

## Class

# XZPLCommand

```
@interface XZPLCommand : NSObject
```

## getCommand

### Get the Print Command

```
- (NSData *)getCommand;
```

### Description

Retrieve the current ZPL object's print data.

### Return Value

- NSData object containing the print data.

## setCharEncoding:

### Set Character Encoding

```
- (XZPLCommand *)setCharEncoding:(NSStringEncoding)encoding;
```

### Parameters

- `encoding`  
Encoding format, default is GB\_18030\_2000.

### Return Value

- XZPLCommand object.

## addData:

### Add Custom Data

```
- (XZPLCommand *)addData:(NSData *)customData;
```

## Parameters

- `customData`  
Custom data content.

## Return Value

- XZPLCommand object.
- 

# start

---

## Start of the Label

```
- (XZPLCommand *)start;
```

## Description

This method is used for the beginning of the label.

**Note:** The start command must be added at the beginning of the print content.

## Return Value

- XZPLCommand object.
- 

# end

---

## End of the Label

```
- (XZPLCommand *)end;
```

## Description

End of the label format. Calling this method will print the label.

**Note:** The end command must be added at the end of the print content.

## Return Value

- XZPLCommand object.
- 

# addTextAtX:y:content:

---

# Text Printing

```
- (XZPLCommand *)addTextAtX:(int)x y:(int)y content:(NSString *)content;
```

## Parameters

- `x`  
Starting x value of the text.
- `y`  
Starting y value of the text.
- `content`  
Text content.

## Return Value

- XZPLCommand object.

---

## addTextAtX:y:fontName:content:

# Text Printing

```
- (XZPLCommand *)addTextAtX:(int)x  
                        y:(int)y  
      fontName:(NSString *)fontName  
      content:(NSString *)content;
```

## Parameters

- `x`  
Starting x value of the text.
- `y`  
Starting y value of the text.
- `fontName`  
Text font type, default is FNT\_F.

Variable	Basic Font Size (Height x Width)
FNT_A	9 x 5
FNT_B	11 x 7
FNT_C	18 x 10
FNT_D	18 x 10
FNT_E	28 x 15
FNT_F	26 x 13
FNT_G	60 x 40
FNT_O	15 x 12

- `content`  
Text content.

## Return Value

- XZPLCommand object.

# addTextAtX:y:fontName:sizeW:sizeH:content:

## Text Printing

```

- (XZPLCommand *)addTextAtX:(int)x
    y:(int)y
    fontName:(NSString *)fontName
    sizeW:(int)sizeW
    sizeH:(int)sizeH
    content:(NSString *)content;

```

## Parameters

- `x`  
Starting x value of the text.
- `y`  
Starting y value of the text.
- `fontName`  
Text font type, default is FNT\_F.
- `sizeW`  
Effective text width, default is the basic size. Use integer multiples of the basic size.

- `sizeH`  
Effective text height, default is the basic size. Use integer multiples of the basic size.
- `content`  
Text content.

## Return Value

- XZPLCommand object.

---

# addTextAtX:y:fontName:rotation:sizeW:sizeH:content:

## Text Printing

```
- (XZPLCommand *)addTextAtX:(int)x
    y:(int)y
    fontName:(NSString *)fontName
    rotation:(RotationZPL)rotation
    sizeW:(int)sizeW
    sizeH:(int)sizeH
    content:(NSString *)content;
```

## Parameters

- `x`  
Starting x value of the text.
- `y`  
Starting y value of the text.
- `fontName`  
Text font type, default is FNT\_F.
- `rotation`  
Clockwise rotation angle, default is ZPLRotation0.

Variable	Description
ZPLRotation0	No rotation
ZPLRotation90	Rotate 90 degrees
ZPLRotation180	Rotate 180 degrees
ZPLRotation270	Rotate 270 degrees

- `sizeW`  
Effective text width, default is the basic size. Use integer multiples of the basic size.
- `sizeH`  
Effective text height, default is the basic size. Use integer multiples of the basic size.

- `content`  
Text content.

## Return Value

- XZPLCommand object.

