

Class

XCPCLCommand

```
class XCPCLCommand: NSObject
```

getCommand()

Get the Printing Command

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

Description

- This method retrieves the current CPCL object's printing data.

setCharEncoding(_:)

Set Character Encoding

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

Parameters

- `encoding`
Encoding format, default is GB_18030_2000.

Return Value

- XCPCLCommand object.

addData(_:)

Add Custom Data

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

Parameters

- `customData`
Custom data content.

Return Value

- XCPCLCommand object.
-

initialize(withHeight:)

Label Initialization

```
func initialize(withHeight height: Int) -> XCPCLCommand!
```

Parameters

- `height`
Maximum height of the label.

Return Value

- XCPCLCommand object.
-

initialize(withHeight:count:)

Label Initialization

```
func initialize(withHeight height: Int, count: Int) -> XCPCLCommand!
```

Parameters

- `height`
Maximum height of the label.
- `count`
Number of labels to print, default is 1.

Return Value

- XCPCLCommand object.
-

initialize(withHeight:count:offsetX:)

Label Initialization

```
func initialize(withHeight height: Int, count: Int, offsetX: Int) -> XCPCLCommand!
```

Parameters

- `height`
Maximum height of the label.
- `count`
Number of labels to print, default is 1.
- `offsetx`
Horizontal offset of the label, default is 0.

Return Value

- XCPCLCommand object.
-

addTextAt(x:y:content:)

Text Printing

```
func addTextAt(x: Int, y: Int, content: String) -> XCPCLCommand!
```

Parameters

- `x`
Starting X coordinate of the text.
- `y`
Starting Y coordinate of the text.
- `content`
Text content.

Return Value

- XCPCLCommand object.
-

addTextAt(x:y:font:content:)

Text Printing

```
func addTextAt(x: Int, y: Int, font: FontCPCL, content: String) -> XCPCLCommand!
```

Parameters

- `x`
Starting X coordinate of the text.
- `y`
Starting Y coordinate of the text.
- `font`
Text font type, default is CPCLFont0.

Variable	Description
CPCLFont0	Font 0
CPCLFont1	Font 1
CPCLFont2	Font 2
CPCLFont3	Font 3
CPCLFont4	Font 4
CPCLFont5	Font 5
CPCLFont6	Font 6
CPCLFont7	Font 7
CPCLFont24	Font 24
CPCLFont55	Font 55

- `content`
Text content.

Return Value

- XCPCLCommand object.

addTextAt(x:y:rotation:font:content:)

Text Printing

```
func addTextAt(x: Int, y: Int, rotation: RotationCPCL, font: FontCPCL, content: String)
-> XCPCLCommand!
```

Parameters

- `x`
Starting X coordinate of the text.
- `y`
Starting Y coordinate of the text.
- `rotation`
Clockwise rotation angle, default is CPCLRotation0.

Variable	Description
CPCLRotation0	No rotation
CPCLRotation90	Rotate 90 degrees
CPCLRotation180	Rotate 180 degrees
CPCLRotation270	Rotate 270 degrees

- `font`
Text font type, default is CPCLFont0.
- `content`
Text content.

Return Value

- XCPCLCommand object.

setMag(withWidth:height:)

Enlarge the Resident Font

```
func setMag(withWidth w: Int, height h: Int) -> XCPCLCommand!
```

Parameters

- `w`
Width magnification factor (1~16).
- `h`
Height magnification factor (1~16).

Return Value

- XCPCLCommand object.
-

addPrint()

End Command for Printing

```
func addPrint() -> XCPCLCommand!
```

Description

- This is the end command for the entire command set, which starts file printing.

Return Value

- XCPCLCommand object.
-

addBarcodeAt(x:y:type:height:content:)

Horizontal 1D Barcode

```
func addBarcodeAt(x: Int, y: Int, type: CPCLBarCodeType, height: Int, content: String)
-> XCPCLCommand!
```

Parameters

- `x`
Starting X coordinate of the barcode, in dots.
- `y`
Starting Y coordinate of the barcode, in dots.
- `type`
Barcode type, refer to `CPCLBarCodeType` for details.

Variable	Description
CPCLBarCode128	Code 128
CPCLBarCodeUPCA	UPC-A
CPCLBarCodeUPCE	UPC-E
CPCLBarCodeEAN13	EAN-13
CPCLBarCodeEAN8	EAN-8
CPCLBarCode39	Code 39
CPCLBarCode93	Code 93
CPCLBarCodeCODABAR	CODABAR

- `height`
Barcode unit height.
- `content`
Barcode data content.

