

Class

XTSPLCommand

```
class XTSPLCommand: NSObject
```

getCommand()

Get Print Command

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

Return Value

- NSData object containing the print data of the current TSPL object.

setCharEncoding(_:)

Set Character Encoding

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

Parameters

- `encoding`
Encoding format.

Return Value

- XTSPLCommand object.

addData(_:)

Add Custom Data

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

Parameters

- `customData`
Custom data content.

Return Value

- XTSPLCommand object.
-

sizeInch(width:height:)

Set Label Size in Inches

```
func sizeInch(width: Double, height: Double) -> XTSPLCommand!
```

Parameters

- `width`
Size width in inches.
- `height`
Size height in inches.

Return Value

- XTSPLCommand object.
-

sizeMm(width:height:)

Set Label Size in Millimeters

```
func sizeMm(width: Double, height: Double) -> XTSPLCommand!
```

Parameters

- `width`
Size width in mm.
- `height`
Size height in mm.

Return Value

- XTSPLCommand object.
-

gapInch(m:n:)

Define the Gap Height Between Two Labels in Inches

```
func gapInch(m: Double, n: Double) -> XTSPLCommand!
```

Parameters

- `m`
Gap height in inches.
- `n`
Gap height compensation value.

Return Value

- XTSPLCommand object.
-

gapMm(m:n:)

Define the Gap Height Between Two Labels in Millimeters

```
func gapMm(m: Double, n: Double) -> XTSPLCommand!
```

Parameters

- `m`
Gap height in mm.
- `n`
Gap height compensation value.

Return Value

- XTSPLCommand object.
-

speed(_:)

Set Printer Print Speed

```
func speed(_ speed: Double) -> XTSPLCommand!
```

Parameters

- `speed`
Print speed in inches per second.

Return Value

- XTSPCommand object.
-

density(_:)

Set Printer Print Density

```
func density(_ density: Int) -> XTSPCommand!
```

Parameters

- `density`
Density, range [0, 15].

Return Value

- XTSPCommand object.
-

cls()

Clear Print Buffer

```
func cls() -> XTSPCommand!
```

Return Value

- XTSPCommand object.
-

offsetInch(_:)

Set Label Paper Offset in Inches

```
func offsetInch(_ offset: Double) -> XTSPCommand!
```

Parameters

- `offset`
Offset in inches.

Return Value

- XTSPCommand object.
-

offsetMm(_:)

Set Label Paper Offset in Millimeters

```
func offsetMm(_ offset: Double) -> XTSPCommand!
```

Parameters

- `offset`
Offset in mm.

Return Value

- XTSPCommand object.
-

direction(_:)

Set Printer Print Direction

```
func direction(_ direction: DirectionTSPL) -> XTSPCommand!
```

Parameters

- `direction`
Print direction.

Return Value

- XTSPCommand object.
-

direction(_:isMirror:)

Set Printer Print Direction with Mirroring Option

```
func direction(_ direction: DirectionTSPL, isMirror: Bool) -> XTSPCommand!
```

Parameters

- `direction`
Print direction.
- `isMirror`
Whether to mirror the print.

Return Value

- XTSPCommand object.
-

feed(_:)

Feed Forward

```
func feed(_ length: Int) -> XTSPCommand!
```

Parameters

- `length`
Feed length, unit is dot. Range [1, 9999].

Return Value

- XTSPCommand object.
-

referenceAt(x:y:)

Define the Origin Coordinates of the Label Paper

```
func referenceAt(x: Int, y: Int) -> XTSPCommand!
```

Parameters

- `x`
Horizontal coordinate, unit is dot.
- `y`
Vertical coordinate, unit is dot.

Return Value

- XTSPCommand object.
-

shift(_:)

Move the Vertical Position of the Label Image

```
func shift(_ n: Int) -> XTSPCommand!
```

Parameters

- `n`
Vertical coordinate of the label image, unit is dot, the maximum size that can be set is 1 inch.