---

# setCustomFont:extension:alias:codePage:

## Set Custom Font

```
- (XZPLCommand *)setCustomFont:(NSString *)fontName
                        extension:(NSString *)extension
                        alias:(NSString *)alias
                        codePage:(CodePageZPL)codePage;
```

## Parameters

- `fontName`  
Font library name.
- `extension`  
Font name suffix.
- `alias`  
Font alias, corresponding to the `fontName` in `addText`. Range: `A` to `z` and `0` to `9`.
- `codePage`  
Refer to the `CodePageZPL` enumeration for details:

Value	Encoding (Code Page)	Code Page Number
ZPLCodePageUSA1	USA 1	0 (Code Page 850)
ZPLCodePageUTF8	UTF-8	28

## Return Value

- XZPLCommand object.

---

# setPrinterWidth:

## Set Printer Width

```
- (XZPLCommand *)setPrinterWidth:(int)width;
```

## Parameters

- `width`  
Paper width in dots.

## Return Value

- XZPLCommand object.
- 

# setLabelLength:

## Set Label Length

```
- (XZPLCommand *)setLabelLength:(int)height;
```

## Parameters

- `height`  
Label length in dots. Range: `1` to `32000`, not exceeding the maximum label size.

## Return Value

- XZPLCommand object.
- 

# addReverseAtX:y:width:height:

## Reverse Area

```
- (XZPLCommand *)addReverseAtX:(int)x  
                               y:(int)y  
                               width:(int)width  
                               height:(int)height;
```

## Parameters

- `x`  
Starting x value of the area.
- `y`  
Starting y value of the area.
- `width`  
Area width.
- `height`  
Area height.

## Return Value

- XZPLCommand object.
- 

# addReverseAtX:y:width:height:radius:

---

## Reverse Area with Rounded Corners

```
- (XZPLCommand *)addReverseAtX:(int)x
                        y:(int)y
                        width:(int)width
                        height:(int)height
                        radius:(int)radius;
```

## Parameters

- `x`  
Starting x value of the area.
- `y`  
Starting y value of the area.
- `width`  
Area width.
- `height`  
Area height.
- `radius`  
Rounding degree, range: `0~8`, default is `0`.

## Return Value

- XZPLCommand object.
- 

# addBoxAtX:y:width:height:thickness:

---

## Draw Rectangle

```
- (XZPLCommand *)addBoxAtX:(int)x
                        y:(int)y
                        width:(int)width
                        height:(int)height
                        thickness:(int)thickness;
```



## Parameters

- `x`  
Starting x value of the rectangle.
- `y`  
Starting y value of the rectangle.
- `width`  
Rectangle width.
- `height`  
Rectangle height.
- `thickness`  
Line thickness.

## Return Value

- XZPLCommand object.

---

# addBoxAtX:y:width:height:thickness:radius:

## Draw Rectangle with Rounded Corners

```
- (XZPLCommand *)addBoxAtX:(int)x
                        y:(int)y
                        width:(int)width
                        height:(int)height
                        thickness:(int)thickness
                        radius:(int)radius;
```

## Parameters

- `x`  
Starting x value of the rectangle.
- `y`  
Starting y value of the rectangle.
- `width`  
Rectangle width.
- `height`  
Rectangle height.
- `thickness`  
Line thickness.
- `radius`  
Rounding degree, range: `0~8`, default is `0`.

## Return Value

- XZPLCommand object.

# addGraphicDiagonalLineAtX:y:orientation:width:height:thickness:

## Draw Diagonal Line

```
- (XZPLCommand *)addGraphicDiagonalLineAtX:(int)x
                                y:(int)y
                                orientation:(DiagonalDirection)orientation
                                width:(int)width
                                height:(int)height
                                thickness:(int)thickness;
```

## Parameters

- `x`  
Horizontal starting position.
- `y`  
Vertical starting position.
- `orientation`  
Direction of the diagonal line.

Variable	Description
DiagonalDirectionRight	Right-leaning diagonal line (or <code>/</code> )
DiagonalDirectionLeft	Left-leaning diagonal line (or <code>\</code> )

- `width`  
Frame width (range: `1-32000`, unit: dot).
- `height`  
Frame height (range: `1-32000`, unit: dot).
- `thickness`  
Border thickness (range: `1-32000`, unit: dot).

## Return Value

- XZPLCommand object.

