SHIFT1    = 4,
    _SDK_STARTINGMODE_REGULAR_ALPHANUMERIC_SHIFT2    = 5,
    _SDK_STARTINGMODE_AUTOMATIC                       = 7,
}_SDK_STARTING_MODE;
```

각각의 설명은 아래와 같습니다.

Code	Value	Description
_SDK_STARTINGMODE_REGULAR_ALPHANUMERIC	0	Regular Alphanumeric Mode
_SDK_STARTINGMODE_MULTIPLE_READ_ALPHANUMERIC	1	Multiple Read Alphanumeric
_SDK_STARTINGMODE_REGULAR_NUMERIC	2	Regular Numeric Mode
_SDK_STARTINGMODE_GROUP_ALPHANUMERIC	3	Group Alphanumeric Mode
_SDK_STARTINGMODE_REGULAR_ALPHANUMERIC_SHIFT1	4	Regular Alphanumeric Shift 1
_SDK_STARTINGMODE_REGULAR_ALPHANUMERIC_SHIFT2	5	Regular Alphanumeric Shift 2
_SDK_STARTINGMODE_AUTOMATIC	7	Automatic Mode

2-11 Codablock Mode

Coda block 바코드 출력 시 사용 하는 바코드 옵션에 대한 속성 입니다.

```
typedef enum{
    _SDK_CODABLACK_MODE_A           = 'A',
    _SDK_CODABLACK_MODE_E           = 'E',
    _SDK_CODABLACK_MODE_F           = 'F',
}_SDK_CODABLACK_MODE;
```

각각의 설명은 아래와 같습니다.

Code	Value	Description
_SDK_CODABLACK_MODE_A	'A'	Code 39 의 Character set 을 사용합니다.
_SDK_CODABLACK_MODE_E	'E'	Code 128 의 Character set 을 사용합니다.
_SDK_CODABLACK_MODE_F	'F'	Code 128 의 Character set 을 사용합니다. Function 1이 자동으로 추가 됩니다.

2-12 Check Digit Option

MSI 바코드 출력 시 선택 가능 한 Check digit 속성 입니다.

```
typedef enum{
    _SDK_CHECKDIGIT_NONE           = 0,
    _SDK_CHECKDIGIT_1MOD10         = 1,
    _SDK_CHECKDIGIT_2MOD10         = 2,
    _SDK_CHECKDIGIT_1MOD11_AND_1MOD_10 = 3,
}_SDK_CHECKDIGIT;
```

각각의 설명은 아래와 같습니다.

Code	Value	Description
_SDK_CHECKDIGIT_NONE	0	Check Digit 없음
_SDK_CHECKDIGIT_1MOD10	1	Check Digit 1 Mod 10
_SDK_CHECKDIGIT_2MOD10	2	Check Digit 2 Mod 10
_SDK_CHECKDIGIT_1MOD11_AND_1MOD_10	3	Check Digit 1 Mod 10

2-13 RSS Barcode Type

RSS 바코드 출력 시 바코드 타입에 대한 속성입니다.

```
typedef enum{
    _SDK_RSS_BARCODE_TYPE_RSS14                = 0,
    _SDK_RSS_BARCODE_TYPE_RSS14_TRUNCATED      = 1,
    _SDK_RSS_BARCODE_TYPE_RSS14_STACKED        = 2,
    _SDK_RSS_BARCODE_TYPE_RSS14_STACKED_OMNIDIRECTIONAL = 3,
    _SDK_RSS_BARCODE_TYPE_RSS_LIMITED          = 4,
    _SDK_RSS_BARCODE_TYPE_RSS_EXPANDED         = 5,
    _SDK_RSS_BARCODE_TYPE_UPC_A                = 6,
    _SDK_RSS_BARCODE_TYPE_UPC_E                = 7,
    _SDK_RSS_BARCODE_TYPE_EAN13                = 8,
    _SDK_RSS_BARCODE_TYPE_EAN8                = 9,
    _SDK_RSS_BARCODE_TYPE_UCC_EAN128_CC_A_B    = 10,
    _SDK_RSS_BARCODE_TYPE_UCC_EAN128_CC_C      = 11,
}__SDK_RSS_BARCODE_TYPE;
```

각각의 설명은 아래와 같습니다.

Code	Value	Description
_SDK_RSS_BARCODE_TYPE_RSS14	0	RSS14
_SDK_RSS_BARCODE_TYPE_RSS14_TRUNCATED	1	RSS14 truncated
_SDK_RSS_BARCODE_TYPE_RSS14_STACKED	2	RSS14 stacked
_SDK_RSS_BARCODE_TYPE_RSS14_STACKED_OMNIDIRECTIONAL	3	RSS14 Stacked omnidirectional
_SDK_RSS_BARCODE_TYPE_RSS_LIMITED	4	RSS limited
_SDK_RSS_BARCODE_TYPE_RSS_EXPANDED	5	RSS Expanded
_SDK_RSS_BARCODE_TYPE_UPC_A	6	RSS UPC A
_SDK_RSS_BARCODE_TYPE_UPC_E	7	RSS UPC E
_SDK_RSS_BARCODE_TYPE_EAN13	8	EAN13
_SDK_RSS_BARCODE_TYPE_EAN8	9	EAN 8
_SDK_RSS_BARCODE_TYPE_UCC_EAN128_CC_A_B	10	EAN128 CC-A/B
_SDK_RSS_BARCODE_TYPE_UCC_EAN128_CC_C	11	EAN128 CC-C

2-14 Rotation Degrees

출력물의 회전에 대한 속성입니다.

```
typedef enum{
    _SDK_ROTATION_DEGREES_0           = 0,
    _SDK_ROTATION_DEGREES_90         = 1,
    _SDK_ROTATION_DEGREES_180        = 2,
    _SDK_ROTATION_DEGREES_270        = 3,
}__SDK_ROTATION_DEGREES;
```

각각의 설명은 아래와 같습니다.

Code	Value	Description
_SDK_ROTATION_DEGREES_0	0	회전하지 않습니다.
_SDK_ROTATION_DEGREES_90	1	90도 회전합니다.
_SDK_ROTATION_DEGREES_180	2	190도 회전합니다.
_SDK_ROTATION_DEGREES_270	3	270도 회전 합니다.

2-15 Device Fonts

Device Font 속성 입니다.

```
typedef enum {
    _SDK_DEVICE_FONT_6PT           ='0',
    _SDK_DEVICE_FONT_8PT           ='1',
    _SDK_DEVICE_FONT_10PT          ='2',
    _SDK_DEVICE_FONT_12PT          ='3',
    _SDK_DEVICE_FONT_15PT          ='4',
    _SDK_DEVICE_FONT_20PT          ='5',
    _SDK_DEVICE_FONT_30PT          ='6',
    _SDK_DEVICE_FONT_14PT          ='7',
    _SDK_DEVICE_FONT_18PT          ='8',
    _SDK_DEVICE_FONT_24PT          ='9',
    _SDK_DEVICE_FONT_KOREAN1       ='a',
    _SDK_DEVICE_FONT_KOREAN2       ='b',
    _SDK_DEVICE_FONT_KOREAN3       ='c',
    _SDK_DEVICE_FONT_KOREAN4       ='d',
    _SDK_DEVICE_FONT_KOREAN5       ='e',
    _SDK_DEVICE_FONT_KOREAN6       ='f',
    _SDK_DEVICE_FONT_GB2312        ='m',
    _SDK_DEVICE_FONT_BIG5           ='n',
    _SDK_DEVICE_FONT_SHIFT_JIS     ='j',
}_SDK_DEVICE_FONT;
```

각각의 설명은 아래와 같습니다.

Code	Value	Description
_SDK_DEVICE_FONT_6PT	'0'	9 X 15 (dots)
_SDK_DEVICE_FONT_8PT	'1'	12 X 20 (dots)
_SDK_DEVICE_FONT_10PT	'2'	16 X 25 (dots)
_SDK_DEVICE_FONT_12PT	'3'	19 X 30 (dots)
_SDK_DEVICE_FONT_15PT	'4'	24 X 38 (dots)
_SDK_DEVICE_FONT_20PT	'5'	32 X 40 (dots)
_SDK_DEVICE_FONT_30PT	'6'	48 X 76 (dots)
_SDK_DEVICE_FONT_14PT	'7'	22 X 34 (dots)
_SDK_DEVICE_FONT_18PT	'8'	28 X 44 (dots)
_SDK_DEVICE_FONT_24PT	'9'	37 X 58 (dots)
_SDK_DEVICE_FONT_KOREAN1	'a'	16 X 16 (dots) (ascii 9 X 15)
_SDK_DEVICE_FONT_KOREAN2	'b'	24 X 24 (dots) (ascii 12 X 24)
_SDK_DEVICE_FONT_KOREAN3	'c'	20 X 20 (dots) (ascii 12 X 20)
_SDK_DEVICE_FONT_KOREAN4	'd'	26 X 26 (dots) (ascii 16 X 30)
_SDK_DEVICE_FONT_KOREAN5	'e'	20 X 26 (dots) (ascii 16 X 30)
_SDK_DEVICE_FONT_KOREAN6	'f'	38 X 38 (dots) (ascii 22 X 34)
_SDK_DEVICE_FONT_GB2312	'm'	24 X 24 (dots) (ascii 12 X 24)
_SDK_DEVICE_FONT_BIG5	'n'	24 X 24 (dots) (ascii 12 X 24)
_SDK_DEVICE_FONT_SHIFT_JIS	'j'	24 X 24 (dots) (ascii 12 X 24)

2-16 Vector Fonts

Vector Font 속성 입니다.

```
typedef enum {
    _SDK_VECTOR_FONT_ASCII           ='U',
    _SDK_VECTOR_FONT_KS5601         ='K',
    _SDK_VECTOR_FONT_BIG5           ='B',
    _SDK_VECTOR_FONT_GB2312        ='G',
    _SDK_VECTOR_FONT_SHIFT_JIS     ='J',
    _SDK_VECTOR_FONT_OCR_A         ='a',
    _SDK_VECTOR_FONT_OCR_B         ='b',
}_SDK_VECTOR_FONT;
```

각각의 설명은 아래와 같습니다.

Code	Value	Description
_SDK_VECTOR_FONT_ASCII	'U'	ASCII (1Byte code)
_SDK_VECTOR_FONT_KS5601	'K'	KS5601 (2Byte code)
_SDK_VECTOR_FONT_BIG5	'B'	BIG5 (2Byte code)
_SDK_VECTOR_FONT_GB2312	'G'	GB2312 (2Byte code)
_SDK_VECTOR_FONT_SHIFT_JIS	'J'	Shift-JIS (2Byte code)
_SDK_VECTOR_FONT_OCR_A	'a'	OCR-A (1Byte code)
_SDK_VECTOR_FONT_OCR_B	'b'	OCR-B (1Byte code)

2-17 Draw Block Options

Vector Font 속성 입니다.

