

Class

XPOSCommand

```
class XPOSCommand: NSObject
```

getCommand()

Get the Printing Command

```
func getCommand() -> Data!
```

Return Value

- NSData object representing the current ESC object's printing data.
-

addData(_:)

Add Custom Data

```
func addData(_ customData: Data) -> XPOSCommand!
```

Parameters

- `customData`
The custom data content to be added.

Return Value

- XPOSCommand object.
-

setCharEncoding(_:)

Set Character Encoding

```
func setCharEncoding(_ encoding: String.Encoding) -> XPOSCommand!
```

Parameters

- `encoding`
The encoding format, default is GB_18030_2000.

Return Value

- XPOSCommand object.
-

initializePrinter()

Initialize the Printer

```
func initializePrinter() -> XPOSCommand!
```

Return Value

- XPOSCommand object.
-

lineFeed()

Print and Line Feed

```
func lineFeed() -> XPOSCommand!
```

Return Value

- XPOSCommand object.
-

printAndFeedLines(_:)

Print and Feed n Lines

```
func printAndFeedLines(_ lines: Int) -> XPOSCommand!
```

Parameters

- `lines`
The number of lines to feed after printing.

Return Value

- XPOSCommand object.

setHorizontalTab()

Horizontal Tab

```
func setHorizontalTab() -> XPOSCommand!
```

Return Value

- XPOSCommand object.

select(_:)

Select Alignment

```
func select(_ alignment: POSAlignment) -> XPOSCommand!
```

Parameters

- alignment
The alignment mode to be set, which can be one of the following:

Variable	Description
POSAlignmentLeft	Left alignment
POSAlignmentCenter	Center alignment
POSAlignmentRight	Right alignment

Return Value

- XPOSCommand object.

addText(_:)

Add Text

```
func addText(_ text: String) -> XPOSCommand!
```

Parameters

- `text`

The text content to be added. The default character encoding format is GB_18030_2000.

Return Value

- XPOSCommand object.
-

selectTextSize(_:)

Set Text Size

```
func selectTextSize(_ textSize: Int) -> XPOSCommand!
```

Parameters

- `textSize`

The text font size, default is TextWidth1 | TextHeight1.

Return Value

- XPOSCommand object.
-

addText(_:alignment:attribute:textSize:)

Compound Add Text

```
func addText(_ text: String, alignment: POSAlignment, attribute: POSFontType, textSize: Int) -> XPOSCommand!
```

Parameters

- `text`

The text content to be added.

- `align`

The text alignment, default is POSAlignmentLeft.

- `attribute`

The text attribute, default is FntDefault. Options include:

Variable	Description
FntDefault	Standard font
FntFontB	FontB
FntBold	Bold
FntReverse	Reverse printing
FntUnderline	Underline
FntUnderline2	Double underline

- `textSize`
The text font size, default is TextWidth1 | TextHeight1.

Return Value

- XPOSCommand object.

addTextSize(_:textSize:)

Add Text with Size

```
func addTextSize(_ text: String, textSize: Int) -> XPOSCommand!
```

Parameters

- `text`
The text content to be added.
- `textSize`
The text size, default is TextWidth1 | TextHeight1.

Return Value

- XPOSCommand object.

addTextAttribute(_:attribute:)

Add Text with Attribute

```
func addTextAttribute(_ text: String, attribute: POSFontType) -> XPOSCommand!
```

Parameters

- `text`
The text content to be added.
- `attribute`
The text attribute, default is FntDefault.

Return Value

- XPOSCommand object.

addTextAlignment(_:alignment:)

Add Text with Alignment

```
func addTextAlignment(_ text: String, alignment: POSAlignment) -> XPOSCommand!
```

Parameters

- `text`
The text content to be added.
- `align`
The text alignment, default is POSAlignmentLeft.