# addGraphicEllipseAtX:y:width:height:thickness:

# Graphic Ellipse

```
- (XZPLCommand *)addGraphicEllipseAtX:(int)x
    y:(int)y
    width:(int)width
    height:(int)height
    thickness:(int)thickness;
```

## Parameters

- `x`  
Horizontal starting position.
- `y`  
Vertical starting position.
- `width`  
Ellipse width, range: 3–4095, unit: dot.
- `height`  
Ellipse height, range: 3–4095, unit: dot.
- `thickness`  
Border thickness, range: 2–4095, unit: dot.

## Return Value

- XZPLCommand object.

---

# addGraphicCircleAtX:y:diameter:thickness:

## Graphic Circle

```
- (XZPLCommand *)addGraphicCircleAtX:(int)x
    y:(int)y
    diameter:(int)diameter
    thickness:(int)thickness;
```

## Parameters

- `x`  
Horizontal starting position.
- `y`  
Vertical starting position.
- `diameter`  
Circle diameter, range: 3–4095, unit: dot.

- `thickness`

Border thickness, range: 1–4095, unit: dot.

## Return Value

- XZPLCommand object.

---

# addBarcodeAtX:y:codeType:content:

---

## 1D Barcode

```
- (XZPLCommand *)addBarcodeAtX:(int)x
    y:(int)y
    codeType:(ZPLBarCodeType)codeType
    content:(NSString *)content;
```

## Parameters

- `x`  
Starting x value of the barcode.
- `y`  
Starting y value of the barcode.
- `codeType`  
Barcode type. See the `ZPLBarCodeType` enumeration:

Variable	Description
ZPLBarCode11	Code 11
ZPLBarCode25	Code 25
ZPLBarCode39	Code 39
ZPLBarCodeEAN8	EAN-8
ZPLBarCodeUPCE	UPC-E
ZPLBarCode93	Code 93
ZPLBarCode128	Code 128
ZPLBarCodeEAN13	EAN-13
ZPLBarCodeCODA	CODABAR
ZPLBarCodeMSI	MSI
ZPLBarCodePLESSEY	PLESSEY
ZPLBarCodeUPCEAN	UPC-EAN
ZPLBarCodeUPCA	UPC-A

- `content`  
Barcode text content.

## Return Value

- XZPLCommand object.

# addBarcodeAtX:y:codeType:content:height:

## Add 1D Barcode

```

- (XZPLCommand *)addBarcodeAtX:(int)x
    y:(int)y
    codeType:(ZPLBarCodeType)codeType
    content:(NSString *)content
    height:(int)height;

```

## Parameters

- `x`  
Starting x value of the barcode.

- `y`  
Starting y value of the barcode.
- `codeType`  
Barcode type. See the specific `ZPLBarCodeType` enumeration.
- `content`  
Barcode text content.
- `height`  
Barcode height. Default is 50 dots.

## Return Value

- XZPLCommand object.

# addBarcodeAtX:y:codeType:ratio:textPosition:content:width:height:

## Add 1D Barcode with Additional Options

```
- (XZPLCommand *)addBarcodeAtX:(int)x
    y:(int)y
    codeType:(ZPLBarCodeType)codeType
    ratio:(RotationZPL)ratio
    textPosition:(HriTextZPL)textPosition
    content:(NSString *)content
    width:(int)width
    height:(int)height;
```

## Parameters

- `x`  
Starting x value of the barcode.
- `y`  
Starting y value of the barcode.
- `codeType`  
Barcode type. See the specific `ZPLBarCodeType` enumeration.
- `ratio`  
Rotation angle. See the `RotationZPL` enumeration.
- `textPosition`  
Text position relative to the barcode. Default is `ZPLHriTextBelow`.  
Possible values:

Variable	Description
<code>ZPLHriTextNone</code>	No human-readable text
<code>ZPLHriTextBelow</code>	Human-readable text below the barcode
<code>ZPLHriTextAbove</code>	Human-readable text above the barcode

- `content`  
Barcode text content.
- `width`  
Barcode module width. Default is 2 dots.
- `height`  
Barcode height. Default is 50 dots.

## Return Value

- XZPLCommand object.