```
typedef enum {
    _SDK_DRAW_BLOCK_OPTION_LINE_OVERWRITING      ='O',
    _SDK_DRAW_BLOCK_OPTION_LINE_EXCLUSIVE_OR    ='E',
    _SDK_DRAW_BLOCK_OPTION_LINE_DELETE          ='D',
    _SDK_DRAW_BLOCK_OPTION_SLOPE                ='S',
    _SDK_DRAW_BLOCK_OPTION_BOX                  ='B',
}_SDK_DRAW_BLOCK_OPTION;
```

각각의 설명은 아래와 같습니다.

Code	Value	Description
_SDK_DRAW_BLOCK_OPTION_LINE_OVERWRITING	'O'	Line Overwriting
_SDK_DRAW_BLOCK_OPTION_LINE_EXCLUSIVE_OR	'E'	Line Exclusive OR
_SDK_DRAW_BLOCK_OPTION_LINE_DELETE	'D'	Line Delete
_SDK_DRAW_BLOCK_OPTION_SLOPE	'S'	Slope(a oblique line)
_SDK_DRAW_BLOCK_OPTION_BOX	'B'	Box

2-18 Draw Circle Sizes

DrawCircle Method 사용시 크기와 관련된 속성 입니다.

```
typedef enum {
    _SDK_DRAW_CIRCLE_SIZE_40X40           = 1,    // 5(mm) diameter
    _SDK_DRAW_CIRCLE_SIZE_56X56           = 2,
    _SDK_DRAW_CIRCLE_SIZE_72X72           = 3,
    _SDK_DRAW_CIRCLE_SIZE_88X88           = 4,
    _SDK_DRAW_CIRCLE_SIZE_104X104         = 5,
    _SDK_DRAW_CIRCLE_SIZE_168X168         = 6,
}_SDK_DRAW_CIRCLE_SIZE;
```

각각의 설명은 아래와 같습니다.

Code	Value	Diameter (mm)	Width x Height(Dots)
_SDK_DRAW_CIRCLE_SIZE_40X40	1	5	40 × 40
_SDK_DRAW_CIRCLE_SIZE_56X56	2	7	56 × 56
_SDK_DRAW_CIRCLE_SIZE_72X72	3	9	72 × 72
_SDK_DRAW_CIRCLE_SIZE_88X88	4	11	88 × 88
_SDK_DRAW_CIRCLE_SIZE_104X104	5	13	104 × 104
_SDK_DRAW_CIRCLE_SIZE_168X168	6	21	168 × 168

2-19 International Character Set

International Character Set 속성 입니다.

```
typedef enum {
    _SDK_CONFIG_ICS_USA                = 0,
    _SDK_CONFIG_ICS_FRANCE             = 1,
    _SDK_CONFIG_ICS_GERMANY            = 2,
    _SDK_CONFIG_ICS_UK                 = 3,
    _SDK_CONFIG_ICS_DENMARK_I         = 4,
    _SDK_CONFIG_ICS_SWEDEN             = 5,
    _SDK_CONFIG_ICS_ITALY              = 6,
    _SDK_CONFIG_ICS_SPAIN_I            = 7,
    _SDK_CONFIG_ICS_NORWAY             = 8,
    _SDK_CONFIG_ICS_DENMARK_II        = 9,
    _SDK_CONFIG_ICS_JAPAN              = 10,
    _SDK_CONFIG_ICS_SPAIN_II           = 11,
    _SDK_CONFIG_ICS_LATIN_AMERICA      = 12,
    _SDK_CONFIG_ICS_KOREA              = 13,
    _SDK_CONFIG_ICS_SLOVENIA_CROATIA   = 14,
    _SDK_CONFIG_ICS_CHINA              = 15,
}_SDK_CONFIG_ICS;
```

각각의 설명은 아래와 같습니다.

Code	Value	Description
_SDK_CONFIG_ICS_USA	0	U.S.A
_SDK_CONFIG_ICS_FRANCE	1	France
_SDK_CONFIG_ICS_GERMANY	2	Germany
_SDK_CONFIG_ICS_UK	3	U.K
_SDK_CONFIG_ICS_DENMARK_I	4	Denmark I
_SDK_CONFIG_ICS_SWEDEN	5	Sweden
_SDK_CONFIG_ICS_ITALY	6	Italy
_SDK_CONFIG_ICS_SPAIN_I	7	Spain I
_SDK_CONFIG_ICS_NORWAY	8	Norway
_SDK_CONFIG_ICS_DENMARK_II	9	Denmark II
_SDK_CONFIG_ICS_JAPAN	10	Japan
_SDK_CONFIG_ICS_SPAIN_II	11	Spain II
_SDK_CONFIG_ICS_LATIN_AMERICA	12	Latin America
_SDK_CONFIG_ICS_KOREA	13	Korea
_SDK_CONFIG_ICS_SLOVENIA_CROATIA	14	Slovenia/Croatia
_SDK_CONFIG_ICS_CHINA	15	China

2-20 Code Pages

Code Page 속성 입니다.

```
typedef enum {  
    _SDK_CONFIG_CODEPAGE_CP437           = 0,  
    _SDK_CONFIG_CODEPAGE_CP850           = 1,  
    _SDK_CONFIG_CODEPAGE_CP852           = 2,  
    _SDK_CONFIG_CODEPAGE_CP860           = 3,  
    _SDK_CONFIG_CODEPAGE_CP863           = 4,  
    _SDK_CONFIG_CODEPAGE_CP865           = 5,  
    _SDK_CONFIG_CODEPAGE_CP81252         = 6,  
    _SDK_CONFIG_CODEPAGE_CP865_WCP1252   = 7,  
    _SDK_CONFIG_CODEPAGE_CP857           = 8,  
    _SDK_CONFIG_CODEPAGE_CP737           = 9,  
    _SDK_CONFIG_CODEPAGE_WPC1250         = 10,  
    _SDK_CONFIG_CODEPAGE_WPC1253         = 11,  
    _SDK_CONFIG_CODEPAGE_WPC1254         = 12,  
    _SDK_CONFIG_CODEPAGE_CP855           = 13,  
    _SDK_CONFIG_CODEPAGE_CP862           = 14,  
    _SDK_CONFIG_CODEPAGE_CP866           = 15,  
    _SDK_CONFIG_CODEPAGE_WCP1251         = 16,  
    _SDK_CONFIG_CODEPAGE_WCP1255         = 17,  
    _SDK_CONFIG_CODEPAGE_CP928           = 18,  
    _SDK_CONFIG_CODEPAGE_CP864           = 19,  
    _SDK_CONFIG_CODEPAGE_CP775           = 20,  
    _SDK_CONFIG_CODEPAGE_WCP1257         = 21,  
    _SDK_CONFIG_CODEPAGE_CP858           = 22,  
}_SDK_CONFIG_CODEPAGE;
```


각각의 설명은 아래와 같습니다.

Code	Value		
_SDK_CONFIG_CODEPAGE_CP437	0	CP437	U.S.A
_SDK_CONFIG_CODEPAGE_CP850	1	CP850	Latin1
_SDK_CONFIG_CODEPAGE_CP852	2	CP 852	Latin2
_SDK_CONFIG_CODEPAGE_CP860	3	CP 860	Portuguese
_SDK_CONFIG_CODEPAGE_CP863	4	CP 863	Canadian French
_SDK_CONFIG_CODEPAGE_CP865	5	CP 865	Nordic
_SDK_CONFIG_CODEPAGE_CP1252	6	WCP 1252	Latin I
_SDK_CONFIG_CODEPAGE_CP865_WCP1252	7	CP 865 + WCP 1252	European Combined
_SDK_CONFIG_CODEPAGE_CP857	8	CP 857	Turkish
_SDK_CONFIG_CODEPAGE_CP737	9	CP 737	Greek
_SDK_CONFIG_CODEPAGE_WPC1250	10	WCP 1250	Latin 2
_SDK_CONFIG_CODEPAGE_WPC1253	11	WCP 1253	Greek
_SDK_CONFIG_CODEPAGE_WPC1254	12	WCP 1254	Turkish
_SDK_CONFIG_CODEPAGE_CP855	13	CP 855	Cyrillic
_SDK_CONFIG_CODEPAGE_CP862	14	CP 862	Hebrew
_SDK_CONFIG_CODEPAGE_CP866	15	CP 866	Cyrillic
_SDK_CONFIG_CODEPAGE_WCP1251	16	WCP 1251	Cyrillic
_SDK_CONFIG_CODEPAGE_WCP1255	17	WCP 1255	Hebrew
_SDK_CONFIG_CODEPAGE_CP928	18	CP 928	Greek
_SDK_CONFIG_CODEPAGE_CP864	19	CP 864	Arabic
_SDK_CONFIG_CODEPAGE_CP775	20	CP 775	Baltic
_SDK_CONFIG_CODEPAGE_WCP1257	21	WCP1257	Baltic
_SDK_CONFIG_CODEPAGE_CP858	22	CP858	Latin 1 + Euro

3. LabelPrinterSDK Class Reference

Inherits from	NSObject
Framework	libLabelPrinterSDK.a
Declared	LabelPrinterSDK.h

3-1 Overview

libLabelPrinterSDK Class는 프린터를 제어하는 메인 객체입니다.

3-2 Methods

3-2-1 open

LabelPrinterSDK 클래스 사용을 위한 초기화를 진행 합니다.

[Syntax]

```
-(long) open;
```

[Return Value]

성공 시 `_SDK_RESULT_SUCCESS` 을 반환 합니다.

그 외 결과값에 대한 자세한 내용은 [2-1 Result Codes](#) 를 참조하십시오.

[Discussion]

LabelPrinterSDK 내 다른 Methods 를 사용하기 위해 반드시 선행되어 호출 되어야 합니다.

[See Also]

[2-1 Result Codes](#)

3-2-2 close

LabelPrinterSDK 클래스 사용을 종료 합니다.

[Syntax]