Return Value

- XPOSCommand object.

selectFont(_:)

Select Font

```
func selectFont(_ font: FontSelection) -> XPOSCommand!
```

Parameters

- `font`
The font selection. Options include:

Variable	Description
FontStandardASCII	n = 0, 48 Select standard ASCII font (12 x 24)
FontCompressedASCII	n = 1, 49 Select compressed ASCII font (9 x 17)

Return Value

- XPOSCommand object.

selectCharacter(_:)

Select International Character Set

```
func selectCharacter(_ charset: POSCharacterSet) -> XPOSCommand!
```

Parameters

- `charset`

The character set number (0~15). Options include:

Variable	Description
CharacterSetUSA	USA
CharacterSetFrance	France
CharacterSetGermany	Germany
CharacterSetUK	UK
CharacterSetDenmarkI	Denmark I
CharacterSetSweden	Sweden
CharacterSetItaly	Italy
CharacterSetSpainI	Spain I
CharacterSetJapan	Japan
CharacterSetNorway	Norway
CharacterSetDenmarkII	Denmark II
CharacterSetSpainII	Spain II
CharacterSetLatinAmerica	Latin America
CharacterSetKorea	Korea
CharacterSetSloveniaCroatia	Slovenia/Croatia
CharacterSetChina	China

Return Value

- XPOSCommand object.
-

selectOrCancelUnderlineMode(_:)

Select/Cancel Underline Mode

```
func selectOrCancelUnderlineMode(_ mode: Int) -> XPOSCommand!
```

Parameters

- `mode`
The underline mode to select or cancel.

Return Value

- XPOSCommand object.
-

setDefaultLineSpacing()

Set Default Line Spacing

```
- (XPOSCommand *)setDefaultLineSpacing;
```

Return Value

- XPOSCommand object.
-

setLineSpacing(_:)

Set Line Spacing

```
func setLineSpacing(_ spacing: Int) -> XPOSCommand
```

Parameters

- `spacing`
The line spacing to set.

Return Value

- XPOSCCommand object.
-

selectOrCancelBoldMode(_:)

Select/Cancel Bold Mode

```
func selectOrCancelBoldMode(_ mode: Int) -> XPOSCCommand!
```

Parameters

- `mode`
The bold mode to select or cancel.

Return Value

- XPOSCCommand object.
-

setCharacterRightSpacing(_:)

Set Character Right Spacing

```
func setCharacterRightSpacing(_ spacing: Int) -> XPOSCCommand!
```

Parameters

- `spacing`
The right spacing of the character.

Return Value

- XPOSCCommand object.
-

printImage(_:type:mode:)

Print Image

```
func printImage(_ image: UIImage, type: BMPType, mode: BmpScaleType) -> XPOSCCommand!
```

Parameters

- `image`
The image source to print.
- `type`
The image effect: Binarization, Dithering algorithm.
- `mode`
The image scaling mode: Normal/Double Width/Double Height/Double Width and Height.

Return Value

- XPOSCommand object.

compressionPrintImage(_:type:mode:)

Print Image via Compression Algorithm

```
func compressionPrintImage(_ image: NSImage, type: BMPType, mode: BmpScaleType) -> XPOSCommand!
```

Parameters

- `image`
The image source to print.
- `type`
The image effect: Binarization, Dithering algorithm.
- `mode`
The image scaling mode: Normal/Double Width/Double Height/Double Width and Height.

Return Value

- XPOSCommand object.

selectBitmapMode(_:width:image:)

Select Bitmap Mode

```
func selectBitmapMode(_ mode: Int, width: Int, image: NSImage) -> XPOSCommand!
```

Parameters

- `mode`
The bitmap mode to select.
- `width`
The width of the bitmap.
- `image`
The bitmap image.

Return Value

- XPOSCommand object.
-

definedFlashBmpWithImage(_:)

Define Flash Bitmap

```
func definedFlashBmpWithImage(_ images: [UIImage]) -> XPOSCommand!
```

Parameters

- `images`
An array containing bitmaps to define.

Return Value

- XPOSCommand object.
-

printBitmapToFlashWithN(_:andM:)

Print the Bitmap Downloaded to FLASH

```
func printBitmapToFlash(withN n: Int32, andM m: Int32) -> XPOSCommand!
```

Parameters

- `n`
Mode parameter.
- `m`
Mode parameter.

Return Value

- XPOSCommand object.
-

clearFlashBmp()

Clear the BMP Stored in Flash

```
func clearFlashBmp() -> XPOSCommand!
```

Return Value

- XPOSCommand object.
-

defineDownloadedRAMBmp(_:bmpType:)

Define the Bitmap Downloaded to RAM

```
func defineDownloadedRAMBmp(_ image: NSImage, bmpType: BMPType) -> XPOSCommand!
```

Parameters

- `image`
The bitmap image to define.
- `type`
The image effect.