---

## addQRCodeAtX:y:factor:text:

### QR Code

```
- (XZPLCommand *)addQRCodeAtX:(int)x
                        y:(int)y
                factor:(int)factor
                text:(NSString *)text;
```

## Parameters

- `x`  
Starting x value of the QR code.
- `y`  
Starting y value of the QR code.
- `factor`  
Magnification factor, range: 1–10, default is 3.
- `text`  
QR code content.

## Return Value

- XZPLCommand object.

---

## printImageAtX:y:image:

## Print Image

```
- (XZPLCommand *)printImageAtX:(int)x  
                        y:(int)y  
                        image:(UIImage *)image;
```

### Parameters

- `x`  
Starting x value of the image.
- `y`  
Starting y value of the image.
- `image`  
Image object to print.

### Return Value

- XZPLCommand object.

---

## printImageAtX:y:wRatio:hRatio:image:

## Print Image

```
- (XZPLCommand *)printImageAtX:(int)x y:(int)y wRatio:(int)wRatio hRatio:(int)hRatio  
image:(UIImage *)image;
```

### Parameters

- `x`  
Starting x value of the image.
- `y`  
Starting y value of the image.
- `wRatio`  
Width scaling ratio, range: 1-10.
- `hRatio`  
Height scaling ratio, range: 1-10.
- `image`  
Image object.



## Return Value

- XZPLCommand object.

---

# downloadGraphic:source:name:image:

## Download Graphic

```
- (XZPLCommand *)downloadGraphic:(ZPLDeviceType)source name:(NSString *)name image:
(NSImage *)image;
```

## Parameters

- `source`  
Storage device for the image. Default is `DeviceTypeR`. Optional types are defined in `ZPLDeviceType`.
- `name`  
Name of the stored image. Accepted values: 1 to 8 alphanumeric characters. If no name is specified, `UNKNOWN` is used.
- `image`  
Image object.

## Return Value

- XZPLCommand object.

---

# printGraphicAtX:y:source:name:mx:my:

## Call Graphic

```
- (XZPLCommand *)printGraphicAtX:(int)x
                        y:(int)y
source:(ZPLDeviceType)source
name:(NSString *)name
mx:(int)mx
my:(int)my;
```

## Parameters

- `x`  
Starting x value of the image.
- `y`  
Starting y value of the image.
- `source`  
Storage device for the image. Default is `DeviceTypeR`. Optional types are defined in `ZPLDeviceType`.

- `name`  
Name and extension obtained when downloading the image.
- `mx`  
Magnification factor in the x-axis direction. Default is `1`, range: 1–10.
- `my`  
Magnification factor in the y-axis direction. Default is `1`, range: 1–10.

## Return Value

- XZPLCommand object.

---

# deleteDownloadGraphic:source:name:

## Delete Downloaded Graphic

```
- (XZPLCommand *)deleteDownloadGraphic:(ZPLDeviceType)source name:(NSString *)name;
```

## Parameters

- `source`  
Storage device for the image. Default is `DeviceTypeR`. Optional types are defined in `ZPLDeviceType`.
- `name`  
Name of the stored image. Accepted values: 1 to 8 alphanumeric characters. If no name is specified, `UNKNOWN` is used.

## Return Value

- XZPLCommand object.

---

# addPrintCount:

## Set Print Quantity

```
- (XZPLCommand *)addPrintCount:(int)count;
```

## Parameters

- `count`  
Number of labels to print.

## Return Value

- XZPLCommand object.
- 

## setPrintSpeed:

### Set Print Speed

```
- (XZPLCommand *)setPrintSpeed:(int)speed;
```

## Parameters

- `speed`  
Print speed in inches per second, range: 1-14.

## Return Value

- XZPLCommand object.
- 

## direction:

### Rotate Label

```
- (XZPLCommand *)direction:(BOOL)n;
```

## Parameters

- `n`  
YES = normal orientation, NO = rotate 180 degrees. Default: YES.

## Return Value

- XZPLCommand object.
- 

## setPrintDensity:

### Set Print Density

```
- (XZPLCommand *)setPrintDensity:(int)density;
```

## Parameters

- `density`  
Print density, range: 0–30.

## Return Value

- XZPLCommand object.
-