```
-(long) close;
```

[Return Value]

성공 시 `_SDK_RESULT_SUCCESS` 을 반환 합니다.

그 외 결과값에 대한 자세한 내용은 [2-1 Result Codes](#) 를 참조하십시오.

[Discussion]

이 Method 가 호출 되면 LabelPrinterSDK 내부의 리소스를 정리 합니다. 사용이 종료 된 후 반드시 호출 해야 합니다.

[See Also]

[2-1 Result Codes](#)

3-2-3 connectWithAddress

프린터에 접속을 시도 합니다.

[Syntax]

-(long) connectWithAddress:(NSString*)address port:(NSString*) port;

[Parameters]

address

wifi 프린터 : IP Address 를 입력 합니다.

Bluetooth 프린터 : Mac Address 를 입력 합니다.

port :

프린터의 통신 포트를 입력합니다.

당사의 프린터 통신 포트는 9100 입니다.

블루투스 프린터는 지원하지 않습니다.

[Return Value]

성공 시 `_SDK_RESULT_SUCCESS` 을 반환 합니다.

그 외 결과값에 대한 자세한 내용은 [2-1 Result Codes](#) 를 참조하십시오.

[See Also]

[2-1 Result Codes](#)

3-2-4 isConnected

프린터 접속상태를 확인 합니다.

[Syntax]

-(BOOL) isConnected;

[Return Value]

접속 상태 인 경우 YES 를 반환 합니다.

3-2-5 disconnect

프린터 접속을 해제 합니다.

[Syntax]

-(long) disconnect;

[Return Value]

성공 시 `_SDK_RESULT_SUCCESS` 을 반환 합니다.

그 외 결과값에 대한 자세한 내용은 [2-1 Result Codes](#) 를 참조하십시오.

[See Also]

[2-1 Result Codes](#)

3-2-6 disconnectWithTimeout

지정한 Timeout 내에 프린터 접속을 해제 합니다.

[Syntax]

```
-(long) disconnectWithTimeout:(NSInteger)timeout;
```

[Parameters]

timeout

Timeout 을 지정 합니다.(단위:초)

[Return Value]

성공 시 `_SDK_RESULT_SUCCESS` 을 반환 합니다.

그 외 결과값에 대한 자세한 내용은 [2-1 Result Codes](#) 를 참조하십시오.

[Discussion]

출력 버퍼에 Data가 남아있을 경우,

지정한 Timeout 동안 버퍼의 Data를 비우고 접속을 해제 합니다.

-1 을 입력 할 경우, 버퍼를 모두 비울 때까지 Return 하지 않습니다.

[See Also]

[2-1 Result Codes](#)

3-2-7 getDeviceFontList

DeviceFont 관련하여 선언된 정보와 String 값을 NSMutableArray 형태로 얻어옵니다.

[Syntax]

```
-(NSMutableArray*) getDeviceFontList;
```

[Return Value]

실패 시 nil 을 반환 합니다.

[See Also]

[2-12 Device Fonts](#)

3-2-8 getVectorFontList

VectorFont 관련하여 선언된 정보와 String 값을 NSMutableArray 형태로 얻어옵니다.

[Syntax]

```
-(NSMutableArray*) getVectorFontList
```

[Return Value]

실패 시 nil 을 반환 합니다.

[See Also]

[2-16 Vector Fonts](#)

3-2-9 getDrawBlockOptions

Draw Block Options 관련하여 선언된 정보와 String 값을 NSMutableArray 형태로 얻어옵니다.

[Syntax]

```
-(NSMutableArray*) getDrawBlockOptions;
```

[Return Value]

실패 시 nil 을 반환 합니다.

[See Also]

[2-17 Draw Block Options](#)

3-2-10 getDrawCircleSizes

Draw Circle Sizes 관련하여 선언된 정보와 String 값을 NSMutableArray 형태로 얻어옵니다.

[Syntax]

```
-(NSMutableArray*) getDrawCircleSizes;
```

[Return Value]

실패 시 nil 을 반환 합니다.

[See Also]

[2-18 Draw Circle Sizes](#)

3-2-11 getBarcodeType1D

1D Barcode Type 관련하여 선언된 정보와 String 값을 NSMutableArray 형태로 얻어옵니다.

[Syntax]

```
-(NSMutableArray*) getBarcodeType1D;
```

[Return Value]

실패 시 nil 을 반환 합니다.

[See Also]

[2-5 1D Barcode Types](#)

3-2-12 getBarcodeHRI

1D Barcode Type 관련하여 선언된 정보와 String 값을 NSMutableArray 형태로 얻어옵니다.

[Syntax]

```
-(NSMutableArray*) getBarcodeHRI;
```

[Return Value]

실패 시 nil 을 반환 합니다.

[See Also]

[2-3 1D Barcode HRI](#)

3-2-13 doPrint

프린터의 출력 버퍼에 쌓여있는 내용을 인쇄 합니다.

[Syntax]

```
-(long) doPrint:(NSInteger)numberOfCopies;
```

[Parameters]

numberOfCopies

출력을 반복할 횟수를 저장 합니다.

[Return Value]

성공 시 `_SDK_RESULT_SUCCESS` 을 반환 합니다.

그 외 결과값에 대한 자세한 내용은 [2-1 Result Codes](#) 를 참조하십시오.

[See Also]

[2-1 Result Codes](#)

3-2-14 drawTextDeviceFont

이미지 버퍼에 Device Font 를 이용하여 Text String 출력을 요청 합니다.

[Syntax]

```
-(long) drawTextDeviceFont:(NSString*)text
      xPosition:(NSInteger)xPosition
      yPosition:(NSInteger)yPosition
      fontSelection:(char)fontSelection
      fontWidth:(NSInteger)fontWidth
      fontHeight:(NSInteger)fontHeight
      rightSideCharacterSpacing:(NSInteger)rightSideCharacterSpacing
      fontRotation:(NSInteger)fontRotation
      reverse:(BOOL)reverse
      bold:(BOOL)bold
      textAlignment:(NSInteger)textAlignment;
```

[Parameters]

text

출력을 원하는 스트링.

xPosition

출력을 원하는 위치의 X 좌표.

yPosition

출력을 원하는 위치의 Y 좌표.

fontSelection

Font 를 선택 합니다.(참조: [2-15 Device Fonts](#))

fontWidth

Font의 가로 확대 배율을 선택 합니다. (1~4).

fontHeight

Font의 세로 확대 배율을 선택 합니다. (1~4).

rightSideCharacterSpacing

문자의 오른쪽 여백을 지정 합니다. (ex: 5, +3, -10...).

fontRotation

폰트의 회전 방향을 선택 합니다. (참조 : [2-14 Rotation Degrees](#))

reverse

역상체 사용 여부를 선택 합니다.

YES 일 경우 **역상체**로 출력 합니다..

bold

굵은 글씨 사용 여부를 선택 합니다.

YES 일 경우 **굵은 글씨**로 출력 합니다..

textAlignment

정렬 방식을 선택 합니다.(참조 : [2-2 Alignment](#))

[Return Value]

성공 시 `_SDK_RESULT_SUCCESS` 을 반환 합니다.
그 외 결과값에 대한 자세한 내용은 [2-1 Result Codes](#) 를 참조하십시오.

[Discussion]

- `textAlignment` 로 `_SDK_ALIGNMENT_CENTER` 사용 할 경우 `_SDK_ALIGNMENT_STRING_FROM_RIGHT_2_LEFT`와 같이 동작 합니다.
- 해당 API 사용 직 후 프린터에서는 아무것도 출력하지 않습니다..
[3-2-13 doPrint](#) API 호출 이후에 본 API 를 이용하여 요청한 내용이 출력 됩니다.

[See Also]

[2-1 Result Codes](#)
[2-2 Alignment](#)
[2-14 Rotation Degrees](#)
[2-15 Device Fonts](#)
[3-2-13 doPrint](#)

3-2-15 drawTextVectorFont

이미지 버퍼에 Vector Font 를 이용하여 Text String 출력을 요청 합니다.

[Syntax]

```

-(long) drawTextVectorFont:(NSString*)text
        xPosition:(NSInteger)xPosition
        yPosition:(NSInteger)yPosition
        fontSelection:(char)fontSelection
        fontWidth:(NSInteger)fontWidth
        fontHeight:(NSInteger)fontHeight
        rightSideCharacterSpacing:(NSInteger)rightSideCharacterSpacing
        fontRotation:(NSInteger)fontRotation
        reverse:(BOOL)reverse
        bold:(BOOL)bold
        italic:(BOOL)italic
        textWriteDirectionRightToLeft:(BOOL)textWriteDirectionRightToLeft
        textAlignment:(NSInteger)textAlignment;

```

[Parameters]

text

출력을 원하는 스트링.

xPosition

출력을 원하는 위치의 X 좌표.

yPosition

출력을 원하는 위치의 Y 좌표.

fontSelection

Font 를 선택 합니다.(참조: [2-15 Device Fonts](#))

fontWidth

Font의 가로 확대 배율을 선택 합니다. (1~4).

fontHeight

Font의 세로 확대 배율을 선택 합니다. (1~4).

rightSideCharacterSpacing

문자의 오른쪽 여백을 지정 합니다. (ex: 5, +3, -10...).

fontRotation

폰트의 회전 방향을 선택 합니다. (참조 : [2-14 Rotation Degrees](#))

reverse

역상체 사용 여부를 선택 합니다.

YES 일 경우 **역상체**로 출력 합니다..

bold

굵은 글씨 사용 여부를 선택 합니다.

YES 일 경우 **굵은 글씨**로 출력 합니다..

italic

기울임체 사용 여부를 선택 합니다.

YES 일 경우 **기울임체** 로 출력 합니다..

textWriteDirectionRightToLeft

문자열을 오른쪽에서 왼쪽 방향으로 출력 합니다.

textAlignment

정렬 방식을 선택 합니다.(참조 : [2-2 Alignment](#))

[Return Value]

성공 시 `_SDK_RESULT_SUCCESS` 을 반환 합니다.

그 외 결과값에 대한 자세한 내용은 [2-1 Result Codes](#) 를 참조하십시오.

[Discussion]

- `textAlignment` 로 `_SDK_ALIGNMENT_STRING_FROM_RIGHT_2_LEFT` 사용 할 경우 `_SDK_ALIGNMENT_CENTER` 와 같이 동작 합니다.
- 해당 API 사용 직 후 프린터에서는 아무것도 출력하지 않습니다..
[3-2-13 doPrint](#) API 호출 이후에 본 API 를 이용하여 요청한 내용이 출력 됩니다.

[See Also]

[2-1 Result Codes](#)

[2-2 Alignment](#)

[2-14 Rotation Degrees](#)

[2-15 Device Fonts](#)

[3-2-13 doPrint](#)

3-2-16 drawBarcode1D

이미지 버퍼에 1D 바코드 출력을 요청 합니다.

[Syntax]