Return Value

- XTSPCommand object.
-

barAt(x:y:width:height:)

Print a Bar at the Specified Position

```
func barAt(x: Int, y: Int, width: Int, height: Int) -> XTSPCommand!
```

Parameters

- `x`
Bar starting horizontal coordinate, unit is dot.
- `y`
Bar starting vertical coordinate, unit is dot.
- `width`
Bar width, unit is dot.
- `height`
Bar height, unit is dot.

Return Value

- XTSPCommand object.
-

boxAt(x:y:width:height:thickness:)

Print a Box at the Specified Position

```
func boxAt(x: Int, y: Int, width: Int, height: Int, thickness: Int) -> XTSPCommand!
```

Parameters

- `x`
Box upper left X coordinate.
- `y`
Box upper left Y coordinate.
- `width`
Box width.
- `height`
Box height.
- `thickness`
Border thickness.

Return Value

- XTSPCommand object.
-

ellipseAt(x:y:width:height:thickness:)

Draw an Ellipse

```
func ellipseAt(x: Int, y: Int, width: Int, height: Int, thickness: Int) ->  
XTSPCommand!
```

Parameters

- `x`
X coordinate of the upper left corner, unit is dot.
- `y`
Y coordinate of the upper left corner, unit is dot.
- `width`
Width of the ellipse, unit is dot.
- `height`
Height of the ellipse, unit is dot.
- `thickness`
Ellipse line thickness, unit is dot.

Return Value

- XTSPCommand object.
-

backFeed(_:)

Label Backfeed

```
func backFeed(_ length: Int) -> XTSPCommand!
```

Parameters

- `length`
Backfeed length, unit is dot. Range [1, 9999].

Return Value

- XTSPCommand object.
-

formFeed()

Print and Feed Forward to the Next Label

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

Return Value

- XTSPCommand object.
-

home()

Print and Feed Forward to the Label Starting Position

```
func home() -> XTSPCommand!
```

Return Value

- XTSPCommand object.
-

print()

Print Label

```
func print() -> XTSPCommand!
```

Return Value

- XTSPCommand object.

print(m:n:)

Print Label with Copies

```
func print(m: Int, n: Int) -> XTSPCommand!
```

Parameters

- `m`
Specify how many sets of labels to print.
- `n`
Print how many copies of each set.

Return Value

- XTSPCommand object.

barcodeAt(x:y:codeType:height:readable:rotation:narrow:wide:content:)

Print Barcode at Specified Position

```
func barcodeAt(x: Int, y: Int, codeType: String, height: Int, readable: ReadableType, rotation: TSPLRotation, narrow: Int, wide: Int, content: String) -> XTSPCommand!
```

Parameters

- `x`
Barcode starting horizontal coordinate, unit is dot.
- `y`
Barcode starting vertical coordinate, unit is dot.
- `codeType`
Barcode type, available types refer to Barcode Code Types in `xConstants.h`.

Variable	Description
<code>kBarcodeTypeCode128</code>	Code 128, auto switch code subset.
<code>kBarcodeTypeCode128Manual</code>	Code 128, manual switch code subset.

kBarcodeTypeEAN128	EAN128, auto switch code subset.
kBarcodeTypeInterleaved25	Interleaved 2 of 5.
kBarcodeTypeInterleaved25C	Interleaved 2 of 5 with check digit.
kBarcodeTypeCode39	Code 39, auto switch standard and full ASCII mode.
kBarcodeTypeCode39C	Code 39 with check digit.
kBarcodeTypeCode93	Code 93.
kBarcodeTypeEAN13	EAN 13.
kBarcodeTypeEAN13_2	EAN 13 with 2-digit add-on.
kBarcodeTypeEAN13_5	EAN 13 with 5-digit add-on.
kBarcodeTypeEAN8	EAN 8.
kBarcodeTypeEAN8_2	EAN 8 with 2-digit add-on.
kBarcodeTypeEAN8_5	EAN 8 with 5-digit add-on.
kBarcodeTypeCodabar	Codabar.
kBarcodeTypePostnet	Postnet.
kBarcodeTypeUPCA	UPC-A.
kBarcodeTypeUPCA_2	UPC-A with 2-digit add-on.
kBarcodeTypeUPCA_5	UPC-A with 5-digit add-on.
kBarcodeTypeUPCE	UPC-E.
kBarcodeTypeUPCE_2	UPC-E with 2-digit add-on.
kBarcodeTypeUPCE_5	UPC-E with 5-digit add-on.
kBarcodeTypeCpost	China post.
kBarcodeTypeMSI	MSI.
kBarcodeTypeMSIC	MSI with check digit.
kBarcodeTypePlessey	PLESSEY.
kBarcodeTypeITF14	ITF14.
kBarcodeTypeEAN14	EAN14.
kBarcodeTypeCode11	Code 11.
kBarcodeTypeTelepen	Telepen.