Return Value

- XPOSCommand object.
-

printDownloadRAMBmp(_:)

Print the Bitmap Downloaded to RAM

```
func printDownloadRAMBmp(_ mode: BmpScaleType) -> XPOSCommand!
```

Parameters

- `mode`
The print mode. Options include:

Variable	Description
BmpNormal	Original size
BmpWidthDouble	Double width
BmpHeightDouble	Double height
BmpWidthHeightDouble	Double width and height

Return Value

- XPOSCommand object.

printBarcodeWithType(_:andContent:)

Print Barcode

```
func printBarcodeWithType(_ type: BarcodeType, andContent content: String) ->
XPOSCommand!
```

Parameters

- `type`

The barcode type. Options include:

Variable	Description
BarcodeTypeUPCA	UPC A
BarcodeTypeUPCE	UPCE
BarcodeTypeJAN13_EAN13	JAN-13
BarcodeTypeJAN8_EAN8	JAN-8
BarcodeTypeCode39	Code 39
BarcodeTypeITF	ITF
BarcodeTypeCodabar	Codabar
BarcodeTypeCode93	Code 93
BarcodeTypeCode128	Code 128

- `content`

The barcode content to print.

Return Value

- XPOSCommand object.
-

selectHRICharacterPrintPosition(_:)

Select HRI Character Print Position

```
func selectHRICharacterPrintPosition(_ position: Int) -> XPOSCommand!
```

Parameters

- `position`
The position parameter for HRI character printing.

Return Value

- XPOSCommand object.
-

selectHRIFont(_:)

Select HRI Font

```
func selectHRIFont(_ font: Int) -> XPOSCommand!
```

Parameters

- `font`
The font parameter for HRI character printing.

Return Value

- XPOSCommand object.
-

selectBarcodeHeight(_:)

Select Barcode Height

```
func selectBarcodeHeight(_ height: Int) -> XPOSCommand!
```

Parameters

- `height`
The barcode height to set.

Return Value

- XPOSCommand object.
-

setBarcodeWidth(_:)

Set Barcode Width

```
func setBarcodeWidth(_ width: Int) -> XPOSCommand!
```

Parameters

- `width`
The barcode width to set.

Return Value

- XPOSCommand object.
-

printQRCodeWithContent(_:)

Print QR Code

```
func printQRCodeWithContent(_ content: String) -> XPOSCommand!
```

Parameters

- `content`
The QR code content to print.

Return Value

- XPOSCommand object.
-

printQRCodeWithContent(_:moduleSize:)

Print QR Code with Module Size

```
func printQRCodeWithContent(_ content: String, moduleSize: Int) -> XPOSCommand!
```

Parameters

- `content`
The QR code content to print.
- `moduleSize`
The module size, range [1, 16], default is 8.

Return Value

- XPOSCommand object.

printQRCodeWithContent(_:moduleSize:ecLevel:)

Print QR Code with Error Correction Level

```
func printQRCodeWithContent(_ content: String, moduleSize: Int, ecLevel: QRErrorCorrectionLevel) -> XPOSCommand!
```

Parameters

- `content`
The QR code content to print.
- `moduleSize`
The module size, range [1, 16], default is 8.
- `ecLevel`
The error correction level, default is QRErrorCorrectionLevelL. Options include:

Variable	Description
QRErrorCorrectionLevelL	Error correction level L (7%)
QRErrorCorrectionLevelM	Error correction level M (15%)
QRErrorCorrectionLevelQ	Error correction level Q (25%)
QRErrorCorrectionLevelH	Error correction level H (30%)

Return Value

- XPOSCommand object.
-

formFeed()

Print and Return to Standard Mode

```
func formFeed() -> XPOSCommand!
```

Return Value

- XPOSCommand object.
-

cancelPrintData()

Cancel Print Data in Page Mode

```
func cancelPrintData() -> XPOSCommand!
```

Return Value

- XPOSCommand object.
-

pageModePrint()

Print in Page Mode

```
func pageModePrint() -> XPOSCommand!
```

Return Value

- XPOSCommand object.
-

selectPageMode()

Select Page Mode

```
func selectPageMode() -> XPOSCommand!
```

Return Value

- XPOSCommand object.
-

selectStandardMode()

Select Standard Mode

```
func selectStandardMode() -> XPOSCommand!
```

Return Value

- XPOSCommand object.
-

selectPrintDirection(_:)

Select Print Direction in Page Mode

```
func selectPrintDirection(_ direction: Int) -> XPOSCommand!
```

Parameters

- `direction`
The direction to set for printing.