```
-(long) drawBarcode1D:(NSString*)data
    xPosition:(NSInteger)xPosition
    yPosition:(NSInteger)yPosition
    barcodeType:(NSInteger)barcodeType
    widthNarrow:(NSInteger)widthNarrow
    widthWide:(NSInteger)widthWide
    height:(NSInteger)height
    hri:(NSInteger)hri
    quietZoneWidth:(NSInteger)quietZoneWidth
    rotation:(NSInteger)rotation;
```

[Parameters]

text

출력을 원하는 바코드 값..

xPosition

출력을 원하는 위치의 X 좌표.

yPosition

출력을 원하는 위치의 Y 좌표.

barcodeType

Barcode Type 을 선택 합니다.(참조: [2-5 1D Barcode Types](#))

widthNarrow

narrow bar 의 width.

widthWide

wide bar 의 width.

height

바코드의 높이

hri

바코드의 Data 값의 출력 위치 Human Readable Interpretation
(참조 : [2-3 Barcode HRI](#))

rotation

바코드의 회전 방향을 선택 합니다. (참조 : [2-14 Rotation Degrees](#))

[Return Value]

성공 시 `_SDK_RESULT_SUCCESS` 을 반환 합니다.

그 외 결과값에 대한 자세한 내용은 [2-1 Result Codes](#) 를 참조하십시오.

[Discussion]

해당 API 사용 직 후 프린터에서는 아무것도 출력하지 않습니다..

[3-2-13 doPrint](#) API 호출 이후에 본 API 를 이용하여 요청한 내용이 출력 됩니다.

[See Also]

[2-1 Result Codes](#)

[2-5 1D Barcode Types](#)

[2-14 Rotation Degrees](#)

[2-3 Barcode HRI](#)

[3-2-13 doPrint](#)

3-2-17 drawBarcodeMaxiCode

이미지 버퍼에 MaxiCode 바코드 출력을 요청 합니다.

[Syntax]

```
-(long) drawBarcodeMaxiCode:(NSString*)data
                        xPosition:(NSInteger)xPosition
                        yPosition:(NSInteger)yPosition
                        mode:(NSInteger)mode;
```

[Parameters]

text

출력을 원하는 바코드 값..

xPosition

출력을 원하는 위치의 X 좌표.

yPosition

출력을 원하는 위치의 Y 좌표.

mode

MaxiCode 의 Mode 를 선택 합니다. (참조 : [2-4 MaxiCode Modes](#))

[Return Value]

성공 시 `_SDK_RESULT_SUCCESS` 을 반환 합니다.

그 외 결과값에 대한 자세한 내용은 [2-1 Result Codes](#) 를 참조하십시오.

[Discussion]

해당 API 사용 직 후 프린터에서는 아무것도 출력하지 않습니다..

[3-2-13 doPrint](#) API 호출 이후에 본 API 를 이용하여 요청한 내용이 출력 됩니다.

[See Also]

[2-1 Result Codes](#)

[2-4 MaxiCode Modes](#)

[3-2-13 doPrint](#)

3-2-18 drawBarcodePDF417

이미지 버퍼에 PDF417 바코드 출력을 요청 합니다.

[Syntax]

```

-(long) drawBarcodePDF417:(NSString*)data
        xPosition:(NSInteger)xPosition
        yPosition:(NSInteger)yPosition
        maximumRowCount:(NSInteger)maximumRowCount // 3~90
        maximumColumnCount:(NSInteger)maximumColumnCount // 1~30
        errorCorrectionLevel:(NSInteger)errorCorrectionLevel
        dataCompressionMethod:(NSInteger)dataCompressionMethod
        printBarcodeText:(BOOL)printBarcodeText
        barcodeOriginPoint:(NSInteger)barcodeOriginPoint
        moduleWidth:(NSInteger)moduleWidth // 2~9
        barHeight:(NSInteger)barHeight // 4~99
        rotation:(NSInteger)rotation;

```

[Parameters]

text

출력을 원하는 바코드 값..

xPosition

출력을 원하는 위치의 X 좌표.

yPosition

출력을 원하는 위치의 Y 좌표.

maximumRowCount

maximum Row Count (3~90).

errorCorrectionLevel

에러 보정 레벨 (참조: [2-7 Error Correction Level](#))

dataCompressionMethod

Data Compression Method (참조: [2-8 Data Compression Method](#))

printBarcodeText

HRI 출력 여부 선택.

barcodeOriginPoint

바코드의 위치 지정 시 기준으로 사용 할 위치 선택.

(참조: [2-6 Barcode Origin Point](#))

moduleWidth

module Width (2~9)

BarHeight

Barcode의 높이 (4~99)

rotation

바코드의 회전 방향을 선택 합니다. (참조 : [2-14 Rotation Degrees](#))

[Return Value]

성공 시 `_SDK_RESULT_SUCCESS` 을 반환 합니다.

그 외 결과값에 대한 자세한 내용은 [2-1 Result Codes](#) 를 참조하십시오.

[Discussion]

해당 API 사용 직 후 프린터에서는 아무것도 출력하지 않습니다..

[3-2-13 doPrint](#) API 호출 이후에 본 API 를 이용하여 요청한 내용이 출력 됩니다.

[See Also]

[2-1 Result Codes](#)

[2-6 Barcode Origin Point](#)

[2-7 Error Correction Level](#)

[2-8 Data Compression Method](#)

[2-14 Rotation Degrees](#)

[3-2-13 doPrint](#)

3-2-19 drawBarcodeQRCode

이미지 버퍼에 QRCode 바코드 출력을 요청 합니다.

[Syntax]

```
-(long) drawBarcodeQRCode:(NSString*)data
                    xPosition:(NSInteger)xPosition
                    yPosition:(NSInteger)yPosition
                    barcodeSize:(NSInteger)barcodeSize // 1~4
                    model:(NSInteger)model
                    errorCorrectionLevel:(NSInteger)errorCorrectionLevel
                    rotation:(NSInteger)rotation;
```

[Parameters]

data

출력을 원하는 바코드 값..

xPosition

출력을 원하는 위치의 X 좌표.

yPosition

출력을 원하는 위치의 Y 좌표.

barcodeSize

바코드의 크기를 선택 합니다. (1~4)

model

QRCode 의 Model 를 선택 합니다. (참조 : [2-9 QRCode Model](#))

errorCorrectionLevel

에러 보정 레벨 (참조: [2-7 Error Correction Level](#))

rotation

바코드의 회전 방향을 선택 합니다. (참조 : [2-14 Rotation Degrees](#))

[Return Value]

성공 시 `_SDK_RESULT_SUCCESS` 을 반환 합니다.

그 외 결과값에 대한 자세한 내용은 [2-1 Result Codes](#) 를 참조하십시오.

[Discussion]

해당 API 사용 직 후 프린터에서는 아무것도 출력하지 않습니다..

[3-2-13 doPrint](#) API 호출 이후에 본 API 를 이용하여 요청한 내용이 출력 됩니다.

[See Also]

[2-1 Result Codes](#)

[2-4 MaxiCode Modes](#)

[2-9 QRCode Model](#)

[2-14 Rotation Degrees](#))

[3-2-13 doPrint](#)

3-2-20 drawBarcodeDataMatrix

이미지 버퍼에 Data Matrix 바코드 출력을 요청 합니다.

[Syntax]

```
-(long) drawBarcodeDataMatrix:(NSString*)data
                        xPosition:(NSInteger)xPostion
                        yPosition:(NSInteger)yPostion
                        barcodeSize:(NSInteger)barcodeSize // 1~4
                        reverse:(BOOL)reverse
                        rotation:(NSInteger)rotation;
```

[Parameters]

data

출력을 원하는 바코드 값..

xPosition

출력을 원하는 위치의 X 좌표.

yPosition

출력을 원하는 위치의 Y 좌표.

barcodeSize

바코드의 크기를 선택 합니다. (1~4)

reverse

reverse 모드를 선택합니다.

rotation

바코드의 회전 방향을 선택 합니다. (참조 : [2-14 Rotation Degrees](#))

[Return Value]

성공 시 `_SDK_RESULT_SUCCESS` 을 반환 합니다.

그 외 결과값에 대한 자세한 내용은 [2-1 Result Codes](#) 를 참조하십시오.

[Discussion]

해당 API 사용 직 후 프린터에서는 아무것도 출력하지 않습니다..

[3-2-13 doPrint](#) API 호출 이후에 본 API 를 이용하여 요청한 내용이 출력 됩니다.

[See Also]

[2-1 Result Codes](#)

[2-14 Rotation Degrees](#)

[3-2-13 doPrint](#)

3-2-21 drawBarcodeAztec

이미지 버퍼에 Aztec 바코드 출력을 요청 합니다.

[Syntax]

```

-(long) drawBarcodeAztec:(NSString*)data
        xPosition:(NSInteger)xPostion
        yPosition:(NSInteger)yPostion
        barcodeSize:(NSInteger)barcodeSize // 1~10
        extendedChannel:(BOOL)extendedChannel
        errorCorrectionLevel:(NSInteger)errorCorrectionLevel
        menuSymbol:(BOOL)menuSymbol
        numberOfSymbols:(NSInteger)numberOfSymbols
        optionalID:(NSString*)optionalID
        rotation:(NSInteger)rotation;
    
```

[Parameters]

data

출력을 원하는 바코드 값..

xPosition

출력을 원하는 위치의 X 좌표.

yPosition

출력을 원하는 위치의 Y 좌표.

barcodeSize

바코드의 크기를 선택 합니다. (1~10)

extendedChannel

reverse 모드를 선택합니다.

errorCorrectionLevel

에러 보정 레벨

Value	Error control and symbol size/type
0	Default error correction level
1 ~ 99	Error correction percentage
101 ~ 104	1 ~ 4 layer compact symbol
201 ~ 232	1 ~ 32 layer full range symbol
300	Simple Aztec "Rune"

menuSymbol

Menu symbol 사용 여부를 선택합니다.

numberOfSymbols

Number of symbols for structured append: (1 ~ 26)

optionalID

Optional ID field for structured append: ID field string (Maximum 24 character)

rotation

바코드의 회전 방향을 선택 합니다. (참조 : [2-14 Rotation Degrees](#))

[Return Value]

성공 시 `_SDK_RESULT_SUCCESS` 을 반환 합니다.

그 외 결과값에 대한 자세한 내용은 [2-1 Result Codes](#) 를 참조하십시오.

[Discussion]

해당 API 사용 직 후 프린터에서는 아무것도 출력하지 않습니다..

[3-2-13 doPrint](#) API 호출 이후에 본 API 를 이용하여 요청한 내용이 출력 됩니다.

[See Also]

[2-1 Result Codes](#)

[2-14 Rotation Degrees](#)

[3-2-13 doPrint](#)

3-2-22 drawBarcodeCode49

이미지 버퍼에 Code49 바코드 출력을 요청 합니다.

[Syntax]