- `height`
Barcode height, unit is dot.
- `readable`
Whether to print readable characters, default is `ReadableTypeLeft`, displayed on the left.
- `rotation`
Barcode rotation type.
- `narrow`
Narrow bar code ratio factor, unit is dot, default is 2.
- `wide`
Wide bar code ratio factor, unit is dot, default is 2.
- `content`
Barcode content.

Return Value

- XTSPLCommand object.

codaBlockFAt(x:y:rotation:content:)

Draw CODABLOCK F Type Barcode

```
func codaBlockFAt(x: Int, y: Int, rotation: TSPLRotation, content: String) ->
XTSPLCommand!
```

Parameters

- `x`
X coordinate.
- `y`
Y coordinate.
- `rotation`
Rotation angle.
- `content`
CODABLOCK barcode content.

Return Value

- XTSPLCommand object.

codePage(_:)

Set Printer International Code Page

```
func codePage(_ code: String) -> XTSPCommand!
```

Parameters

- `code`
Code page name.

Return Value

- XTSPCommand object.
-

sound(level:interval:)

Set Printer Buzzer Beep Times and Interval

```
func sound(level: Int, interval: Int) -> XTSPCommand!
```

Parameters

- `level`
Sound volume: 0~9.
- `interval`
Sound interval: 1~4095.

Return Value

- XTSPCommand object.
-

country(_:)

Select International Character Set

```
func country(_ code: String) -> XTSPCommand!
```

Parameters

- `code`
Available character set code.

Return Value

- XTSPLCommand object.
-

limitFeedInch(_:)

Limit Forward Feed in Inches

```
func limitFeedInch(_ length: Double) -> XTSPLCommand!
```

Parameters

- `length`
Limit length in inches.

Return Value

- XTSPLCommand object.
-

limitFeedMm(_:)

Limit Forward Feed in Millimeters

```
func limitFeedMm(_ length: Double) -> XTSPLCommand!
```

Parameters

- `length`
Limit length in mm.

Return Value

- XTSPLCommand object.
-

bitmapAt(x:y:mode:image:)

Print Bitmap

```
func bitmapAt(x: Int, y: Int, mode: BMPMode, image: NSImage) -> XTSPLCommand!
```

Parameters

- `x`
Bitmap upper left X coordinate.
- `y`
Bitmap upper left Y coordinate.
- `mode`
Bitmap mode.
- `image`
Bitmap object.

Return Value

- XTSPCommand object.
-

zlibBitmapAt(x:y:mode:image:)

Transfer Compressed Image to Printer

```
func zlibBitmapAt(x: Int, y: Int, mode: BMPMode, image: NSImage) -> XTSPCommand!
```

Parameters

- `x`
Bitmap upper left X coordinate.
- `y`
Bitmap upper left Y coordinate.
- `mode`
Bitmap mode.
- `image`
Bitmap object.

Return Value

- XTSPCommand object.
-

putBmpAt(x:y:filename:)

Print BMP Format Image

```
func putBmpAt(x: Int, y: Int, filename: String) -> XTSPCommand!
```

Parameters

- `x`
BMP image X coordinate.
- `y`
BMP image Y coordinate.
- `filename`
Downloaded BMP image name.

Return Value

- XTSPLCommand object.
-

putBmpAt(x:y:filename:contrast:)

Print BMP Format Image

```
func putBmpAt(x: Int, y: Int, filename: String, contrast: Int) -> XTSPLCommand!
```

Parameters

- `x`
BMP image X coordinate.
- `y`
BMP image Y coordinate.
- `filename`
Downloaded BMP image name (supports ZPL *.GRF files).
- `contrast`
Grayscale contrast ratio, default is 80, recommended range is 60 to 100.