Return Value

- XPOSCommand object.
-

setAbsolutePrintPositionWithLow(_:high:)

Set Absolute Print Position

```
func setAbsolutePrintPositionWithLow(_ nL: Int, high nH: Int) -> XPOSCommand!
```

Parameters

- `nL`
The low byte.
- `nH`
The high byte.

Return Value

- XPOSCommand object.
-

printAndFeedPaper(_:)

Print and Feed Paper

```
func printAndFeedPaper(_ lines: Int) -> XPOSCommand!
```

Parameters

- `lines`
The number of lines to feed after printing.

Return Value

- XPOSCommand object.
-

selectOrCancelRotation(_:)

Select/Cancel 90-degree Clockwise Rotation

```
func selectOrCancelRotation(_ rotation: Int) -> XPOSCommand!
```

Parameters

- `rotation`
The rotation flag (0 for cancel, 1 for select).

Return Value

- XPOSCommand object.
-

setPrintArea(_:y:width:height:)

Set Print Area in Page Mode

```
func setPrintArea(_ x: Int, y: Int, width: Int, height: Int) -> XPOSCommand!
```

Parameters

- `x`
The horizontal starting position, default is 0.
- `y`
The vertical starting position, default is 0.
- `width`
The area width.
- `height`
The area height.

Return Value

- XPOSCommand object.
-

enableOrDisableKeys(_:)

Enable/Disable Keys

```
func enableOrDisableKeys(_ flag: Int) -> XPOSCommand!
```

Parameters

- `flag`
The flag to enable or disable keys.

Return Value

- XPOSCommand object.
-

selectCodePage(_:)

Select Character Code Table

```
func selectCodePage(_ charset: CharacterCodePage) -> XPOSCommand!
```

Parameters

- `charset`
The code page number to select.

Return Value

- XPOSCommand object.
-

selectOrCancelInvertedPrint(_:)

Select/Cancel Inverted Print Mode

```
func selectOrCancelInvertedPrint(_ mode: Int) -> XPOSCommand!
```

Parameters

- `mode`
The mode to select or cancel.

Return Value

- XPOSCommand object.
-

setRelativeHorizontal(_:)

Set Relative Horizontal Print Position

```
func setRelativeHorizontal(_ xPosition: Int) -> XPOSCommand!
```

Parameters

- `xPosition`
The horizontal relative displacement to set.

Return Value

- XPOSCommand object.
-

setAbsoluteVertical(_:)

Set Absolute Vertical Position in Page Mode

```
func setAbsoluteVertical(_ yPosition: Int) -> XPOSCommand!
```

Parameters

- `yPosition`
Absolute vertical position.

Return Value

- XPOSCommand object.
-

setRelativeVertical(_:)

Set Relative Vertical Position in Page Mode

```
func setRelativeVertical(_ yPosition: Int) -> XPOSCommand!
```

Parameters

- `yPosition`
Relative vertical position.

Return Value

- XPOSCommand object.
-

setMotionUnitsWithHorizontal(_:vertical:)

Set Horizontal and Vertical Motion Units

```
func setMotionUnitsWithHorizontal(_ x: Int, vertical y: Int) -> XPOSCommand!
```

Parameters

- `x`
Horizontal motion unit.
- `y`
Vertical motion unit.

Return Value

- XPOSCommand object.
-

selectCancelBWReversePrintMode(_:)

Select/Cancel Black/White Reverse Print Mode

```
func selectCancelBWReversePrintMode(_ mode: Int) -> XPOSCommand!
```

Parameters

- `mode`
Mode parameter.

Return Value

- XPOSCommand object.
-

setLeftMargin(_:)

Set Left Margin

```
func setLeftMargin(_ left: Int) -> XPOSCommand!
```

Parameters

- `left`
Left margin.