```
-(long)drawBarcodeCode49:(NSString*)data
    xPosition:(NSInteger)xPosition
    yPosition:(NSInteger)yPosition
    widthNarrow:(NSInteger)widthNarrow
    widthWide:(NSInteger)widthWide
    height:(NSInteger)height
    hri:(NSInteger)hri
    startingMode:(NSInteger)startingMode
    rotation:(NSInteger)rotation;
```

[Parameters]

data

출력을 원하는 바코드 값..

xPosition

출력을 원하는 위치의 X 좌표.

yPosition

출력을 원하는 위치의 Y 좌표.

widthNarrow

narrow bar 의 width.

widthWide

wide bar 의 width.

hri

바코드의 Data 값의 출력 위치 Human Readable Interpretation
(참조: [2-3 Barcode HRI](#))

height

바코드의 높이

startingMode

Starting Mode 를 선택 합니다. (참조 : [2-10 Code 49 Starting Mode](#))

rotation

바코드의 회전 방향을 선택 합니다. (참조 : [2-14 Rotation Degrees](#))

[Return Value]

성공 시 `_SDK_RESULT_SUCCESS` 을 반환 합니다.

그 외 결과값에 대한 자세한 내용은 [2-1 Result Codes](#) 를 참조하십시오.

[Discussion]

해당 API 사용 직 후 프린터에서는 아무것도 출력하지 않습니다..

[3-2-13 doPrint](#) API 호출 이후에 본 API 를 이용하여 요청한 내용이 출력 됩니다.

[See Also]

[2-1 Result Codes](#)

[2-3 Barcode HRI](#)

[2-10 Code 49 Starting Mode](#)

[2-14 Rotation Degrees](#)

[2-3 Barcode HRI](#)

[3-2-13 doPrint](#)

3-2-23 drawBarcodeCodaBlock

이미지 버퍼에 CodaBlock 바코드 출력을 요청 합니다.

[Syntax]

```
-(long) drawBarcodeCodaBlock:(NSString*)data
    xPosition:(NSInteger)xPosition
    yPosition:(NSInteger)yPosition
    widthNarrow:(NSInteger)widthNarrow
    widthWide:(NSInteger)widthWide
    height:(NSInteger)height
    securityLevel:(BOOL)securityLevel
    numberOfCharactersPerrow:(NSInteger)numberOfCharactersPerrow
    mode:(char)mode
    numberOfRowToEncode:(NSInteger)numberOfRowToEncode;
```

[Parameters]

data

출력을 원하는 바코드 값..

xPosition

출력을 원하는 위치의 X 좌표.

yPosition

출력을 원하는 위치의 Y 좌표.

widthNarrow

narrow bar 의 width.

widthWide

wide bar 의 width.

height

바코드의 높이

securityLevel

securityLevel 사용 여부를 선택 합니다.

numberOfCharactersPerrow

Number of characters per row (data columns): 2~62

mode

바코드 출력 모드를 선택 합니다.(참조: [2-11 Codablock Mode](#))

numberOfRowToEncode

Number of rows to encode

Mode	Value
_SDK_CODABLACK_MODE_A	1 ~ 18
_SDK_CODABLACK_MODE_E	2 ~ 4
_SDK_CODABLACK_MODE_F	2 ~ 4

[Return Value]

성공 시 `_SDK_RESULT_SUCCESS` 을 반환 합니다.

그 외 결과값에 대한 자세한 내용은 [2-1 Result Codes](#) 를 참조하십시오.

[Discussion]

해당 API 사용 직 후 프린터에서는 아무것도 출력하지 않습니다..

[3-2-13 doPrint](#) API 호출 이후에 본 API 를 이용하여 요청한 내용이 출력 됩니다.

[See Also]

[2-1 Result Codes](#)

[2-11 Codablock Mode](#)

[3-2-13 doPrint](#)

3-2-24 drawBarcodeMicroPDF

이미지 버퍼에 Micro PDF417 바코드 출력을 요청 합니다.

[Syntax]

```

-(long) drawBarcodeMicroPDF:(NSString*)data
        xPosition:(NSInteger)xPostion
        yPosition:(NSInteger)yPostion
        moduleWidth:(NSInteger)moduleWidth // 2~8
        barcodeHeight:(NSInteger)barcodeHeight // 1~99
        mode:(NSInteger)mode // 0~33
        rotation:(NSInteger)rotation;

```

[Parameters]

text

출력을 원하는 바코드 값..

xPosition

출력을 원하는 위치의 X 좌표.

yPosition

출력을 원하는 위치의 Y 좌표.

moduleWidth

module Width (2~8)

barcodeHeight

Barcode의 높이 (1~99)

rotation

바코드의 회전 방향을 선택 합니다. (참조 : [2-14 Rotation Degrees](#))

[Return Value]

성공 시 `_SDK_RESULT_SUCCESS` 을 반환 합니다.

그 외 결과값에 대한 자세한 내용은 [2-1 Result Codes](#) 를 참조하십시오.

[Discussion]

해당 API 사용 직 후 프린터에서는 아무것도 출력하지 않습니다..

[3-2-13 doPrint](#) API 호출 이후에 본 API 를 이용하여 요청한 내용이 출력 됩니다.

[See Also]

[2-1 Result Codes](#)

[2-14 Rotation Degrees](#)

[3-2-13 doPrint](#)

3-2-25 drawBarcodeIMB

이미지 버퍼에 drawBarcodeIMB 바코드 출력을 요청 합니다.

[Syntax]

```
-(long) drawBarcodeIMB:(NSString*)data
                xPosition:(NSInteger)xPosition
                yPosition:(NSInteger)yPosition
                printBarcodeText:(BOOL)printBarcodeText
                rotation:(NSInteger)rotation;
```

[Parameters]

data

출력을 원하는 바코드 값..

xPosition

출력을 원하는 위치의 X 좌표.

yPosition

출력을 원하는 위치의 Y 좌표.

printBarcodeText

HRI 출력 여부 선택.

rotation

바코드의 회전 방향을 선택 합니다. (참조 : [2-14 Rotation Degrees](#))

[Return Value]

성공 시 `_SDK_RESULT_SUCCESS` 을 반환 합니다.

그 외 결과값에 대한 자세한 내용은 [2-1 Result Codes](#) 를 참조하십시오.

[Discussion]

해당 API 사용 직 후 프린터에서는 아무것도 출력하지 않습니다..

[3-2-13 doPrint](#) API 호출 이후에 본 API 를 이용하여 요청한 내용이 출력 됩니다.

[See Also]

[2-1 Result Codes](#)

[2-14 Rotation Degrees](#)

[3-2-13 doPrint](#)

3-2-26 drawBarcodeMSI

이미지 버퍼에 drawBarcodeMSI 바코드 출력을 요청 합니다.

[Syntax]

```
-(long) drawBarcodeMSI:(NSString*)data
      xPosition:(NSInteger)xPosition
      yPosition:(NSInteger)yPosition
      widthNarrow:(NSInteger)widthNarrow
      widthWide:(NSInteger)widthWide
      height:(NSInteger)height
      checkDigitSelection:(NSInteger)checkDigitSelection
      printCheckDigitInHRI:(BOOL)printCheckDigitInHRI
      hri:(NSInteger)hri
      rotation:(NSInteger)rotation;
```

[Parameters]

data

출력을 원하는 바코드 값..

xPosition

출력을 원하는 위치의 X 좌표.

yPosition

출력을 원하는 위치의 Y 좌표.

widthNarrow

narrow bar 의 width.

widthWide

wide bar 의 width.

height

바코드의 높이.

checkDigitSelection

checkDigit Option 선택 (참조 : [2-12 Check Digit Option](#))

printCheckDigitInHRI

HRI 에 Check Digit를 포함 할 지 여부 선택

hri

바코드의 Data 값의 출력 위치 Human Readable Interpretation
(참조: [2-3 Barcode HRI](#))

rotation

바코드의 회전 방향을 선택 합니다. (참조 : [2-14 Rotation Degrees](#))

[Return Value]

성공 시 `_SDK_RESULT_SUCCESS` 을 반환 합니다.

그 외 결과값에 대한 자세한 내용은 [2-1 Result Codes](#) 를 참조하십시오.

[Discussion]

해당 API 사용 직 후 프린터에서는 아무것도 출력하지 않습니다..

[3-2-13 doPrint](#) API 호출 이후에 본 API 를 이용하여 요청한 내용이 출력 됩니다.

[See Also]

[2-1 Result Codes](#)

[2-3 Barcode HRI](#)

[2-12 Check Digit Option](#)

[2-14 Rotation Degrees](#)

[3-2-13 doPrint](#)

3-2-27 drawBarcodePlessey

이미지 버퍼에 drawBarcodePlessey 바코드 출력을 요청 합니다.

[Syntax]

```
-(long) drawBarcodePlessey:(NSString*)data
    xPosition:(NSInteger)xPosition
    yPosition:(NSInteger)yPosition
    widthNarrow:(NSInteger)widthNarrow
    widthWide:(NSInteger)widthWide
    height:(NSInteger)height
    printCheckDigit:(BOOL)printCheckDigit
    hri:(NSInteger)hri
    rotation:(NSInteger)rotation;
```

[Parameters]

data

출력을 원하는 바코드 값..

xPosition

출력을 원하는 위치의 X 좌표.

yPosition

출력을 원하는 위치의 Y 좌표.

widthNarrow

narrow bar 의 width.

widthWide

wide bar 의 width.

height

바코드의 높이.

printCheckDigitInHRI

HRI 에 Check Digit를 포함 할 지 여부 선택

hri

바코드의 Data 값의 출력 위치 Human Readable Interpretation
(참조: [2-3 Barcode HRI](#))

rotation

바코드의 회전 방향을 선택 합니다. (참조 : [2-14 Rotation Degrees](#))

[Return Value]

성공 시 `_SDK_RESULT_SUCCESS` 을 반환 합니다.

그 외 결과값에 대한 자세한 내용은 [2-1 Result Codes](#) 를 참조하십시오.

[Discussion]

해당 API 사용 직 후 프린터에서는 아무것도 출력하지 않습니다..

[3-2-13 doPrint](#) API 호출 이후에 본 API 를 이용하여 요청한 내용이 출력 됩니다.

[See Also]

[2-1 Result Codes](#)

[2-3 Barcode HRI](#)

[2-14 Rotation Degrees](#)

[3-2-13 doPrint](#)

3-2-28 drawBarcodeTLC39

이미지 버퍼에 drawBarcodeTLC39 바코드 출력을 요청 합니다.

[Syntax]