Return Value

- XTSPLCommand object.
-

putPcxAt(x:y:filename:)

Print PCX Format Image

```
func putPcxAt(x: Int, y: Int, filename: String) -> XTSPLCommand!
```


Parameters

- `x`
PCX image X coordinate.
- `y`
PCX image Y coordinate.
- `filename`
Downloaded PCX image name (case sensitive).

Return Value

- XTSPCommand object.

qrCodeAt(x:y:ecLevel:cellWidth:mode:rotation:content:)

Print QR Code at Specified Position

```
func qrCodeAt(x: Int, y: Int, ecLevel: String, cellWidth: Int, mode: String, rotation: TSPLRotation, content: String) -> XTSPCommand!
```

Parameters

- `x`
QR code upper left X coordinate.
- `y`
QR code upper left Y coordinate.
- `ecclevel`
Error correction level: `kECLLevelL` (7%), `kECLLevelM` (15%), `kECLLevelQ` (25%), `kECLLevelH` (30%).
- `cellwidth`
QR code cell size, range [1, 10].
- `mode`
QR code encoding mode: `kQRCodeModeAuto` (auto generate encoding), `kQRCodeModeManual` (manual generate encoding).
- `rotation`
Rotation angle [0, 90, 180, 270].
- `content`
QR code content.

Return Value

- XTSPCommand object.

dataMatrixAt(x:y:width:height:content:)

Draw DataMatrix 2D Barcode

```
func dataMatrixAt(x: Int, y: Int, width: Int, height: Int, content: String) ->
XTSPLCommand!
```

Parameters

- `x`
X coordinate (unit DOT).
- `y`
Y coordinate (unit DOT).
- `width`
Barcode area width (unit DOT).
- `height`
Barcode area height (unit DOT).
- `content`
DataMatrix data content.

Return Value

- XTSPLCommand object.

pdf417At(x:y:width:height:rotate:content:)

Draw PDF417 2D Barcode

```
func pdf417At(x: Int, y: Int, width: Int, height: Int, rotate: TSPLRotation, content:
String) -> XTSPLCommand!
```

Parameters

- `x`
X coordinate (unit DOT).
- `y`
Y coordinate (unit DOT).
- `width`
Expected width (unit DOT).
- `height`
Expected height (unit DOT).
- `rotate`
Counterclockwise rotation (0, 90, 180, 270).

- `content`
PDF417 barcode data content.

Return Value

- XTSPCommand object.
-

textAt(x:y:font:content:)

Print Text at Specified Position

```
func textAt(x: Int, y: Int, font: String, content: String) -> XTSPCommand!
```

Parameters

- `x`
Text upper left X coordinate.
- `y`
Text upper left Y coordinate.
- `font`
Font name, available font names refer to Font name in XConstants.h.
- `content`
Text content.

Return Value

- XTSPCommand object.
-

textAt(x:y:font:xRatio:yRatio:content:)

Print Text with Magnification Ratios

```
func textAt(x: Int, y: Int, font: String, xRatio: Int, yRatio: Int, content: String) -> XTSPCommand!
```

Parameters

- `x`
Text upper left X coordinate.
- `y`
Text upper left Y coordinate.
- `font`
Font name, available font names refer to Font name in XConstants.h.

- `xRatio`
Text horizontal magnification ratio, range [1, 10].
- `yRatio`
Text vertical magnification ratio, range [1, 10].
- `content`
Text content.