Return Value

- XPOSCommand object.
-

cutPaper()

Select Paper Cut Mode and Cut Paper

```
func cutPaper() -> XPOSCommand!
```

Return Value

- XPOSCommand object.
-

cutPaperWithMode(_:)

Select Paper Cut Mode and Cut Paper

```
func cutPaperWithMode(_ mode: PaperCutMode) -> XPOSCommand!
```

Parameters

- `mode`

Paper cut mode. Options include:

Variable	Description
PaperCutModeFullCut	Full cut (0, 48)
PaperCutModeFullCutAlt	Full cut (0, 48) alternate
PaperCutModePartialCut	Partial cut (1, 49)
PaperCutModePartialCutAlt	Partial cut (1, 49) alternate

Return Value

- XPOSCommand object.

cutPaperWithDistance(_:)

Select Paper Cut Mode and Cut Paper with Distance

```
func cutPaperWithDistance(_ distance: Int) -> XPOSCommand!
```

Parameters

- `distance`

Feed distance, feed distance × (vertical motion unit) inches and partial cut.

Return Value

- XPOSCommand object.

setPrintAreaWidth(_:)

Set Print Area Width

```
func setPrintAreaWidth(_ width: Int) -> XPOSCommand!
```


Parameters

- `width`
Print area width.

Return Value

- XPOSCommand object.

realTimeStatus(_:)

Real-time Status Transmission

```
func realTimeStatus(_ status: PrinterAllStatusType) -> XPOSCommand!
```

Parameters

- `status`
Status. Options include:

Variable	Description
PrinterStatusTypePrint	Printing status
PrinterStatusTypeOffline	Offline status
PrinterStatusTypeError	Error status
PrinterStatusTypePaper	Paper status

Return Value

- XPOSCommand object.

returnStatus(_:)

Return Status

```
func returnStatus(_ statusType: Int) -> XPOSCommand!
```

Parameters

- `statusType`
Status type.

Return Value

- XPOSCommand object.
-

printerBeep(_:time:)

Printer Receives Order Beep

```
func printerBeep(_ count: Int, time: Int) -> XPOSCommand!
```

Parameters

- `count`
Buzzer beep count, range (1~9).
- `time`
Beep duration for each time, range (1~9), each beep duration is time * 50ms.

Return Value

- XPOSCommand object.
-

printerBeepAndFlashAlarm(_:count:time:)

Printer Receives Order Beep and Alarm Light Flashing

```
func printerBeepAndFlashAlarm(_ count: Int, count interval: Int, time mode: AlarmState)  
-> XPOSCommand!
```

Parameters

- `count`
Range 1~20, indicating the number of alarm light flashes or buzzer beeps.
- `interval`
Range 1~20, indicating the alarm light flash interval time is interval * 50ms, or the buzzer beep time is interval * 50ms.
- `mode`
Mode. Options include:

Variable	Description
AlarmStateNone	Buzzer does not beep, and the alarm light does not flash
AlarmStateBuzzer	Buzzer beeps
AlarmStateLight	Alarm light flashes
AlarmStateBoth	Buzzer beeps, and the alarm light flashes

Return Value

- XPOSCommand object.

openCashBoxWithPinNumber(_:)

Open the Cash Drawer

```
func openCashBoxWithPinNumber(_ pinNum: CashDrawerPin) -> XPOSCommand!
```

Parameters

- `pinNum`
Pin number connected. Options include:

Variable	Description
CashDrawerPin2	Cash drawer socket pin 2 (0, 48)
CashDrawerPin2Alt	Cash drawer socket pin 2 (0, 48) alternate
CashDrawerPin5	Cash drawer socket pin 5 (1, 49)
CashDrawerPin5Alt	Cash drawer socket pin 5 (1, 49) alternate

Return Value

- XPOSCommand object.

openCashBoxWithPinNumber(_:onTime:offTime:)

Open the Cash Drawer with Timing

```
func openCashBoxWithPinNumber(_ pinNum: CashDrawerPin, onTime: Int, offTime: Int) -> XPOSCommand!
```

Parameters

- `pinNum`
Pin number connected.
- `onTime`
Cash drawer pulse start time ($\text{onTime} * 2\text{ms}$), range [0, 255]. Default is 30.
- `offTime`
Cash drawer pulse end time ($\text{offTime} * 2\text{ms}$), range [0, 255]. Default is 255. If $\text{onTime} > \text{offTime}$, the end time is $\text{onTime} * 2\text{ms}$.

Return Value

- XPOSCommand object.
-