```
-(long) drawBarcodeTLC39:(NSString*)data
        xPosition:(NSInteger)xPosition
        yPosition:(NSInteger)yPosition
        widthNarrow:(NSInteger)widthNarrow
        widthWide:(NSInteger)widthWide
        height:(NSInteger)height
        rowHeightOfMicroPDF417:(NSInteger)rowHeightOfMicroPDF417
        narrowWidthOfMicroPDF417:(NSInteger)narrowWidthOfMicroPDF417
        rotation:(NSInteger)rotation;
```

[Parameters]

data

출력을 원하는 바코드 값..

xPosition

출력을 원하는 위치의 X 좌표.

yPosition

출력을 원하는 위치의 Y 좌표.

widthNarrow

narrow bar 의 width.

widthWide

wide bar 의 width.

height

바코드의 높이.

rowHeightOfMicroPDF417

microPDF417 바코드의 row Height

narrowWidthOfMicroPDF417

microPDF417 바코드의 narrowWideHeight

rotation

바코드의 회전 방향을 선택 합니다. (참조 : [2-14 Rotation Degrees](#))

[Return Value]

성공 시 `_SDK_RESULT_SUCCESS` 을 반환 합니다.

그 외 결과값에 대한 자세한 내용은 [2-1 Result Codes](#) 를 참조하십시오.

[Discussion]

해당 API 사용 직 후 프린터에서는 아무것도 출력하지 않습니다..

[3-2-13 doPrint](#) API 호출 이후에 본 API 를 이용하여 요청한 내용이 출력 됩니다.

[See Also]

[2-1 Result Codes](#)

[2-14 Rotation Degrees](#)

[3-2-13 doPrint](#)

3-2-29 drawBarcodeRSS

이미지 버퍼에 drawBarcodeRSS 바코드 출력을 요청 합니다.

[Syntax]

```
-(long) drawBarcodeRSS:(NSString*)data
        xPosition:(NSInteger)xPosition
        yPosition:(NSInteger)yPosition
        barcodeType:(NSInteger)barcodeType
        magnification:(NSInteger)magnification // 1~10
        separatorHeight:(NSInteger)separatorHeight // 1~2
        barcodeHeight:(NSInteger)barcodeHeight
        segmentWidth:(NSInteger)segmentWidth // 0~22
        rotation:(NSInteger)rotation;
```

[Parameters]

data

출력을 원하는 바코드 값..

xPosition

출력을 원하는 위치의 X 좌표.

yPosition

출력을 원하는 위치의 Y 좌표.

barcodeType

RSS 바코드 종류 선택(참조: [2-13 RSS Barcode Type](#)).

magnification

magnification. (1~10)

separatorHeight

separator 높이(1~2)

barcodeHeight

바코드의 높이.

segmentWidth

segmentWidth(0~22)

rotation

바코드의 회전 방향을 선택 합니다. (참조 : [2-14 Rotation Degrees](#))

[Return Value]

성공 시 `_SDK_RESULT_SUCCESS` 을 반환 합니다.

그 외 결과값에 대한 자세한 내용은 [2-1 Result Codes](#) 를 참조하십시오.

[Discussion]

해당 API 사용 직 후 프린터에서는 아무것도 출력하지 않습니다..

[3-2-13 doPrint](#) API 호출 이후에 본 API 를 이용하여 요청한 내용이 출력 됩니다.

[See Also]

[2-1 Result Codes](#)

[2-13 RSS Barcode Type](#)

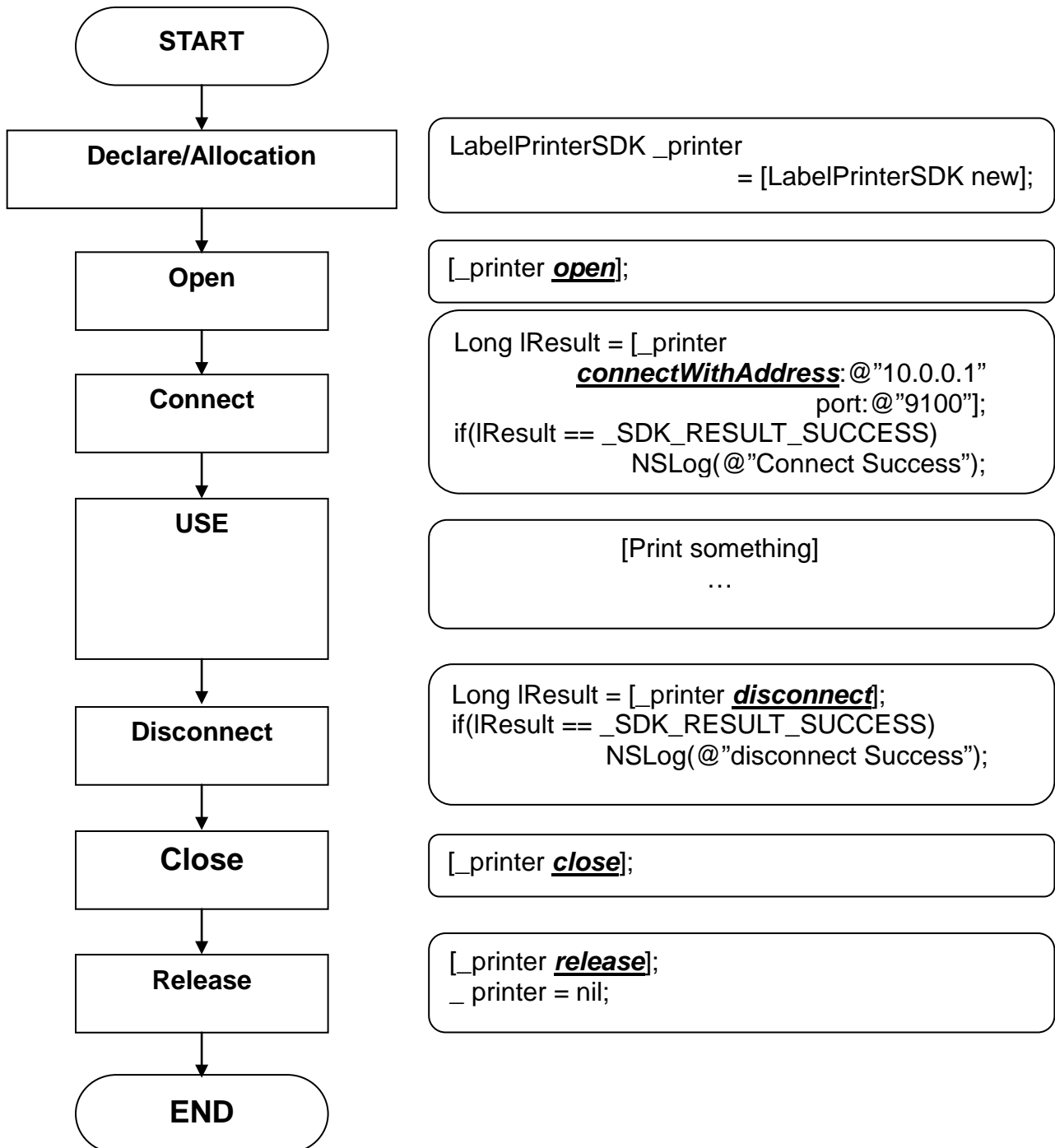
[2-14 Rotation Degrees](#)

[3-2-13 doPrint](#)

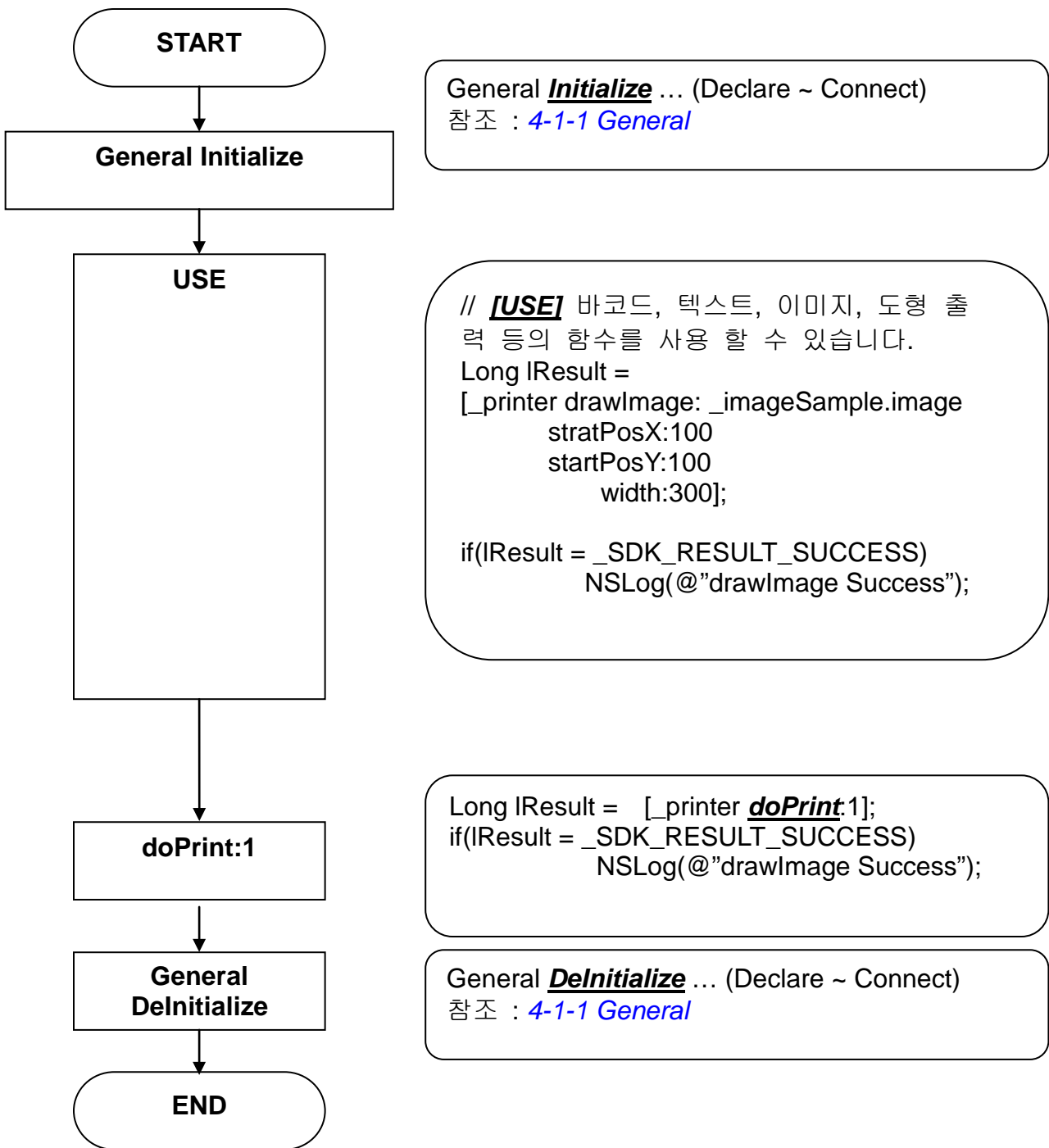
4. Appendix

4-1 권장 사용 순서도

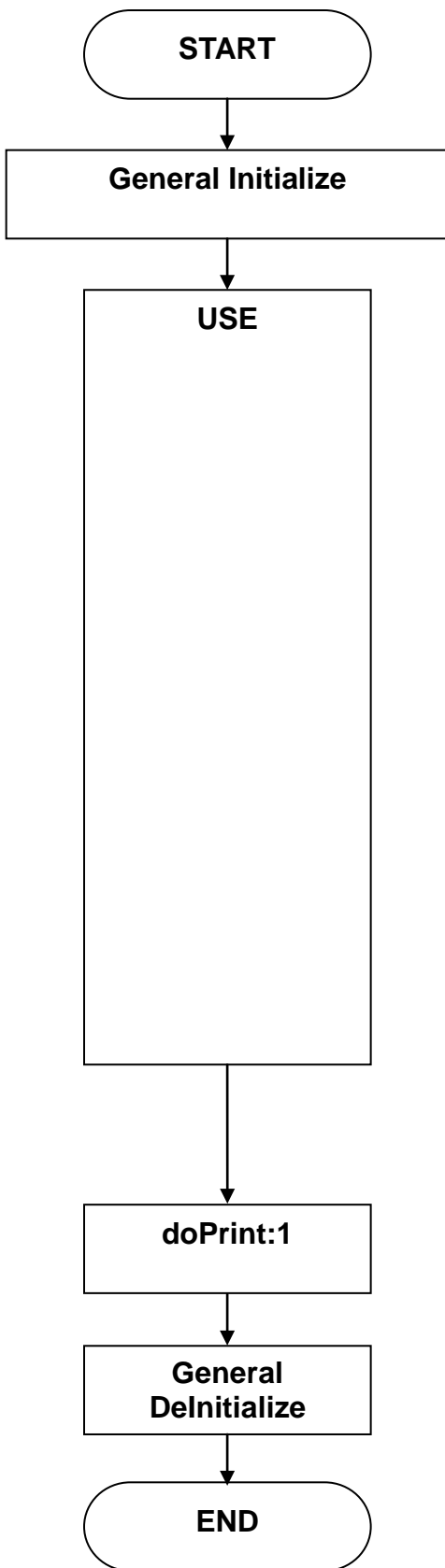
4-1-1 General



4-1-2 콘텐츠 출력 하기



4-1-3 2개 이상의 콘텐츠를 한 면에 출력 할 때.



General ***Initialize*** ... (Declare ~ Connect)
 참조 : [4-1-1 General](#)