Return Value

- XTSPCommand object.

textAt(x:y:font:rotation:xRatio:yRatio:content:)

Print Text at Specified Position

```
func textAt(x: Int, y: Int, font: String, rotation: TSPLRotation, xRatio: Int, yRatio: Int, content: String) -> XTSPCommand!
```

Example

```
let command = XTSPCommand()  
command  
  .sizeMm(70, height: 85)  
  .gapMm(2, n: 0)  
  .cls()  
  .textAt(x: 0, y: 50, font: kFNT_8_12, rotation: .rotation0, xRatio: 1, yRatio: 1,  
content: "kFNT_8_12")  
  .textAt(x: 0, y: 100, font: kFNT_12_20, rotation: .rotation0, xRatio: 1, yRatio: 1,  
content: "kFNT_12_20")  
  .textAt(x: 0, y: 150, font: kFNT_16_24, rotation: .rotation0, xRatio: 1, yRatio: 1,  
content: "kFNT_16_24")  
  .textAt(x: 0, y: 200, font: kFNT_24_32, rotation: .rotation0, xRatio: 1, yRatio: 1,  
content: "kFNT_24_32")  
  .textAt(x: 0, y: 250, font: kFNT_32_48, rotation: .rotation0, xRatio: 1, yRatio: 1,  
content: "kFNT_32_48")  
  .textAt(x: 0, y: 300, font: kFNT_14_19, rotation: .rotation0, xRatio: 1, yRatio: 1,  
content: "kFNT_14_19")  
  .textAt(x: 0, y: 350, font: kFNT_14_25, rotation: .rotation0, xRatio: 1, yRatio: 1,  
content: "kFNT_14_25")  
  .textAt(x: 0, y: 400, font: kFNT_21_27, rotation: .rotation0, xRatio: 1, yRatio: 1,  
content: "kFNT_21_27")  
  .print()
```

Parameters

- `x`
Text upper left X coordinate.
- `y`
Text upper left Y coordinate.
- `font`
Font name, common font names refer to Font name in `XConstants.h`.

Variable	Description
<code>FNT_8_12</code>	8 x 12 alphanumeric font
<code>FNT_12_20</code>	12 x 20 alphanumeric font
<code>FNT_16_24</code>	16 x 24 alphanumeric font
<code>FNT_24_32</code>	24 x 32 alphanumeric font
<code>FNT_32_48</code>	32 x 48 alphanumeric font
<code>FNT_14_19</code>	14 x 19 alphanumeric font OCR-B
<code>FNT_14_25</code>	14 x 25 alphanumeric font OCR-A
<code>FNT_21_27</code>	21 x 27 alphanumeric font OCR-B
<code>FNT_SIMPLIFIED_CHINESE</code>	Simplified Chinese 24x24 font (GB code)
<code>FNT_TRADITIONAL_CHINESE</code>	Traditional Chinese 24x24 font (Big5 code)
<code>FNT_KOREAN</code>	Korean 24x24 font (KS code)

- `rotation`
Rotation angle [0, 90, 180, 270].
- `xRatio`
Text horizontal magnification ratio, range [1, 10].
- `yRatio`
Text vertical magnification ratio, range [1, 10].
- `content`
Text content.

Return Value

- XTSPLCommand object.
-

textBlockAt(x:y:width:height:font:rotation:xMul:yMul:content:)

Print Paragraph of Text at Specified Position

```
func textBlockAt(x: Int, y: Int, width: Int, height: Int, font: String, rotation: TSPLRotation, xMul: Int, yMul: Int, content: String) -> XTSPCommand!
```

Parameters

- `x`
Text X starting coordinate.
- `y`
Text Y starting coordinate.
- `width`
Text area width.
- `height`
Text area height.
- `font`
Font name, available common font names refer to Font name in XConstants.h.
- `rotation`
Text rotation direction (0, 90, 180, 270).
- `x_mul`
Horizontal magnification ratio, maximum 10x, range [1, 10].
- `y_mul`
Vertical magnification ratio, maximum 10x, range [1, 10].
- `content`
Text content.

Return Value

- XTSPCommand object.

eraseAt(x:y:width:height:)

Erase Content in Specified Area

```
func eraseAt(x: Int, y: Int, width: Int, height: Int) -> XTSPCommand!
```

Parameters

- `x`
Area upper left X coordinate.
- `y`
Area upper left Y coordinate.
- `width`
Area width.
- `height`
Area height.

Return Value

- XTSPLCommand object.
-

reverseAt(x:y:width:height:)

Reverse Print Content in Specified Area

```
func reverseAt(x: Int, y: Int, width: Int, height: Int) -> XTSPLCommand!
```

Parameters

- `x`
Area upper left X coordinate.
- `y`
Area upper left Y coordinate.
- `width`
Area width.
- `height`
Area height.