Return Value

- `XCPCLCommand` object.

addBarcodeAt(x:y:type:width:ratio:height:content:)

Horizontal 1D Barcode with Width and Ratio

```
func addBarcodeAt(x: Int, y: Int, type: CPCLBarCodeType, width: Int, ratio:
BarCodeRatioCPCL, height: Int, content: String) -> XCPCLCommand!
```

Parameters

- `x`
Starting X coordinate of the barcode, in dots.
- `y`
Starting Y coordinate of the barcode, in dots.
- `type`
Barcode type, refer to `CPCLBarCodeType` for details.
- `width`
Narrow bar unit width, default is `1`.
- `ratio`
Ratio of wide bar to narrow bar. Refer to the following for available options:

Variable	Description
CPCLBarcodeRatio0	Wide:Narrow = 1.5:1
CPCLBarcodeRatio1	Wide:Narrow = 2.0:1
CPCLBarcodeRatio2	Wide:Narrow = 2.5:1
CPCLBarcodeRatio3	Wide:Narrow = 3.0:1
CPCLBarcodeRatio4	Wide:Narrow = 3.5:1
CPCLBarcodeRatio20	Wide:Narrow = 2.0:1
CPCLBarcodeRatio21	Wide:Narrow = 2.1:1
CPCLBarcodeRatio22	Wide:Narrow = 2.2:1
CPCLBarcodeRatio23	Wide:Narrow = 2.3:1
CPCLBarcodeRatio24	Wide:Narrow = 2.4:1
CPCLBarcodeRatio25	Wide:Narrow = 2.5:1
CPCLBarcodeRatio26	Wide:Narrow = 2.6:1
CPCLBarcodeRatio27	Wide:Narrow = 2.7:1
CPCLBarcodeRatio28	Wide:Narrow = 2.8:1
CPCLBarcodeRatio29	Wide:Narrow = 2.9:1
CPCLBarcodeRatio30	Wide:Narrow = 3.0:1

- `height`
Barcode unit height.
- `content`
Barcode data content.

Return Value

- `XCPCCommand` object.

addBarcodeVAt(x:y:type:height:content:)

Vertical 1D Barcode

```
func addBarcodeVAt(x: Int, y: Int, type: CPCLBarCodeType, height: Int, content: String)
-> XCPCLCommand!
```

Parameters

- `x`
Starting X coordinate of the barcode, in dots.
- `y`
Starting Y coordinate of the barcode, in dots.
- `type`
Barcode type, refer to `CPCLBarCodeType` for details.
- `height`
Barcode unit height.
- `content`
Barcode data content.

Return Value

- `XCPCLCommand` object.

addBarcodeVAt(x:y:type:width:ratio:height:content:)

Vertical 1D Barcode with Width and Ratio

```
func addBarcodeVAt(x: Int, y: Int, type: CPCLBarCodeType, width: Int, ratio:
BarCodeRatioCPCL, height: Int, content: String) -> XCPCLCommand!
```

Parameters

- `x`
Starting X coordinate of the barcode, in dots.
- `y`
Starting Y coordinate of the barcode, in dots.
- `type`
Barcode type, refer to `CPCLBarCodeType` for details.
- `width`
Narrow bar unit width, default is `1`.
- `ratio`
Ratio of wide bar to narrow bar, default is `CPCLBarCodeRatio1 (2.0:1)`.

- `height`
Barcode unit height.
- `content`
Barcode data content.

Return Value

- `XCPCLError` object.
-

barcodeText(_:)

Add Barcode Text

```
func barcodeText(_ offsetX: Int) -> XCPCLError!
```

Parameters

- `offsetx`
Text offset from the barcode, in units.

Return Value

- `XCPCLError` object.
-

barcodeTextOff()

Cancel Barcode Text

```
func barcodeTextOff() -> XCPCLError!
```

Return Value

- `XCPCLError` object.
-

addQRCodeAt(x:y:content:)

Draw QR Code (Simple Version)

```
func addQRCodeAt(x: Int, y: Int, content: String) -> XCPCLError!
```

Parameters

- `x`
Starting X coordinate of the QR code.
- `y`
Starting Y coordinate of the QR code.
- `content`
QR code data content.