```

// [Contents 1]
Long IResult =
[_printer drawImage: _imageSample.image
  stratPosX:100
  startPosY:100
  width:300];
if(IResult = _SDK_RESULT_SUCCESS)
  NSLog(@"drawImage Success");
// [Contents 2]
IResult =
[_printer drawTextDeviceFont:@"0* 12pt,
70, 220" xPosition:70 yPosition:220
fontSelection:_SDK_DEVICE_FONT_12PT
fontWidth:1 fontHeight:1
rightSideCharacterSpacing:0
fontRotation:_SDK_ROTATION_DEGREES
_0 reverse:NO bold:NO
textAlignment:_SDK_ALIGNMENT_LEFT];
if(IResult = _SDK_RESULT_SUCCESS)
  NSLog(@"drawText Success");
  
```

```

Long IResult = [_printer doPrint:1];
if(IResult = _SDK_RESULT_SUCCESS)
  NSLog(@"drawImage Success");
  
```

General ***Deinitialize*** ... (Declare ~ Connect)
 참조 : [4-1-1 General](#)

4-2 Sample Codes

4-2-1 접속 / 접속 해제 하기

```
// 선언/할당 및 SDK 사용 시작
LabelPrinterSDK _printer = [LabelPrinterSDK new];
[_printer open];

// IP Address "10.0.0.1" Port 9100 으로 할당 된 Printer 에 접속 시도.
long IResult = [_printer connectWithAddress:@"10.0.0.1" port:@"9100"];
if(IResult == _SDK_RESULT_SUCCESS)
    NSLog(@"Connect Success");

// SDK 사용이 끝난 뒤
IResult = [_printer disconnect]; // 접속 해제
[_printer release];
_printer = nil;
```

4-2-2 Printer의 Buffer에 그린 내용을 출력 하기

```
long IResult = [_printer doPrint:1]; // 1 이외 다른 수 입력 시, 같은 내용을 반복해서
출력.
if(IResult == _SDK_RESULT_SUCCESS)
```

4-2-3 Printer의 Buffer에 사각형 상자 출력 하기.

```
IResult = [_printer drawBlock:100
startPosY:100
endPosX:300 endPosY:300
option:_SDK_DRAW_BLOCK_OPTION_SLOPE
thickness:1];

NSLog(@" drawBlock Success");
```


4-2-4 Printer의 Buffer에 원형 상자 출력 하기.

```

IResult = [_printer drawCircle:200
startPosY:200
sizeSelection:_SDK_DRAW_CIRCLE_SIZE_40X40
multiplier:1];

if(IResult == _SDK_RESULT_SUCCESS)
NSLog(@"drawTextDeviceFont Success");

```

4-2-5 Printer의 Buffer에 Device Font 출력 하기.

```

IResult = [_printer drawTextDeviceFont:@"device Font Test"
xPosition:70 yPosition:220
fontSelection:_SDK_DEVICE_FONT_12PT
fontWidth:1 fontHeight:1
rightSideCharacterSpacing:0
fontRotation:_SDK_ROTATION_DEGREES_0
reverse:NO bold:NO textAlignment:_SDK_ALIGNMENT_LEFT];
if(IResult == _SDK_RESULT_SUCCESS)
NSLog(@"drawTextDeviceFont Success");

```

4-2-6 Printer의 Buffer에 Vector Font 출력 하기.

```

IResult = [_printer drawTextVectorFont:@"draw Font Test - Vector"
xPosition:50 yPosition:100
fontSelection:_SDK_VECTOR_FONT_ASCII
fontWidth:10 fontHeight:10
rightSideCharacterSpacing:0
fontRotation:_SDK_ROTATION_DEGREES_0
reverse:NO bold:YES italic:YES
textWriteDirectionRightToLeft:NO textAlignment:_SDK_ALIGNMENT_LEFT];

if(IResult == _SDK_RESULT_SUCCESS)
NSLog(@"drawTextVectorFont Success");

```

4-2-7 Printer의 Buffer에 1D Barcode 출력 하기.

```

IResult = [_printer drawBarcode1D:@">A1234567890"
xPosition:50 yPosition:100
barcodeType:_SDK_BARCODE_TYPE_CODE128
widthNarrow:2 widthWide:6 height:100
hri:_SDK_BARCODE_HRI_BELOW_FONTSIZE1
quietZoneWidth:0
rotation:_SDK_ROTATION_DEGREES_0];

if(IResult == _SDK_RESULT_SUCCESS)
    NSLog(@"drawBarcode1D Success");

```

4-2-8 Printer의 Buffer에 PDF417 Barcode 출력 하기.

```

IResult =[_printer drawBarcodePDF417:@"PDF417 Test"
xPosition:50 yPosition:300
maximumRowCount:30 maximumColumnCount:5
errorCorrectionLevel:_SDK_ERROR_CORRECTION_LEVEL0
dataCompressionMethod:_SDK_DATA_COMPRESSION_METHOD_TEXT
printBarcodeText:YES
barcodeOriginPoint:_SDK_BARCODE_ORIGIN_POINT_UpperLeft
moduleWidth:3 barHeight:10
rotation:_SDK_ROTATION_DEGREES_0];

if(IResult == _SDK_RESULT_SUCCESS)
    NSLog(@"drawBarcodePDF417 Success");

```

4-2-9 Printer의 Buffer에 QRCode Barcode 출력 하기.

```

IResult =[_printer drawBarcodeQRCode:@"QR Code Test"
xPosition:450 yPosition:300
barcodeSize:3
model:_SDK_QRCODE_MODEL_2
errorColectionLevel:_SDK_ERROR_CORRECTION_LEVEL1
rotation:_SDK_ROTATION_DEGREES_0];

if(IResult == _SDK_RESULT_SUCCESS)
    NSLog(@"drawBarcodeQRCode Success");

```

4-2-10 Printer의 Buffer에 DataMatrix Barcode 출력 하기.

```
IResult =[_printer drawBarcodeDataMatrix:@"dataMatrix Test"  
xPosition:600 yPosition:300  
barcodeSize:3 reverse:NO  
rotation:_SDK_ROTATION_DEGREES_0];  
  
if(IResult == _SDK_RESULT_SUCCESS)  
NSLog(@" drawBarcodeDataMatrix Success");
```

4-2-11 Printer의 Buffer에 MaxiCode Barcode 출력 하기.

```
IResult = [_printer drawBarcodeMaxiCode:@"Maxi Code Test!!!"  
xPosition:50 yPosition:400  
mode:_SDK_MAXICODE_MODE_0];  
  
if(IResult == _SDK_RESULT_SUCCESS)  
NSLog(@"drawBarcodeMaxiCode Success");
```

4-2-12 Printer의 Buffer에 Aztec Barcode 출력 하기.

```
IResult = [_printer drawBarcodeAztec:@"Aztec Barcode Test"  
xPosition:450 yPosition:400  
barcodeSize:3  
extendedChannel:NO  
errorCorrectionLevel:0  
menuSymbol:NO numberOfSymbols:1  
optionalID:@""  
rotation:_SDK_ROTATION_DEGREES_0];  
  
if(IResult == _SDK_RESULT_SUCCESS)  
NSLog(@"drawBarcodeAztec Success");
```

4-2-13 Printer의 Buffer에 Code49 Barcode 출력 하기.

```

IResult = [_printer drawBarcodeCode49:@"Code 49 Test"
xPosition:600 yPosition:400
widthNarrow:1 widthWide:3
height:100
hri:_SDK_BARCODE_HRI_ABOVE_FONTSIZE1
startingMode:_SDK_STARTINGMODE_AUTOMATIC
rotation:_SDK_ROTATION_DEGREES_0];

if(IResult == _SDK_RESULT_SUCCESS)
NSLog(@" drawBarcodeCode49 Success");

```

4-2-14 Printer의 Buffer에 Codablock Barcode 출력 하기.