Return Value

- XTSPLCommand object.
-

cut()

Perform Paper Cutting

```
func cut() -> XTSPLCommand!
```

Return Value

- XTSPLCommand object.
-

setPeel(_:)

Set Peel Mode

```
func setPeel(_ isOpen: Bool) -> XTSPLCommand!
```

Parameters

- `isOpen`
Whether to enable peel mode.

Return Value

- XTSPLCommand object.
-

setTear(_:)

Set Tear Mode

```
func setTear(_ isOpen: Bool) -> XTSPLCommand!
```

Parameters

- `isOpen`
Whether to enable tear mode.

Return Value

- XTSPLCommand object.
-

setCutter(_:)

Set Cutter Working Mode

```
func setCutter(_ pieces: Int) -> XTSPLCommand!
```


Parameters

- `pieces`
 - `0`: Disable cutter function.
 - `-1`: Set the printer to cut after the entire print job is finished.
 - `pieces >= 1`: Set to cut multiple labels at a time ($1 \leq \text{pieces} \leq 65535$).

Example

```
setCutter(0) // Disable cutter function
setCutter(-1) // Set the printer to cut after the entire print job is finished
setCutter(1) // Set to cut 1 label at a time
```

Return Value

- XTSPCommand object.

blineInch(m:n:)

Set Black Mark Spacing (Inches)

```
func blineInch(m: Double, n: Double) -> XTSPCommand!
```

Parameters

- `m`
Black mark height, range: [0.1, 1] feet or [2.54, 25.4] mm.
- `n`
Additional paper feed length. Range [0, label length].

Return Value

- XTSPCommand object.

blineMm(m:n:)

Set Black Mark Spacing (Millimeters)

```
func blineMm(m: Double, n: Double) -> XTSPCommand!
```

Parameters

- `m`
Black mark height, range: [0.1, 1] feet or [2.54, 25.4] mm.
- `n`
Additional paper feed length. Range [0, label length].

Return Value

- XTSPLCommand object.
-

gapDetect()

Gap Calibration

```
func gapDetect() -> XTSPLCommand!
```

Return Value

- XTSPLCommand object.
-

blinedDetect()

Black Mark Calibration

```
func blinedDetect() -> XTSPLCommand!
```

Return Value

- XTSPLCommand object.
-

autoDetect()

Auto Calibration

```
func autoDetect() -> XTSPLCommand!
```

Return Value

- XTSPLCommand object.
-

initialPrinter()

Restore Printer to Factory Settings

```
func initialPrinter() -> XTSPLCommand!
```

Note

- Use with caution. Please distinguish it from the `initializePrinter` method of `XESCCCommand`.

Return Value

- XTSPLCommand object.
-

move()

Move Files from DRAM to FLASH

```
func move() -> XTSPLCommand!
```

Return Value

- XTSPLCommand object.
-

files()

Print Current Memory Capacity

```
func files() -> XTSPLCommand!
```

Return Value

- XTSPLCommand object.
-

killDRAM(_:)

Delete Files in DRAM

```
func killDRAM(_ filename: String) -> XTSPLCommand!
```

Parameters

- `filename`
File name.

Example

```
killDRAM("FILENAME") // Delete the corresponding file in DRAM
killDRAM("*.PCX") // Delete all PCX files in DRAM
killDRAM("*") // Delete all files in DRAM
```

Return Value

- XTSPLCommand object.
-

killFLASH(_:)

Delete Files in Built-in FLASH

```
func killFLASH(_ filename: String) -> XTSPLCommand!
```

Parameters

- `filename`
File name.

Example

```
killFLASH("FILENAME") // Delete the corresponding file in FLASH
killFLASH("*.PCX") // Delete all PCX files in FLASH
killFLASH("*") // Delete all files in FLASH
```

Return Value

- XTSPLCommand object.
-

autoResponse(_:)

Set Printer Auto Response

```
func autoResponse(_ mode: AutoResMode) -> XTSPLCommand!
```

Parameters

- `mode`

Mode options: ON/OFF/BATCH.

Return Value

- XTSPCommand object.
-