Return Value

- `XCPCCLCommand` object.

addQRCodeAt(x:y:codeModel:cellWidth:content:)

Draw QR Code (Advanced Version)

```
func addQRCodeAt(x: Int, y: Int, codeModel: QRCodeModesCPCL, cellWidth: Int, content: String) -> XCPCCLCommand!
```

Parameters

- `x`
Starting X coordinate of the QR code.
- `y`
Starting Y coordinate of the QR code.
- `codeModel`
QR Code specification number. Possible values:
 - `CPCLQRCodeModeORG`: Original specification.
 - `CPCLQRCodeModeEnhance`: Enhanced specification.
- `cellWidth`
Cell size, range [1, 32]. Default is 6.
- `content`
QR code data content.

Return Value

- `XCPCCLCommand` object.

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

Draw Rectangle

```
func addBoxAt(x: Int, y: Int, width: Int, height: Int, thickness: Int) -> XCPCLCommand!
```

Parameters

- `x`
Starting X coordinate of the rectangle, in dots.
- `y`
Starting Y coordinate of the rectangle, in dots.
- `width`
Rectangle width, in dots.
- `height`
Rectangle height, in dots.
- `thickness`
Line width.

Return Value

- `XCPCLCommand` object.

addLineAt(x:y:xEnd:yEnd:width:)

Draw Line

```
func addLineAt(x: Int, y: Int, xEnd: Int, yEnd: Int, width: Int) -> XCPCLCommand!
```

Parameters

- `x`
Starting X coordinate of the line, in dots.
- `y`
Starting Y coordinate of the line, in dots.
- `xend`
Ending X coordinate of the line, in dots.
- `yend`
Ending Y coordinate of the line, in dots.
- `width`
Line width.

Return Value

- `XCPCLCommand` object.
-

addInverseLineAt(x:y:xEnd:yEnd:width:)

Reverse the Data in the Specified Area

```
func addInverseLineAt(x: Int, y: Int, xEnd: Int, yEnd: Int, width: Int) ->
XCPCLCommand!
```

Parameters

- `x`
Starting X coordinate of the inverse area, in dots.
- `y`
Starting Y coordinate of the inverse area, in dots.
- `xend`
Ending X coordinate of the inverse area, in dots.
- `yend`
Ending Y coordinate of the inverse area, in dots.
- `width`
Inverse area width.

Return Value

- `XCPCLCommand` object.
-

addGraphicsAt(x:y:image:)

Draw Image

```
func addGraphicsAt(x: Int, y: Int, image: UIImage) -> XCPCLCommand!
```

Parameters

- `x`
Starting X coordinate of the image, in dots.
- `y`
Starting Y coordinate of the image, in dots.
- `image`
Image object.

Return Value

- `XCPCCLCommand` object.
-

addAlign(_:)

Set Field Alignment (Simple Version)

```
func addAlign(_ align: AlignmentsCPCL) -> XCPCCLCommand!
```

Parameters

- `align`

Alignment mode. Possible values:

- `CPCLAlignmentLeft`: Left-align all subsequent fields.
- `CPCLAlignmentCenter`: Center-align all subsequent fields.
- `CPCLAlignmentRight`: Right-align all subsequent fields.

Return Value

- `XCPCCLCommand` object.
-

addAlign(__END)

Set Field Alignment (With Endpoint)

```
func addAlign(_ align: AlignmentsCPCL, end: Int) -> XCPCCLCommand!
```

Parameters

- `align`

Alignment mode. Possible values:

- `CPCLAlignmentLeft`: Left-align all subsequent fields.
- `CPCLAlignmentCenter`: Center-align all subsequent fields.
- `CPCLAlignmentRight`: Right-align all subsequent fields.

- `end`

Alignment end point.

Return Value

- `XCPCLError` object.
-

addSpeed(_:)

Set the Print Speed

```
func addSpeed(_ level: Int) -> XCPCLError!
```

Parameters

- `level`
Speed level, range [0, 5].

Return Value

- `XCPCLError` object.
-

addPageWidth(_:)

Set the Page Width

```
func addPageWidth(_ width: Int) -> XCPCLError!
```

Parameters

- `width`
Page unit width.

Return Value

- `XCPCLError` object.
-

addBeep(_:)

Set the Beep Duration

```
func addBeep(_ length: Int) -> XCPCLError!
```

Parameters

- `length`

Beep duration, in 1/8 second units (e.g., 16 means 2 seconds).

Return Value

- `XCPCLCommand` object.
-

form()

Form Feed

```
func form() -> XCPCLCommand!
```

Return Value

- `XCPCLCommand` object.
-