```

IResult = [_printer drawBarcodeIMB:@"01234567890123456789"
xPosition:50 yPosition:1000
printBarcodeText:YES
rotation:_SDK_ROTATION_DEGREES_0];

if(IResult == _SDK_RESULT_SUCCESS)
NSLog(@" drawBarcodeCodaBlock Success");

```

4-2-15 Printer의 Buffer에 MicroPDF417 Barcode 출력 하기.

```

IResult = [_printer drawBarcodeMicroPDF:@"MICRO PDF 417 TEST"
xPosition:50 yPosition:900
moduleWidth:4
barcodeHeight:4
mode:8
rotation:_SDK_ROTATION_DEGREES_0];

if(IResult == _SDK_RESULT_SUCCESS)
NSLog(@" drawBarcodeMicroPDF Success");

```

4-2-16 Printer의 Buffer에 IMB Barcode 출력 하기.

```

IResult = [_printer drawBarcodeIMB:@"01234567890123456789"
xPosition:50 yPosition:1000
printBarcodeText:YES
rotation:_SDK_ROTATION_DEGREES_0];

if(IResult == _SDK_RESULT_SUCCESS)
NSLog(@" drawBarcodeIMB Success");

```

4-2-17 Printer의 Buffer에 MSI Barcode 출력 하기.

```

IResult = [_printer drawBarcodeMSI:@"123456"
xPosition:50 yPosition:1100
widthNarrow:2 widthWide:7
height:100
checkDigitSelection:_SDK_CHECKDIGIT_1MOD10
printCheckDigitInHRI:YES
hri:_SDK_BARCODE_HRI_BELOW_FONTSIZE1
rotation:_SDK_ROTATION_DEGREES_0];

if(IResult == _SDK_RESULT_SUCCESS)
NSLog(@" drawBarcodeMSI Success");

```

4-2-18 Printer의 Buffer에 Plessey Barcode 출력 하기.

```

IResult = [_printer drawBarcodePlessey:@"12345"
xPosition:100 yPosition:100
widthNarrow:2 widthWide:7
height:100 printCheckDigit:YES
hri:_SDK_BARCODE_HRI_BELOW_FONTSIZE1
rotation:_SDK_ROTATION_DEGREES_0];

if(IResult == _SDK_RESULT_SUCCESS)
NSLog(@" drawBarcodePlessey Success");

```

4-2-19 Printer의 Buffer에 TLC39 Barcode 출력 하기.

```
IResult = [_printer drawBarcodeTLC39:@"123456,ABCD12345678901234"  
xPosition:100 yPosition:300  
widthNarrow:2 widthWide:7  
height:50 rowHeightOfMicroPDF417:3  
narrowWidthOfMicroPDF417:2  
rotation:_SDK_ROTATION_DEGREES_0];  
  
if(IResult == _SDK_RESULT_SUCCESS)  
NSLog(@" drawBarcodeTLC39 Success");
```

4-2-20 Printer의 Buffer에 RSS Barcode 출력 하기.

```
IResult = [_printer drawBarcodeRSS:@"12345678901|this is composite info"  
xPosition:100 yPosition:600  
barcodeType:_SDK_RSS_BARCODE_TYPE_RSS14  
magnification:2  
separatorHeight:1 barcodeHeight:20 segmentWidth:10  
rotation:_SDK_ROTATION_DEGREES_0];  
  
if(IResult == _SDK_RESULT_SUCCESS)  
NSLog(@" drawBarcodeRSS Success");
```

4-3 Sample Application

4-3-1 접속 / 접속 해제

1. Interface 를 선택 합니다.

2. 대상 Printer 의 Mac Address를 입력합니다.

3. 대상 프린터에 접속 합니다.

4. 테스트용 인쇄를 합니다.

5. 접속을 종료 합니다.

※ 다른 기능(text 출력, Image 출력 등)을 이용 할 경우 접속을 종료하지 마십시오.

4-3-2 Device Font 출력

1. 출력할 문자열을 입력합니다.

2. 문자가 출력 될 위치의 좌표를 입력합니다.

3. 폰트 크기를 입력합니다.(1~4)

4. 문자를 프린터 버퍼에 저장합니다.

5. 프린트를 실행합니다.

4-3-3 Vector Font 출력

1. 출력할 문자열을 입력합니다.

2. 문자가 출력 될 위치의 좌표를 입력합니다.

3. 폰트 크기를 입력합니다.(1~4)

4. 문자를 프린터 버퍼에 저장합니다.

5. 프린트를 실행합니다.

4-3-4 Image 출력

1. 이미지가 출력 될 위치의 좌표를 입력합니다.

2. 출력을 원하는 이미지의 너비를 입력합니다.

3. 이미지를 프린터 버퍼에 저장합니다.

4. 프린트를 실행합니다.

4-3-5 다른 기능 이용하기



SIM 없음 오후 7:14 96% 🔋

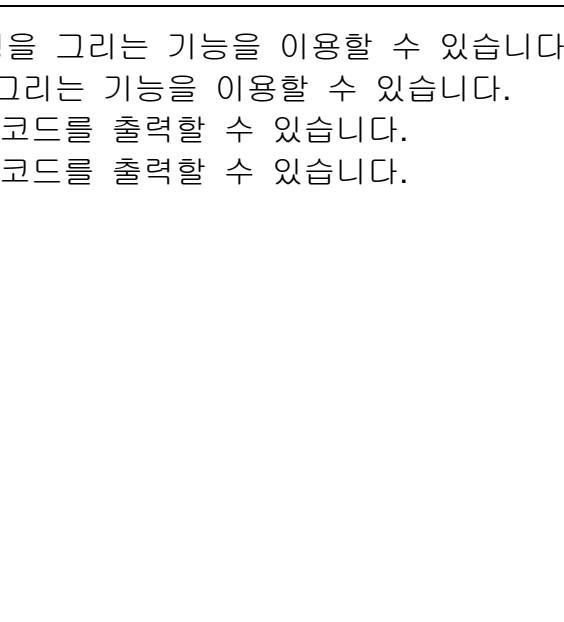
More Edit

- 1 Block
- 2 Circle
- 3 Barcode
- 4 2D Barcode

Connection Device Text Vector Text Image More

1. 사각형을 그리는 기능을 이용할 수 있습니다.
2. 원을 그리는 기능을 이용할 수 있습니다.
3. 1D 바코드를 출력할 수 있습니다.
4. 2D 바코드를 출력할 수 있습니다.

4-3-6 사각형 출력하기



SIM 없음 오후 7:15 95% 🔋

< More

3 DrawBlock 4 Print

BLOCK_BOX Select Option

1 Start Position 10 10

2 End Position 100 100

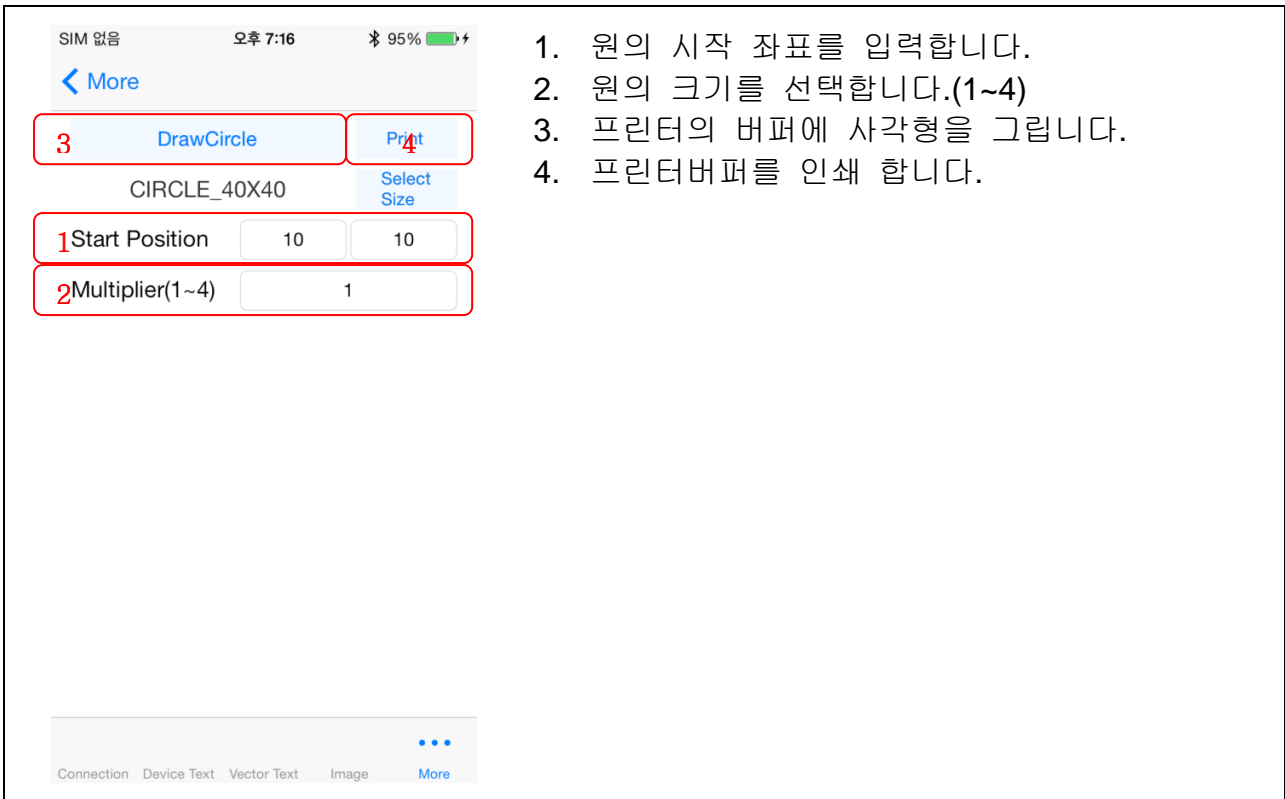
Thickness 2

available modes : 'slopeBlock', 'drawBoxMode'

Connection Device Text Vector Text Image More

1. 사각형의 시작 좌표(좌측 상단)를 입력합니다.
2. 사각형의 끝 좌표(우측 하단)를 입력합니다.
3. 프린터의 버퍼에 사각형을 그립니다.
4. 프린터버퍼를 인쇄 합니다.

4-3-7 원 출력하기



1. 원의 시작 좌표를 입력합니다.

2. 원의 크기를 선택합니다.(1~4)

3. 프린터의 버퍼에 사각형을 그립니다.

4. 프린터버퍼를 인쇄 합니다.